

z/OS



MVS System Commands

z/OS



MVS System Commands

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page B-1.

Sixteenth Edition, April 2007

This is a major revision of SA22-7627-14.

This edition applies to Version 1 Release 8 of z/OS (5694-A01), Version 1 Release 8 of z/OS.e (5655-G52), and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this document, or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P181
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405

FAX (Other Countries):

Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)

Internet e-mail: mhvrccfs@us.ibm.com

World Wide Web: www.ibm.com/servers/eserver/zseries/zos/webqs.html

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this document
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1988, 2007. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	xiii
Tables	xv
About this document	xvii
Who should use this document	xvii
How to use this document	xvii
Where to find more information	xvii
Information updates on the web	xviii
Using LookAt to look up message explanations	xviii
Using IBM Health Checker for z/OS	xviii
Summary of changes	xxi
Chapter 1. System Operations	1-1
Starting, Loading, and Initializing the System	1-2
Starting the System	1-2
Preparing the System Hardware	1-2
Loading the System Software	1-3
Logging On to the System	1-6
Initializing the System Software	1-7
Starting and Specifying Parameters for the Job Entry Subsystem	1-9
Controlling the System	1-9
Displaying Current System Status	1-10
Displaying the Status of Devices and Availability of Paths	1-10
Sending Commands to Systems in a Sysplex	1-11
Using Commands That Have Sysplex Scope	1-11
Sharing System Commands	1-14
Setting the Time and Changing the System Parameters	1-18
Using the System Restart Function	1-18
Responding To IEA502A	1-19
Responding To BLW004A	1-20
Activating a Workload Management Service Policy	1-20
Switching Workload Management Modes	1-21
Controlling Time-Sharing	1-21
Controlling Jobs	1-22
Starting a Job	1-22
Stopping a Job	1-22
Cancelling a Job	1-22
Passing Information to a Job	1-22
Restarting a Job	1-22
Deferred Restart	1-23
Controlling Started Tasks	1-24
Controlling System Information Recording	1-25
System Management Facilities	1-25
System Trace	1-26
The Generalized Trace Facility	1-26
Master Trace	1-26
Component Trace	1-27
Logrec Recording Medium	1-27
Controlling Automatic Tape Switching	1-27
Defining Automatically Switchable Devices	1-28
Displaying Information About Automatically Switchable Devices	1-28

Interacting with System Functions	1-29
Device Allocation	1-29
Hot I/O Detection	1-31
Device Boxing	1-32
Command Flooding	1-35
Class M1 Commands	1-35
Class M2 Commands	1-36
Class M3 Commands	1-37
Class C1 Commands	1-37
Class C2 Commands	1-38
Class C3 Commands	1-38
Inline Commands	1-38
Responding to Failing Devices	1-39
Quiescing the System	1-39
Stopping the System	1-40
Recovery from Hardware Problems	1-40
Hardware Problems	1-40
Information Provided with Machine Checks	1-41
CPU Errors	1-41
Service Processor Damage	1-44
Storage Errors	1-45
Channel Subsystem Errors	1-46
I/O Device Errors	1-50
Additional Recovery Actions	1-59
Chapter 2. System Configuration	2-1
Reconfiguration	2-1
Dynamic I/O Configuration	2-1
Logical and Physical Reconfiguration	2-2
Reconfiguration Support According to Processor Types	2-2
Reconfiguring a Central Processor	2-2
Reconfiguring Central Storage	2-4
Reconfiguring Channel Paths	2-10
Reconfiguring I/O Devices	2-11
Reconfiguring a Coupling Facility	2-11
Chapter 3. System Console Operations	3-1
Console Characteristics and Operations	3-1
General Characteristics of Display Consoles	3-1
Operations on Display Consoles in Full-Capability Mode	3-7
Handling Consoles in Error Conditions	3-14
Placing a Console in Offline Status	3-19
Interchanging Your Consoles on a Control Unit	3-19
Defining and Changing Console Characteristics	3-20
Using Operator Commands to Change CONSOLxx Statements	3-20
Changing Console Characteristics	3-24
Controlling System Messages and Commands	3-29
Defining Program Function Keys (PFKs)	3-35
Processing Hardcopy	3-38
Chapter 4. MVS System Commands Reference	4-1
Command Syntax Notation	4-12
How to read syntax conventions	4-12
System Command Formats	4-13
Typical Format	4-13
A Second Format	4-14

ACTIVATE Command	4-15
Restrictions	4-15
Syntax	4-15
Parameters	4-15
CANCEL Command	4-19
Syntax	4-20
Parameters	4-20
CHNGDUMP Command	4-25
Dump Options and Modes	4-25
Changing the Dump Mode and Options	4-26
Scope in a Sysplex	4-26
Syntax	4-26
Removing Options from or Resetting the System Dump Options Lists	4-26
Parameters	4-27
Options for SDUMP, SYSABEND, SYSUDUMP, and SYSDUMP	4-29
Resetting Dump Mode to ADD and the Dump Options to Initial Values	4-32
Example: How CHNGDUMP Commands Affect Dump Modes and Options	4-33
Setting the Dump Modes and Options.	4-34
CMDS Command	4-44
Syntax	4-44
Parameters	4-44
CONFIG Command	4-46
Syntax	4-46
Reconfiguring the System Directly	4-47
Reconfiguring the System with a CONFIGxx Parmlib Member	4-53
Reconfiguring the System in Response to a Configuration Display	4-54
CONTROL Command	4-56
Scope in a Sysplex	4-57
Syntax	4-57
Changing Out Of Line Display Area Specifications	4-57
Deleting Retained Action Messages	4-58
Halting the Printing or the Display of a Status Display	4-60
Controlling Displays in Areas	4-60
Removing Information From the Screen	4-62
Activating, Deactivating, or Displaying the Status of the Action Message Retention Facility	4-63
Changing or Displaying the Number of Allowed WTL SYSLOG Buffers	4-64
Changing or Displaying the Number of Allowed WTO and WTOR Message Buffers	4-64
Changing the Time the System Waits for ROUTE Command Responses	4-65
Increasing the Maximum Number of Reply IDs	4-66
Changing or Displaying the Status of WTO Installation Exit IEAVMXIT	4-67
Displaying the SMCS APPLID of the current system and VTAM generic resource name for SMCS	4-68
Setting the APPLID of the System	4-68
Setting or Turning Off the VTAM Generic Resource Name for SMCS	4-69
Changing a PFK Definition	4-70
Deleting Message Queues	4-72
Changing or Displaying Message Deletion and Format Specifications	4-72
Changing or Displaying Time Intervals for Dynamic Displays	4-76
Changing the Operating Mode of a Console	4-77
Selecting the Message Levels for a Console	4-78
DEVSERV Command.	4-80
Using the DEVSERV QDASD option	4-81
Using the DEVSERV QTape option	4-81
Using the DEVSERV QLIB option	4-81

Syntax	4-82
Parameters	4-82
DISPLAY Command	4-97
Scope in a Sysplex	4-99
Syntax	4-100
Displaying APPC/MVS Information	4-100
Displaying ASCH Configuration Information	4-106
Displaying Page Data Set Information	4-108
Displaying the current system level Language Environment run-time options	4-110
Displaying CONTROL Command Functions	4-110
Displaying Attached Coupling Facility Information	4-111
Displaying Console Group Definitions	4-111
Displaying Console Status Information	4-112
Displaying DIAG Parmlib Information	4-116
Displaying Data Lookaside Facility Information	4-117
Displaying Dump Options or Dump Data Set Status	4-119
Displaying Extended MCS Information	4-122
Displaying the Timer Synchronization Mode and ETR Ports	4-127
Displaying Global Resource Serialization Information	4-128
Displaying TSO/E Parmlib Information	4-139
Displaying I/O Configuration Information	4-140
Displaying Captured UCB Information	4-141
Displaying Dynamic Channel Path Management Information	4-141
Displaying Encryption Key Manager (EKM) Status	4-142
Displaying FICON Switch Data Information	4-143
Displaying IOS Group Information	4-143
Displaying IOS HYPERPAV Information	4-143
Displaying MIDAW Facility Status	4-144
Displaying MIH and I/O Timing Limits	4-144
Displaying IOS Storage Residency Information	4-148
Displaying the Devices Stopped by the IOACTION Command	4-148
Displaying IPL Information	4-148
Displaying System Activity	4-149
Displaying Started Task Status	4-157
Displaying Library Lookaside Information	4-160
Displaying the System Logger and its Log Streams	4-161
Displaying the Logrec Recording Medium	4-165
Displaying System Configuration Information	4-166
Displaying MVS Message Service Status and Languages	4-171
Displaying Message Suppression, Retention, Color, Intensity, and Highlighting Options	4-172
Displaying z/OS UNIX System Services Status	4-174
Displaying Operator Information (OPDATA)	4-191
Displaying PARMLIB Information	4-192
Displaying Commands Defined for PFKs	4-194
Displaying Registered Products	4-195
Displaying Entries in the List of APF-Authorized Libraries	4-197
Displaying Dynamic Exits	4-198
Displaying LNLST Information	4-200
Displaying LPA Information	4-202
Displaying System Requests	4-202
Displaying Resource Recovery Services (RRS) Information	4-208
Displaying RTLS Information	4-211
Displaying SLIP Trap Information	4-214
Displaying SMF Data	4-215

Displaying Storage Management Subsystem Information	4-215
Displaying Information about All Subsystems.	4-232
Displaying Static System Symbols	4-235
Displaying the Local and Coordinated Universal Time and Date.	4-235
Displaying Component or Transaction Trace Status	4-236
Displaying Device Status and Allocation	4-239
Displaying Unicode Services.	4-242
Displaying Virtual Storage Information	4-245
Displaying Workload Manager Information.	4-245
Displaying Cross System Coupling Facility (XCF) Information	4-251
DUMP Command.	4-261
Wildcards.	4-261
Syntax.	4-261
Parameters	4-262
Specifying Dump Options.	4-263
DUMPDS Command	4-280
Syntax	4-280
Adding System Dump Resources	4-281
Enabling and Disabling Automatic Dump Data Set Allocation	4-284
Making Dump Data Sets Ready To Receive Dumps	4-285
Deleting System Dump Resources	4-285
Setting the Name-Pattern for Dump Data Sets	4-287
FORCE Command	4-290
Considerations.	4-290
Syntax.	4-291
Parameters	4-291
HALT Command	4-295
Syntax.	4-295
IOACTION Command	4-296
Syntax.	4-296
Parameters	4-296
LIBRARY Command.	4-298
LOG Command	4-299
Syntax.	4-299
Parameters	4-299
LOGOFF Command.	4-300
Syntax.	4-300
LOGON Command	4-301
Scope in a Sysplex	4-301
Syntax.	4-301
Parameters	4-301
MODE Command.	4-303
Syntax.	4-303
Controlling the Recording of Hard Machine Check Interruptions	4-304
Controlling the Recording of System Recovery and Degradation Machine Check Interruptions	4-306
Displaying Recording and Monitoring Status	4-308
MODIFY Command	4-309
Summary of MODIFY	4-309
Using Asterisks in MODIFY Commands	4-310
MODIFY Command Syntax	4-311
Syntax.	4-311
Parameters	4-312
Passing Information to a z/OS UNIX System Services Application	4-313
Modifying TSO/VTAM Time Sharing	4-314
Controlling z/OS UNIX System Services (z/OS UNIX)	4-314

Communicating with the Catalog Address Space	4-321
Communicating with the Device Manager Address Space	4-330
Changing the DLF Processing Mode.	4-331
Changing the DLF parameters	4-331
Displaying DLF Status	4-332
Building and Replacing Library Lookaside Directories	4-332
Operating with the Network File System Server.	4-333
Managing the Object Access Method (OAM)	4-333
Recycling z/OS UNIX System Services (z/OS UNIX)	4-333
Dynamically activating maintenance for z/OS UNIX System Services (z/OS UNIX)	4-335
Stopping a physical file system (PFS) through a logical file system (LFS) interface	4-336
Stopping a Temporary File System (TFS)	4-336
Enabling and Disabling the Application Response Measurement (ARM) Agent and Enterprise Workload Manager (EWLM) platform services . .	4-337
Changing Workload Manager Resource States	4-337
Specifying Data Set Selection Criteria for an External Writer	4-338
Causing an External Writer to Pause	4-340
MONITOR Command	4-343
Scope in a Sysplex	4-343
Syntax	4-343
Parameters	4-343
MOUNT Command	4-345
Scope in a Sysplex	4-345
Syntax	4-345
Parameters	4-345
Tape Library Dataserver Considerations	4-346
MSGRT Command	4-347
Syntax	4-347
Parameters	4-347
Stopping Message Routing	4-349
PAGEADD Command	4-349
Syntax	4-350
Parameters	4-350
PAGEDEL Command	4-351
Syntax	4-352
Parameters	4-352
QUIESCE Command	4-354
Syntax	4-354
REPLY Command	4-355
Using System Symbols in REPLY Commands	4-355
Scope in a Sysplex	4-356
Syntax	4-356
Replies to System Information Requests	4-356
Replies to System Requests During Recovery Processing . .	4-357
Replies to System Security WTORs	4-357
Setting the Time-of-Day Clock	4-357
Specifying Component Trace Options	4-359
Specifying Dump Options	4-359
Specifying SMF Options	4-360
Specifying System Parameters	4-360
RESET Command	4-362
Scope in a Sysplex	4-362
Syntax	4-362
Forcing a Hung MCS or SMCS Console Offline	4-362

Changing Service Classes or Quiescing Work	4-363
ROUTE Command	4-367
Restrictions	4-367
How MVS Displays Aggregated Response from ROUTE	4-368
Using System Symbols in ROUTE Commands	4-369
Syntax	4-370
Parameters	4-370
SEND Command	4-375
Scope in a Sysplex	4-375
Syntax	4-375
Communicating with Other Operators	4-375
Communicating with Specified Users	4-376
Communicating with All Logged-On Terminal Users	4-377
Saving Messages in the Broadcast Data Set	4-379
Listing the Notices Section of the Broadcast Data Set	4-379
Deleting a Message from the Broadcast Data Set (Notices Section)	4-381
SET Command	4-382
Scope in a Sysplex	4-383
Syntax	4-384
Parameters	4-384
SETAPPC Command	4-396
Syntax	4-396
Parameters	4-396
SETCEE Command	4-402
Syntax	4-402
Parameters	4-402
SETCON Command	4-403
Syntax	4-403
Parameters	4-403
SETDMN Command	4-405
Syntax	4-405
Restrictions	4-405
Parameters	4-406
SETETR Command	4-407
Syntax	4-407
Parameters	4-407
SETGRS Command	4-408
Syntax	4-408
Parameters	4-408
SETIOS Command	4-412
Syntax	4-412
Parameters	4-413
SETLOAD Command	4-420
Syntax	4-420
Parameters	4-420
SETLOGR Command	4-422
Scope in a Sysplex	4-422
Syntax	4-422
SETLOGR FORCE Command	4-422
SETLOGRC Command	4-425
Syntax	4-425
Parameters	4-425
SETOMVS Command	4-427
Syntax	4-427
Parameters	4-429
SETPROG Command	4-441

Syntax	4-441
Updating the APF List	4-441
Updating Dynamic Exits	4-443
Updating LNKLST Concatenations	4-446
Managing Dynamic LPA Content	4-451
SETRRS CANCEL Command	4-454
Syntax	4-454
Parameters	4-454
SETRRS SHUTDOWN Command	4-454
Syntax	4-455
Parameters	4-455
SETSMF Command	4-456
Syntax	4-456
Parameters	4-456
SETSMS Command	4-457
Scope in a Sysplex	4-459
Syntax	4-459
Parameters	4-459
SETSSI Command	4-472
Syntax	4-472
Parameters	4-472
SETUNI Command	4-475
Syntax	4-475
Parameters	4-475
SETXCF Command	4-483
Scope in a Sysplex	4-483
Syntax	4-483
SETXCF COUPLE Command	4-483
SETXCF FORCE Command	4-487
SETXCF MODIFY Command	4-489
SETXCF PRSMOPOLICY Command	4-491
SETXCF START Command	4-492
SETXCF STOP Command	4-501
SLIP Command	4-506
Syntax	4-506
Using SLIP Commands	4-506
Processing of SLIP Commands	4-507
Coding SLIP Command Parameters	4-508
Setting a SLIP Trap	4-513
Modifying an Existing SLIP Trap	4-573
Deleting an Existing SLIP Trap	4-574
START Command	4-575
Syntax	4-575
Starting a System Task from a Console	4-575
Starting the APPC/MVS Address Space	4-579
Starting the APPC/MVS Transaction Scheduler Address Space	4-580
Starting the Generalized Trace Facility	4-581
Starting the Library Lookaside (LLA) Address Space	4-583
Starting the Object Access Method (OAM)	4-583
Starting Resource Recovery Services (RRS)	4-584
Starting the System Object Model (SOM) Subsystem	4-585
Starting TSO/VTAM Time-Sharing	4-586
Starting the Virtual Lookaside Facility or Data Lookaside Facility	4-587
Starting an External Writer	4-588
STOP Command	4-590
Syntax	4-590

Stopping a Running Program	4-590
Parameters	4-590
Stopping an ASCH Initiator	4-593
Stopping the Data Lookaside Facility (DLF)	4-593
Stopping the Library Lookaside (LLA) Address Space	4-593
Stopping the Object Access Method (OAM) Address Space	4-594
Stopping a System Object Model (SOM)	4-594
Stopping a Temporary File System (TFS)	4-595
Stopping the Virtual Lookaside Facility (VLF)	4-595
STOPMN Command.	4-596
Syntax.	4-596
Parameters	4-596
STOPPTR Command.	4-597
Scope in a Sysplex	4-597
Syntax.	4-597
Parameters	4-597
SWAP Command.	4-599
Syntax.	4-599
Operator-Requested DDR.	4-599
System-Initiated DDR	4-600
SWITCH Command	4-601
Scope in a Sysplex	4-601
Syntax.	4-601
Parameters	4-601
TRACE Command	4-603
Syntax.	4-603
Parameters	4-603
Specifying TRACE CT Options	4-607
Specifying TRACE TT Options	4-609
TRACK Command	4-613
Scope in a Sysplex	4-613
Syntax.	4-613
Parameters	4-613
UNLOAD Command.	4-615
Scope in a Sysplex	4-615
Syntax.	4-615
Parameters	4-615
VARY Command	4-616
Scope in a Sysplex	4-617
Syntax.	4-618
Controlling Problem Determination Mode for the System Console	4-618
Controlling MCS and SMCS Consoles	4-620
Controlling Hardcopy Processing	4-628
Placing a Console Online or Offline	4-631
Defining a Tape Device as Automatically Switchable	4-632
Placing an I/O Device or a Range of I/O Devices Online or Offline	4-633
Controlling a Global Resource Serialization Complex.	4-638
Placing an I/O Path or Paths Online or Offline	4-639
Changing the state of coupling facility cache structures and volumes	4-642
Placing an Optical Drive or Library Online or Offline	4-645
Placing a System-Managed Tape Library Online or Offline	4-645
Analyzing the State of the PDSE Subsystem.	4-645
Releasing PDSE Latches	4-645
Modifying processing of PDSE monitor	4-645
Display current state of the PDSE monitor	4-646
Changing the SMS Status of a Storage Group or Volume	4-646

Controlling DFSMStvs processing	4-648
Controlling CICSVR processing	4-657
Placing a Switch Port Online or Offline	4-658
Controlling an Application Environment	4-659
Activating a Service Policy	4-662
Removing a System from the XCF Sysplex	4-664
WRITELOG Command	4-665
Syntax	4-665
Parameters	4-665
Appendix. Accessibility	A-1
Using assistive technologies	A-1
Keyboard navigation of the user interface	A-1
z/OS information	A-1
Notices	B-1
Trademarks	B-3
Index	X-1

Figures

1-1.	Format of the LOAD Parameter	1-4
1-2.	Example of a Successful Response to a DISPLAY AUTOSWITCH Command	1-29
1-3.	Boxed Device Recovery Procedure	1-34
1-4.	DASD Devices Shared Between Two Systems	1-48
1-5.	DASD Device Shared Between Three Systems.	1-57
3-1.	Comparison of the Display Screens of Full-Capability and Output-Only Display Consoles	3-5
3-2.	Example of DISPLAY CONSOLES,BACKLOG Command Output	3-18
3-3.	Example of DISPLAY CONSOLES,A Command Output.	3-23
3-4.	Format of a Console Screen in Message Stream Mode.	3-28
4-1.	One System Command Format	4-13
4-2.	A Second System Command Format	4-14
4-3.	Display Output Illustration (Column Descriptions).	4-158
4-4.	Display Output from D A,WTOR (Membername)	4-158
4-5.	Display Output for D A,WTOR (Membername and Identifier)	4-159
4-6.	Display Output for D A,WTOR (Membername and JOBNAME).	4-159
4-7.	Display Output from D A,SYM1	4-159
4-8.	Display Output from D A,SYMTEST	4-159
4-9.	Display Output from D A,SYMBOLS	4-160

Tables

1-1.	Possible Values IMSI Characters	1-5
1-2.	MVS System Commands With Sysplex Scope	1-12
1-3.	Correcting Boxed Conditions	1-35
3-1.	Checking the Commands Defined for Each PFK	3-9
3-2.	Comparison of System Commands and CONSOLE Parameters in CONSOLxx	3-20
3-3.	Comparison of System Commands and INIT Statements in CONSOLxx.	3-21
3-4.	Comparison of VARY HARDCPY Commands and HARDCOPY Statements in CONSOLxx	3-22
3-5.	Command Groups Used to Determine Command Authority	3-24
3-6.	Message Routing Codes	3-30
4-1.	System Command Summary	4-2
4-2.	Syntax conventions	4-12
4-3.	Specifying FORCE with EMIF	4-17
4-4.	CANCEL Command Tasks	4-19
4-5.	Summary of the CHNGDUMP Command	4-25
4-6.	Example of How CHNGDUMP Commands Affect Dump Modes and Options	4-33
4-7.	Summary of the CONFIG Command.	4-46
4-8.	Summary of the CONTROL Command	4-56
4-9.	Sysplex Scope for CONTROL Command	4-57
4-10.	Summary of the DISPLAY Command	4-97
4-11.	Sysplex Scope for DISPLAY Command	4-99
4-12.	Displaying System Activity: Information for the LIST Operand	4-151
4-13.	Displaying System Activity: Information for the ALL Operand	4-152
4-14.	Displaying System Activity: Information for a Specific Name.	4-153
4-15.	Examples of START Commands to Start Jobs.	4-155
4-16.	Examples of DISPLAY Commands	4-155
4-17.	Denomination (multiplier) character used for various OMVS commands	4-177
4-18.	Summary of the DUMPDS Command	4-280
4-19.	FORCE Command Tasks	4-290
4-20.	Summary of the MODE Command	4-303
4-21.	MODE Parameters Allowed for Machine Check Interruptions	4-304
4-22.	Summary of the MODIFY Command	4-309
4-23.	Examples of START Commands to Start Jobs.	4-311
4-24.	Examples of MODIFY Commands	4-311
4-25.	Possible Volume and Device Combinations on MOUNT Command	4-346
4-26.	Summary of the REPLY Command.	4-355
4-27.	Summary of the RESET Command.	4-362
4-28.	Summary of the SEND Command	4-375
4-29.	Sysplex Scope for SET Command	4-383
4-30.	Summary of the SETLOGR Command	4-422
4-31.	Sysplex Scope for the SETLOGR Command	4-422
4-32.	1-Character Parameter Limit Multipliers	4-428
4-33.	Acceptable Parameter Statements and Their Applicability	4-436
4-34.	Comparison of SET SMS with SETSMS	4-458
4-35.	Incorrect Combinations of SETSMS Parameters	4-459
4-36.	Summary of the SETXCF Command	4-483
4-37.	Sysplex Scope for the SETXCF Command	4-483
4-38.	Summary of the SLIP Command.	4-506
4-39.	Summary of the SLIP SET parameters	4-515
4-40.	Summary of the START Command.	4-575
4-41.	Summary of the STOP Command	4-590
4-42.	Summary of the SWAP Command	4-599
4-43.	Summary of the VARY Command	4-616
4-44.	Sysplex Scope for VARY Command	4-617

4-45. Goal Mode and Compatibility Mode Actions for VARY WLM,APPLENV.	4-661
--	-------

About this document

This document describes how to use MVS system operator commands for the z/OS (5694-A01) and z/OS.e (5655-G52) operating systems. Although you can also perform many of the tasks described in this book using JES2 or JES3 commands, this book describes only the MVS (base control program) system commands. For information about commands for other z/OS elements, such as communications server (IP and SNA), DFSMS, JES2, JES3, and RACF, see *z/OS Information Roadmap*. For information about commands for other software products that run on z/OS, see *z/OS Software Products Collection*, SK3T-4270.

Who should use this document

This document is intended for anyone using a console and system commands to control the operating system. This document assumes that the user understands the hardware controls and features of the installation. It also assumes that the user understands the general organization and functions of a z/OS operating system.

How to use this document

To describe the basic tasks within these general tasks and to provide a convenient system commands reference, this document is organized as follows:

- Chapter 1, “System Operations,” on page 1-1, describes the tasks of running the system from the time the system comes up to the time the system goes down for a normal or abnormal reason.
- Chapter 2, “System Configuration,” on page 2-1, describes the reconfiguration actions of hardware units and partitions. You can physically or logically reconfigure the hardware units of a system; for a partitionable processor complex, you can partition the complex into two sides or merge the two sides together.
- Chapter 3, “System Console Operations,” on page 3-1, describes the consoles that MVS supports as operators’ consoles. “Console Characteristics and Operations” on page 3-1 describes the operations and characteristics that you *cannot* define, including the operations that are common on all operator’s consoles. “Defining and Changing Console Characteristics” on page 3-20 describes the console characteristics that you *can* define, including the commands that operators and system programmers can use to tailor the consoles and console operations to the installation’s requirements. “Defining and Changing Console Characteristics” on page 3-20 also describes how to restrict the use of system commands based on which operator issues the command and/or which MCS or SMCS console the operator uses.
- Chapter 4, “MVS System Commands Reference,” on page 4-1, summarizes the function, syntax, and parameters of all the MVS system commands that you can use to control both the system and the MCS and SMCS consoles.

Where to find more information

Where necessary, this document references information in other documents, using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

Information updates on the web

For the latest information updates that have been provided in PTF cover letters and Documentation APARs for z/OS and z/OS.e, see the online document at:

<http://www.s390.ibm.com:80/bookmgr-cgi/bookmgr.cmd/B00KS/ZIDOCMST/CCONTENTS>

This document is updated weekly and lists documentation changes before they are incorporated into z/OS publications.

Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM® messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from these locations to find IBM message explanations for z/OS® elements and features, z/VM®, VSE/ESA™, and Clusters for AIX® and Linux™:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations using LookAt from a TSO/E command line (for example: TSO/E prompt, ISPF, or z/OS UNIX® System Services).
- Your Microsoft® Windows® workstation. You can install LookAt directly from the *z/OS Collection* (SK3T-4269) or the *z/OS and Software Products DVD Collection* (SK3T-4271) and use it from the resulting Windows graphical user interface (GUI). The command prompt (also known as the DOS > command line) version can still be used from the directory in which you install the Windows version of LookAt.
- Your wireless handheld device. You can use the LookAt Mobile Edition from www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookatm.html with a handheld device that has wireless access and an Internet browser (for example: Internet Explorer for Pocket PCs, Blazer or Eudora for Palm OS, or Opera for Linux handheld devices).

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from:

- A CD-ROM in the *z/OS Collection* (SK3T-4269).
- The *z/OS and Software Products DVD Collection* (SK3T-4271).
- The LookAt Web site (click **Download** and then select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

Using IBM Health Checker for z/OS

IBM Health Checker for z/OS is a z/OS component that installations can use to gather information about their system environment and system parameters to help identify potential configuration problems before they impact availability or cause outages. Individual products, z/OS components, or ISV software can provide checks that take advantage of the IBM Health Checker for z/OS framework. This book refers to checks or messages associated with this component.

For additional information about checks and about IBM Health Checker for z/OS, see *IBM Health Checker for z/OS: User's Guide*. Starting with z/OS V1R4, z/OS

users can obtain the IBM Health Checker for z/OS from the z/OS Downloads page at www.ibm.com/servers/eserver/zseries/zos/downloads/.

SDSF also provides functions to simplify the management of checks. See *z/OS SDSF Operation and Customization* for additional information.

Summary of changes

**Summary of changes
for SA22-7627-15
z/OS Version 1 Release 8
as updated April 2007**

This document contains information previously presented in *z/OS MVS System Commands*, SA22-7627-14, which supports z/OS Version 1 Release 8.

New information:

- DISPLAY IOS,EKM command allows to display the current status of the encryption key management. See “Displaying Encryption Key Manager (EKM) Status” on page 4-142.
- DISPLAY IOS,HYPERPAV command allows to display the current HyperPav enablement status. See “Displaying IOS HYPERPAV Information” on page 4-143.
- DISPLAY WLM command contains a new option AM,ALL. The APPLENV parameter is updated too. See “Displaying Workload Manager Information” on page 4-245.
- DEVSERV QDASD command contains a new option RDFEATS. See “DEVSERV Command” on page 4-80.
- “DUMP Command” on page 4-261 contains new information about the SYMDEF= parameter.
- “SETIOS Command” on page 4-412 contains the following new options:
 - EKM
 - HYPERPAV
- “SETOMVS Command” on page 4-427 contains the following new options:
 - SHRLIBRGNSIZE
 - SHRLIBMAXPAGES
- “VARY CN command” on page 4-621 contains two new options:
 - INTIDS
 - UNKNIDS
- VARY *devspec*,ONLINE command contains new information about the UNCOND parameter. See “Placing an I/O Device or a Range of I/O Devices Online or Offline” on page 4-633.
- VARY PATH,ONLINE command contains new information about the FORCE parameter. See “Placing an I/O Path or Paths Online or Offline” on page 4-639.
- VARY WLM,APPLENV command contains new information about the command scope. See “Controlling an Application Environment” on page 4-659.

Changed information:

- “ACTIVATE Command” on page 4-15 contains changed information about the SOFT=VALIDATE parameter.
- CHNGDUMP SET command contains changed information about the SDUMP,BUFFERS= parameter. See “Setting the Dump Modes and Options” on page 4-34.
- DISPLAY OMVS,CINET command contains changed information about the METRIC parameter on the output of the command. See “Displaying z/OS UNIX System Services Status” on page 4-174.

- MODIFY OMVS command contains changed information about the RESTART,OMVS= parameter. See “Recycling z/OS UNIX System Services (z/OS UNIX)” on page 4-333.
- “SETSMS Command” on page 4-457 contains changed information about the VOLSELMMSG parameter.
- SLIP SET command contains changed information about the COMP= parameter. See “SLIP SET Parameters” on page 4-528.
- “SETXCF START Command” on page 4-492 contains changed information about enhancements for the REALLOCATE process provided by APAR OA08688.
- “TRACE Command” on page 4-603 contains changed information about the WTRSTART= and WTRSTOP= parameters.

Deleted information:

- The "Example of Partitioning" and "Example of Merging Sides" sections are removed from Chapter 2, “System Configuration,” on page 2-1.
- The following commands are not supported any longer on the current release of z/OS. However, in case that you may run on previous levels of z/OS and use this book for reference, these obsolete commands are not physically removed from this book.
 - CONFIG CPUAD..., ONLINE,VFONIVFOFF
 - CONFIG ESTOR
 - CONFIG VF
 - DISPLAY M=ESTOR
 - MODE VS

This document has been enabled for the following types of advanced searches in the online z/OS Library Center: commands.

You may notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

**Summary of changes
for SA22-7627-14
z/OS Version 1 Release 8**

This document contains information previously presented in *z/OS MVS System Commands*, SA22-7627-13, which supports z/OS Version 1 Release 7.

New information:

- Chapter 1, “System Operations,” on page 1-1 contains a new section “Recovery from Hardware Problems” on page 1-40.
- A new chapter Chapter 2, “System Configuration,” on page 2-1 is added.
- “Displaying Extended MCS Information” on page 4-122 contains new options on the D EMCS command: ,ATTR=INTIDS and ,ATTR=UNKNIDS.
- The DISPLAY ETR command is supported for a sysplex running in STP mode.

- “Displaying System Configuration Information” on page 4-166 contains a new option D M=CU that allows you to display information for a specific control unit.
- The D OMVS,FILE command contains the following new options that allow you to display the file system information with selected criteria:
 - NAMEIN
 - OWNERIO
 - EXCEPTIONIE
 - TYPEIT
- The D OMVS,WAITERS command displays more information about file system latch and other waiting threads.
- “Displaying Resource Recovery Services (RRS) Information” on page 4-208 displays status information about RRS coordinated transactions to the system console and SYSLOG.
- “Displaying Unicode Services” on page 4-242 is added.
- The MODIFY DEVMAN command allows you to communicate with the manager address space. See “Communicating with the Device Manager Address Space” on page 4-330.
- The SET DEVSUP=xx command allows you to dynamically change the DEVSPUPxx parmlib member in use. See “SET Command” on page 4-382.
- “SETGRS Command” on page 4-408 contains four new parameters:
 - CNS
 - ENQMAXA
 - ENQMAXU
 - GRSQ
- “SETIOS Command” on page 4-412 contains the following new parameters:
 - IOTHSWAP=YESINO, IOTTERM=YESINO
- “SETRRS SHUTDOWN Command” on page 4-454 provides a normal shutdown command to bring down RRS without resulting in a X’058’ abend.
- “SETXCF Command” on page 4-483 allows you to START and STOP the CFRM event management protocol with the new MSGBASED parameter.

Changed information:

- “Console Characteristics and Operations” on page 3-1 and “Defining and Changing Console Characteristics” on page 3-20 are merged into a new chapter Chapter 3, “System Console Operations,” on page 3-1.
- “Displaying the Local and Coordinated Universal Time and Date” on page 4-235 contains changed information.

Deleted information:

The following commands are no longer supported and the corresponding information is deleted from this edition:

- SWITCH CN
- VARY...,MSTCONS
- DISPLAY CONSOLES,MCONLY
- MSGRT

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

**Summary of changes
for SA22-7627-13
z/OS Version 1 Release 7
as updated April 2006**

This document contains information previously presented in *z/OS MVS System Commands*, SA22-7627-12, which supports z/OS Version 1 Release 7.

New information

- “Displaying MIDAW Facility Status” on page 4-144 displays the status information about the MIDAW facility.
- “SETIOS Command” on page 4-412 contains the new parameter MIDAW.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

**Summary of changes
for SA22-7627-12
z/OS Version 1 Release 7**

This document contains information previously presented in *z/OS MVS System Commands*, SA22-7627-11, which supports z/OS Version 1 Release 6.

New information:

- “TRACE Command” on page 4-603 contains a new parameter.
- “SETAPPC Command” on page 4-396 is a new command, SETAPPC.
- “SETCON Command” on page 4-403 contains a new parameter, MONITOR.
- “Displaying Operator Information (OPDATA)” on page 4-191 contains a new parameter, MONITOR.
- “Recycling z/OS UNIX System Services (z/OS UNIX)” on page 4-333 contains information about new parameters ACTIVATE=SERVICE and DEACTIVATE=SERVICE on the MODIFY OMVS command.
- “SETLOGR Command” on page 4-422 contains a new command, SETLOGR.
- “Displaying z/OS UNIX System Services Status” on page 4-174 contains information about a new parameter ACTIVATE=SERVICE on the DISPLAY OMVS command.
- “SETUNI Command” on page 4-475 is a new command that provides control of the Unicode tables in a conversion image.

Changed information:

- “SLIP Command” on page 4-506 contains changed parameter information.
- Many commands no longer support the command response routing externals L=cc, L=cca, and CN=cc. This change aligns with the requirement that two-digit console ID values are no longer allowed.
- “Displaying z/OS UNIX System Services Status” on page 4-174 contains information about new output for the DISPLAY OMVS,O command. See “Example 6” on page 4-180.
- You cannot issue the following commands on systems at z/OS V1R7 and above:
 - CONTROL D,H
 - CONTROL D,U
 - CONTROL T,REF
 - CONTROL T,UTME=nnn

- MSGRT TR=
- TRACK
- STOPTR
- References to OpenEdition have been replaced with z/OS UNIX System Services or z/OS UNIX.

Deleted information:

- The START DFSMSPKG command is no longer supported and its corresponding information is deleted from this edition.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

Summary of changes

SA22-7627-11

z/OS Version 1 Release 6
as updated March 2005

This document contains information previously presented in *z/OS MVS System Commands*, SA22-7627-10, which supports z/OS Version 1 Release 6.

New information

- The DISPLAY WLM,AM command displays whether the EWLM ARM services are {ENABLED|DISABLED}. See “Displaying Workload Manager Information” on page 4-245 for more information.
- The MODIFY WLM,AM command enables or disables Application Response Measurement (ARM) services and Enterprise Workload Manager (EWLM) platform services. See “Enabling and Disabling the Application Response Measurement (ARM) Agent and Enterprise Workload Manager (EWLM) platform services” on page 4-337 for more information.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

Summary of changes

for SA22-7627-10

z/OS Version 1 Release 6

This document contains information previously presented in *z/OS MVS System Commands*, SA22-7627-09, which supports z/OS Version 1 Release 5.

New information

- “CHNGDUMP Command” on page 4-25 contains information about new parameter ABDUMP,TIMEENQ=yyyy.
- “MODIFY Command” on page 4-309 contains information about new parameter WLM.
- “SET Command” on page 4-382 contains information about new parameters TIMEZONE and IOS.
- “SETIOS Command” on page 4-412 contains information about new parameter FICON,STATS=YESINO.
- “SETSMS Command” on page 4-457 contains information about new parameter DSSTIMEOUT.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

**Summary of changes
for SA22-7627-09
z/OS Version 1 Release 5**

This document contains information previously presented in *z/OS MVS System Commands*, SA22-7627-08, which supports z/OS Version 1 Release 4.

Changed information

- The “SETGRS Command” on page 4-408 command contains a new default for the SYNCHRES parameter.

Starting with z/OS V1R3, WLM compatibility mode is no longer available. This affects the descriptions of the following system commands: DISPLAY DMN, DISPLAY OMVS, MODIFY WLM, RESET, SET, SETDMN, and SETOMVS.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

Chapter 1. System Operations

The tasks of starting, running, and stopping the MVS operating system involve controlling the MVS system software and most of the installation's hardware, including processors, channel paths, and I/O devices. This book is for people who need reference information about these tasks and the MVS system commands. They include:

- Those who develop procedures for the daily operations, including system programmers and lead operators
- Operators who want to learn how to use a console to control MVS and how to change some of the console's characteristics

System planners and system programmers should refer to the *z/OS MVS Planning: Operations* for information on planning:

- System and sysplex operation management
- MCS consoles
- SMCS consoles
- Extended MCS consoles

This chapter describes how to operate an MVS system using MVS system commands. Subsystem (JES2 or JES3) commands can perform many of the same functions as MVS system commands but are described in *z/OS JES2 Commands* and *z/OS JES3 Commands*.

The tasks of operating the MVS system that are described in this chapter include:

- “Starting, Loading, and Initializing the System” on page 1-2
- “Controlling the System” on page 1-9
- “Controlling Time-Sharing” on page 1-21
- “Controlling Jobs” on page 1-22
- “Controlling Started Tasks” on page 1-24
- “Controlling System Information Recording” on page 1-25
- “Controlling Automatic Tape Switching” on page 1-27
- “Interacting with System Functions” on page 1-29
- “Responding to Failing Devices” on page 1-39
- “Quiescing the System” on page 1-39
- “Stopping the System” on page 1-40
- “Recovery from Hardware Problems” on page 1-40

Controlling MVS involves issuing commands on a console and responding to messages that appear on the console screen. Other books that describe controlling MVS include:

- *z/OS MVS JCL Reference*, which documents two job control language statements (the COMMAND statement and the JCL command statement) that you can use to enter system commands through the input job stream.
- *z/OS MVS Planning: Operations*, which contains information about using MCS and extended MCS consoles as well as MVS message and command processing.
- *z/OS MVS Planning: Global Resource Serialization*, which contains information about controlling a global resource serialization (GRS) ring.

Starting, Loading, and Initializing the System

Before the system can do work, you must:

1. Start the system.
2. Prepare the system hardware.
3. Load the system software.
4. Initialize the system software. At this point, your installation might require you to logon to the console. See “Logging On to the System” on page 1-6.
5. Set the time and date, as required.
6. Start the job entry subsystem (JES2 or JES3).
7. Specify all job entry subsystem parameters.

The following sections describe in detail how to start, load, and initialize the system.

Starting the System

Your installation may choose to use the **system console** as the only console required to initialize the system. This console is connected to the processor controller. From here, you load the system software and specify the load parameter. Then you use this console to initialize the system. The initialization programs may require initial values, specify an alternate master catalog, and, perhaps, set the time and date.

If your installation uses MCS consoles, then you may use two separate consoles to initialize the system. The first device is the **system console**, which is connected to the processor controller. From this console, you load the system software and specify the LOAD parameter. Later, during normal operations, this console is used to monitor and service the hardware.

The second device is called the **NIP (nucleus initialization program) console**. In HCD, you can specify a list of device numbers to use as NIP consoles. The initialization programs use the first online and ready device in the list. NIP consoles must be devices that are locally connected to the system using control units that do not support systems network architecture (SNA) protocols. This means that SMCS consoles cannot be used as NIP consoles. If that device is also specified on a CONSOLE statement in CONSOLxx, it is initialized as an MCS console and appears to “change” to an MCS console when console initialization is complete. If no NIP consoles are defined, or no NIP consoles are online when MVS is loaded, MVS tries to use the system console during initialization.

Preparing the System Hardware

To prepare the system hardware for work:

1. Turn on power for the processor.
2. Perform the initial microprogram load (IML) function for the processor.
3. Specify the central storage configuration.
4. Ensure that all volumes required by the system are online.
5. Turn on power for all devices you plan to use as multiple-console support (MCS) consoles.
6. Switch into the configuration all control units for devices that the system needs.

For more information on these procedures, see the processor operator’s guide or your installation’s operations procedures.

Loading the System Software

Once the system hardware is ready, you can use the hardware management console (HMC) to load the system software. Consider the following information for loading system software through the HMC:

1. This task is available in Operator, Advanced Operator, System Programmer, or Service Representative mode.
2. Other products and documentation may refer to this operation as an initial program load (IPL).
3. For daily or routine loading of images, you can customize activation profiles to specify how you want to load images, then use a profile with the Activate task to perform all the operations necessary to make an image operational, including loading it with a control program.

Load (except for a coupling facility image) causes a program to be read from a designated device and initiates the execution of that program. If the CPC is operating in logically partitioned (LPAR) mode, the logical partition is the target of the load. Otherwise, if the CPC is operating in basic mode, the CPC is the target of the load.

To perform a load, do the following:

1. Open the **Task List** from the **Views** area.
2. Open **CPC Recovery** from the **Task List Work Area**.
3. Open **Groups** from the **Views** area.
4. Open the group that contains the CPC image that you want to load.
5. Select one object.
Load is considered a disruptive task. If the object is locked, you must unlock it before continuing.
6. Drag and drop the selected object on **Load** in the **CPC Recovery** tasks area. The **Load** window is displayed with the information that was last used when the CPC image was loaded.
7. Review the information on the window to verify that the object you will load is the correct one. If the information is correct, select the **OK** push button. The **Load Task Confirmation** window is displayed.
8. Review the information on the window to verify that the object you will load is the correct one. If the information is correct, select the **Yes** push button. The **Load Progress** window displays indicating the progress of the load and the outcome.
9. Select the **OK** push button to close the window when the load completes successfully. Otherwise, if the load does not complete successfully, follow the directions on the window to determine the problem and how to correct it.

Use the online Help to get additional information for loading a CPC image.

Once the system hardware is ready, you can use the system console to load the system software. Load the system as follows, using the following fields on the *system control (SYSCTL) frame*. (This example uses the IBM 3090 for illustration.)

1. **T=TARGET CP:** Specifies the target processor for initialization.
2. **A=INITIALIZE SYSTEM CONTROL PROGRAM, A1:** Specifies the device number that contains the system residence volume (IPL volume).
3. **A=INITIALIZE SYSTEM CONTROL PROGRAM, A2:** Specifies the LOAD parameter. For more information, see “Explanation of the A=INITIALIZE SYSTEM CONTROL PROGRAM, A2 Field” on page 1-4.

4. **A=INITIALIZE SYSTEM CONTROL PROGRAM, A3:** Specifies the operator load function to IPL the MVS operating system. For more information, see “Explanation of the A=INITIALIZE SYSTEM CONTROL PROGRAM, A3 Field” on page 1-6.

Explanation of the A=INITIALIZE SYSTEM CONTROL PROGRAM, A2 Field

This field specifies the LOAD parameter. The format of the LOAD parameter is:

1	45	67	8
IODF DASD	LOADxx	PROMPT FEAT.	ALT NUCx
IODF device number	LOADxx suffix	prompt feature	nucleus suffix

Figure 1-1. Format of the LOAD Parameter

The LOAD parameter is eight characters long and contains the following information:

1. The first four characters (characters 1 through 4 of the LOAD parameter) specify the hexadecimal device number for the device that contains the I/O definition file (IODF) VSAM data set. This is also the device on which the search for the LOADxx member of SYSn.IPLPARM or SYS1.PARMLIB begins. The device number can be in the range X'0000' to X'FFFF'. If the number is less than 4 digits, specify leading zeros before the device number. If you do not specify the device number, the system uses the device number of the system residence (SYSRES) volume.
2. The next two characters (characters 5 and 6 of the LOAD parameter) specify the suffix of the LOADxx parmlib member that the system is to use. The LOADxx member contains information about the name of the IODF data set, which master catalog to use, and which IEASYSxx members of SYS1.PARMLIB to use.

The default for the LOADxx suffix is zeros. The system reads the LOADxx and NUCLSTxx members from SYSn.IPLPARM or SYS1.PARMLIB on the volume specified on the LOAD parameter (or the SYSRES volume, if a volume is not specified). Once the system opens the master catalog, the system reads all other members from the SYS1.PARMLIB data set that is pointed to by the master catalog. This SYS1.PARMLIB might be different from the SYS1.PARMLIB data set to which the LOAD parameter points.

For more information about LOADxx, see the description of LOADxx in *z/OS MVS Initialization and Tuning Reference*.

3. The next character (character 7 of the LOAD parameter) specifies the prompting and message suppression characteristics that the system is to use at IPL. This character is commonly known as an initialization message suppression indicator (IMSI).

Suppressing Informational Messages: Some IMSI characters suppress informational messages from the system console, which can speed up the initialization process and reduce message traffic to the console. It can also cause you to miss some critical messages, so you should always review the hardcopy log after initialization is complete.

When the system suppresses informational messages, it displays the following messages:

- Messages with descriptor codes 1, 2, 3, 11, or 12

- Write-to-operator with reply (WTOR) messages
- Command responses
- Synchronous messages that can indicate problems during initialization.

It does not display the contents of a parmlib member, even if the L option has been specified.

Prompting for Operator Responses: You can specify an IMSI character that tells the system to issue a MASTER CATALOG prompt, a SYSTEM PARAMETERS prompt, both, or none:

- If the system issues a MASTER CATALOG prompt, the operator response overrides the values that are specified on the SYSCAT parameter in the LOADxx parmlib member.
- If the system issues a SYSTEM PARAMETERS prompt, the operator response overrides the values that are specified on the SYSPARM parameter in LOADxx. Also, if LOADxx specifies the IEASYMxx parameter which in turn specifies a SYSPARM parameter for IEASYSxx, then the operator response also overrides the values that the SYSPARM parameter in IEASYMxx specifies.
- If the system *does not* prompt the operator, the system uses the values specified in LOADxx. If the SYSCAT and SYSPARM statements are not specified in LOADxx, the system issues one or both prompts to obtain the missing information.

Prompting for the Name of the Master Catalog: If you choose an IMSI character that tells the system *not* to prompt for the master catalog name, the system uses the name specified on the SYSCAT parameter in the LOADxx parmlib member.

The default for the system parameter prompt is to use IEASYS00 in SYS1.PARMLIB, and the default for the master catalog prompt is to use SYSCATLG in SYS1.NUCLEUS.

The following table shows the possible values for the IMSI character. The default value is **period (.)**.

Table 1-1. Possible Values IMSI Characters

IMSI Character	Display Informational Messages	Prompt for Master Catalog Response	Prompt for System Parameters Response
period (.) or blank	No	No	No
A	Yes	Yes	Yes
C	No	Yes	No
D	Yes	Yes	No
M	Yes	No	No
P	No	Yes	Yes
S	No	No	Yes
T	Yes	No	Yes

4. The last character (character 8 of the LOAD parameter) specifies the alternate nucleus identifier (0-9). Use this character at the system programmer's direction. If you do not specify an alternate nucleus identifier, the system loads the standard (or primary) nucleus (IEANUC01) and an architectural extension of the nucleus (IEANUC11 or IEANUC21), unless the NUCLEUS statement is specified in the LOADxx member. For more information, see *z/OS MVS Initialization and Tuning Reference*.

Also consider the following:

1. Decide whether to accept the system prompt indicator default. The default causes the system to suppress messages and not prompt the operator. You might miss critical messages during initialization, so you should review the hardcopy log.

New installations might want to select prompt feature A (display all messages and prompt the operator) or M (display all messages but do not prompt operator) on the system control frame while validating changes and analyzing system errors during the initialization process. Specifying either A or M might increase message traffic.

2. Omit the LOAD parameter when you accept all the IBM-supplied defaults.
3. Each character in the LOAD parameter is positional. If you change any of the defaults you *must* retype the characters or use periods (...) to hold the positions.
4. You *cannot* leave any leading spaces blank, unless the defaults are accepted for the rest of the LOAD parameter.

Explanation of the A=INITIALIZE SYSTEM CONTROL PROGRAM, A3 Field

This field specifies the operator load function to IPL the MVS operating system.

Selecting the operator load function causes the hardware to read an IPL (initial program loader) program into storage from the system residence volume. For this reason, loading and initializing the system is often called the "IPL procedure" or just "IPL". Likewise, "IPLing" the system means loading and initializing the system.

The IPL program is what actually loads the system software; if the IPL program does not get into storage or does not receive control properly, the entire load process stops and the processor pauses. If the IPL program does not finish properly, it puts the system into a disabled wait state with an error code in the low-order 12 bits of the program status word (PSW). To continue loading the system, display the PSW, note the error code, and follow the instructions for that code given in *z/OS MVS System Codes*. The processor operations manual tells you how to display the PSW.

Logging On to the System

Your installation can control the use of the system commands and access to the MCS and SMCS consoles through the security authorization facility (SAF) and the Resource Access Control Facility (RACF). Your installation can require operators to use the LOGON command to log on to the system and identify themselves.

Your installation can specify the LOGON attribute for MCS and SMCS consoles in two ways. First, a default LOGON attribute can be specified for all consoles active on a system by specifying the LOGON keyword on the DEFAULT statement in the CONSOLxx parmlib member. Second, individual consoles can override the default LOGON attribute by specifying the LOGON keyword on the CONSOLE statement in the CONSOLxx parmlib member. For more information on specifying LOGON consult *z/OS MVS Planning: Operations* and *z/OS MVS Initialization and Tuning Reference*.

Your installation can specify that LOGON is required by specifying LOGON(REQUIRED) on the DEFAULT statement (for all MCS and SMCS consoles) or on the CONSOLE statement (for a single console). When LOGON is a system requirement, you can issue commands only through a master authority console until

RACF is fully initialized and able to process logon requests. Until RACF is initialized, you cannot issue any commands from any non-master authority console.

Once RACF is fully initialized, all operators are required to logon. The message IEE187I prompts you for a userid and password. Optionally, you might enter a group id and a security label. See “LOGON Command” on page 4-301 for more information.

IBM suggests that SMCS consoles be LOGON(REQUIRED), either using the system-wide DEFAULT LOGON specification or the CONSOLE LOGON specification of the console.

Your installation can specify that LOGON is automatic by specifying LOGON(AUTO) on the DEFAULT statement (for all MCS and SMCS consoles) or on the CONSOLE statement (for a single console). When LOGON is not a system requirement, after the security product is fully initialized, the system will automatically issue an MCS LOGON command to each active MCS or SMCS console; system operators may log on to these consoles but are not required to do so. Automatic logon affects only **full capability** consoles.

Your RACF administrator creates RACF user profiles for each operator. Each operator can have access to different commands, consoles, data sets, and other RACF-protected resources, according to the person’s responsibilities. The RACF administrator also creates RACF resource profiles that protect **all** operator commands. If you need more information on creating profiles for operators, consoles, MVS commands, and other resources, see the *z/OS Security Server RACF Security Administrator’s Guide*.

Your installation can specify that LOGON is optional by specifying LOGON(OPTIONAL) on the DEFAULT statement (for all consoles on the system) or on the CONSOLE statement (for a single console). Code the OPTIONAL parameter when your installation has selected consoles defined in RACF to require the operator to log on.

z/OS MVS Planning: Operations has more information about controlling system commands and consoles in a secure environment.

Initializing the System Software

Once the software is loaded into storage, it must be given specific starting values before it can do work. These values are supplied through a LOADxx parmlib member specified by the **LOAD parameter** on the **system control (SYSCTL) frame**, or, depending on the installation hardware level, through the system console or the NIP console during the initialization process.

In certain situations, the system prompts you to specify an alternate master catalog; then it prompts for system parameters that are not specified in LOADxx. The following two sections explain how to respond to those prompts.

Specifying an Alternate Master Catalog

During system initialization, unless the SYSCAT parameter is specified in the LOAD parameter, the system issues the following message:

IEA347A SPECIFY MASTER CATALOG PARAMETER

You must respond to this message. You can respond in one of two ways:

- If your installation uses the default member of SYS1.NUCLEUS, SYSCATLG, to find the master catalog, press the ENTER key.
- If your installation uses an alternate member of SYS1.NUCLEUS, SYSCATnn, to find an alternate master catalog, enter two characters for *nn*.

Specifying System Parameters Not Defined in LOADxx

The LOAD parameter can supply values not defined at system installation time. If this is not done, you must supply them as system parameters in response to the following system message:

```
IEA101A SPECIFY SYSTEM PARAMETERS FOR product-name
```

You must respond to this message. You can respond with specific system parameters, such as

```
REPLY 00,CLPA,SYSP=83,LNK=(04,05,PQ),SYSNAME=AQ
```

However, a typographical error made in this response can lead to undesirable system operation. To help avoid this situation, the system programmer can specify system parameters in IEASYSxx parmlib members. If this has been done, you can respond to message IEA101A in one of the following ways:

- To use the system parameters specified in the IEASYS00 parmlib member, press the ENTER key.
- To use system parameters specified by IEASYSxx parmlib members along with IEASYS00, use the SYSP operand on the REPLY command to specify the 2-character suffixes that identify the IEASYSxx parmlib members.

For example, to use the parameters specified in parmlib members IEASYSAA and IEASYSBB along with IEASYS00, enter:

```
REPLY 00,SYSP=(AA,BB)
```

Note: Depending on the specific system parameter, a parameter value specified in the alternate parmlib members supplements or overrides the value specified in IEASYS00.

If the reply is longer than one line (there are 80 characters per line), you can follow the last parameter with a comma or a blank and CONT. For details on how to continue system parameters, see “Specifying System Parameters” on page 4-360 in the description of the REPLY command in this book.

For details about parmlib members, see *z/OS MVS Initialization and Tuning Reference*.

Setting the Time and Date

If the time-of-day (TOD) clock on the target processor is not set or if your installation specifies the OPERATOR PROMPT parameter in the CLOCKxx member of SYS1.PARMLIB that the system uses for initialization, the system prompts you during initialization to set the correct time and date with message IEA886A and/or message IEA888A. Message IEA886A asks you to specify values for the time and date. Message IEA888A displays the time and date and lets you accept or change these values. In response to either message, set an accurate time and date according to your installation’s requirements.

For example, suppose the system issues:

```
IEA888A UTC DATE=1991.301,CLOCK=22.31.53
*00 IEA888A LOCAL DATE=1991.301,CLOCK=17.31.53 REPLY U, OR UTC/LOCAL TIME
```

The values in this message indicate that the local time is 5:31:53 P.M. on October 28, 1991 and that Coordinated Universal Time (UTC) is five hours later than local time in your time zone. If the local time at your installation is really 8:00:00 A.M. on October 29, 1991, reply as follows:

```
R 00,DATE=1991.302,CLOCK=13.00.00,GMT
```

The system responds with:

```
IEA888A UTC DATE=1991.302,CLOCK=13.00.00  
*00 IEA888A LOCAL DATE=1991.301,CLOCK=08.00.00 REPLY U, OR UTC/LOCAL TIME
```

Note that the system sets the local time but not the local date from the time and date you specify. To set the local date, reply as follows:

```
R 00,DATE=1991.302
```

If the new UTC and local time values are still not accurate enough, you can reply with new UTC time values now (and as many times as you need) to bring the system's values closer to what your installation requires. When you are satisfied with the system's values, reply as follows:

```
R 00,U
```

See "REPLY Command" on page 4-355.

Initializing MCS and SMCS Consoles

Message IEE612I appears on an MCS and SMCS console when it completes initialization.

If you enter the command DISPLAY C,K (or D C,K), the system displays a summary of the CONTROL commands. You can use these commands to change the characteristics of the console. See "Displaying CONTROL Command Functions" on page 4-110 for information about the DISPLAY C,K command.

Starting and Specifying Parameters for the Job Entry Subsystem

Even after the system is initialized, it cannot accept work until the job entry subsystem (JES2 or JES3) is started. The system automatically starts JES2 or JES3 if your installation provides this capability. Otherwise, you must issue the START command. For further information about starting JES, see either *z/OS JES2 Commands* or *z/OS JES3 Commands*. See "START Command" on page 4-575.

Controlling the System

Controlling the operating system effectively, includes the following tasks:

- Display current system status, such as the number of active jobs and teleprocessing functions, so you can take appropriate actions to operate the system efficiently and to correct potential problems
- Display the status of devices and availability of paths.
- Communicate among several consoles.
- Communicate within a sysplex. In a sysplex, several MVS systems function together to process work, and you might need to know about the operations of more than one system in a sysplex.
- Set the time and change the system parameters.
- Use the system restart function to control certain system functions.
- Respond to message IEA502A.
- Respond to message BLW004A.

- Activate a workload management service policy for a sysplex.

MVS provides system and subsystem commands that display job and system status either when requested or continually at a regular interval. Other commands route status information to one or more consoles and provide communication among operators in a multiple-console environment, as well as communication with time-sharing users. Many commands let you display information about all the systems in a sysplex, and some commands allow you to control any target system in the sysplex.

The following sections describe in detail how to control the system.

Displaying Current System Status

Using the **DISPLAY** command, you can display overview information about all current system activity and detailed information about active batch jobs, started tasks, system address spaces, and/or logged-on time-sharing users. (The **DISPLAY** command in Chapter 4 describes the overview and detailed information you can display.) The command produces a one-time display of status as it is at the time you enter the command.

To help you keep up with the system's needs, you can enter the **DISPLAY R** command to display system requests waiting for replies or actions, mount requests not yet fulfilled, and devices waiting for operator intervention. You can use the information in the display to take any necessary actions. See "Displaying System Requests" on page 4-202 for information about the **DISPLAY R** command.

Using the **MONITOR** command, you can keep track of jobs starting and stopping. In response to the **MONITOR** command, the system displays the job identification whenever a job starts or stops. Using this command, you can also request that the system notify you of TSO logons, JCL failures, and data set allocations. See "MONITOR Command" on page 4-343. You can also use the **SETCON MONITOR** command to enable or disable monitoring messages for jobs, TSO/E sessions, and data set allocations. See "SETCON Command" on page 4-403.

Displaying the Status of Devices and Availability of Paths

There are three commands that you can use to display the status of devices and the availability of the paths these devices are on.

The **DISPLAY U** command allows you to keep track of the availability for allocation of the following devices attached to the system:

- Channel-to-channel (CTC) links
- Direct access storage devices (DASDs)
- Graphic devices
- Magnetic tape units
- Communication equipment
- Unit record devices

This command displays device status and the job names and ASIDs of device users. Knowing what jobs and ASIDs are using a particular device allows you to determine whether you can take the device offline. See "Displaying Device Status and Allocation" on page 4-239 for information about the **DISPLAY U** command.

The **DISPLAY M** command allows you to keep track of the availability of channel paths and devices on these paths. See "Displaying System Configuration Information" on page 4-166 for information about the **DISPLAY M** command.

The **DEVSERV PATHS** command can help you solve hardware or configuration problems. The display includes the status of paths, the channel path ids, the logical mode of devices, the number of data sets allocated on volumes, and volume serial labels. Because the DEVSERV command causes the system to issue an I/O request on paths to a device or devices, the resulting display reflects the current physical state of the path. Comparable displays from the DISPLAY M command reflect less recent information from the last use of MVS control blocks. For example, assume that an I/O device is performing below normal and you suspect that some paths to the device are offline. The DISPLAY M command might tell you that there are four paths online to the device. The DEVSERV PATHS command might tell you that there is actually only one online path. The DEVSERV command is more current and thus more accurate. See “**DEVSERV Command**” on page 4-80 for information about the DEVSERV command.

Sending Commands to Systems in a Sysplex

You can use the CONTROL V command to direct commands from a console to a specific system in a sysplex. The CMDSYS parameter on the CONTROL V command specifies which system receives all commands (not specifically routed elsewhere by the ROUTE command) entered from a particular console. See “**CONTROL Command**” on page 4-56.

You can use the ROUTE command to send commands to be processed on other systems in the sysplex. See “**ROUTE Command**” on page 4-367.

You can use the VARY CN command to specify from what systems in a sysplex a specified console receives unsolicited messages. Use the MSCOPE, AMSCOPE, and DMSCOPE parameters for purposes of control. See “**VARY CN command**” on page 4-621.

Some commands have an L=name parameter. You can use this parameter to specify the name of a console on a different system in the sysplex. These commands can communicate with the named console and receive messages from that system.

Using Commands That Have Sysplex Scope

Commands that have sysplex scope have the following characteristics:

- They affect resources that are shared throughout the sysplex. Examples of such resources include the Sysplex Timer, the coupling facility, couple data sets, and certain DASD volumes.
- You can issue them from any system in the sysplex; the results are identical.
- The results of issuing them are sysplex wide ***without the need to use ROUTE *ALL***. You should not use any form of the ROUTE command to issue a command with sysplex scope because doing so is redundant. Here’s why:
 - You use ROUTE to have a command issued on a particular system, group of systems, or all systems in the sysplex. Using the ROUTE command is the logical equivalent of walking up to a console attached to each system you route to, and issuing the command from that console.
 - You do not need to issue a command with sysplex scope on a particular system, group of systems, or all systems in the sysplex. You issue the command once from any system.

Note that a command can have sysplex scope when you use particular parameters, and not have sysplex scope when you use other parameters.

Commands that have sysplex scope are so noted in the documentation for that command, and include those in the following table. If a command has **All** under "Conditions", then the command has sysplex scope under all circumstances and for all variations.

Table 1-2. MVS System Commands With Sysplex Scope

Command	Conditions
CHNGDUMP	Has sysplex scope only when all systems are connected to the same coupling facilities, and you specify ,SDUMP,SYSFAIL,STRLIST=.
CONTROL C,A	All
CONTROL C,D	Has sysplex scope only when you specify L=.
CONTROL M	Has sysplex scope only when you do not specify MLIM, UEXIT, LOGLIM, or APPLID.
CONTROL other	Other parameters of CONTROL have sysplex scope only when you specify L=.
DISPLAY CF	Has sysplex scope only when displaying information about the coupling facility and only for those systems connected to the coupling facility. Does not have sysplex scope when displaying an individual system's coupling facility configuration information (coupling facility channels and paths).
DISPLAY CNGRP	All
DISPLAY CONSOLES	Has sysplex scope unless you specify DISPLAY C,B or DISPLAY C,U.
DISPLAY DUMP	Has sysplex scope only when you issue the OPTIONS parameter to display the results of a CHNGDUMP ...SDUMP,SYSFAIL,STRLIST= command.
DISPLAY EMCS	Has sysplex scope, except when you specify STATUS=B or STATUS=ERR. When you specify STATUS=FULL, consoles from all systems will be displayed (for consoles that are not active on the system where this command is processed, some information will not be displayed).
DISPLAY GRS	Has sysplex scope unless you specify SUSPEND. Also, note the following about DISPLAY GRS,C and DISPLAY GRS,RES: the output generated by these commands includes both system-specific information (S=SYSTEM) and sysplex information (S=SYSTEMS). The S=SYSTEM information is valid only for the system on which you issue the command. The S=SYSTEMS information is identical regardless of the system on which you issue the command.
DISPLAY LOGGER	Has sysplex scope when you use either L or C,SYSPLEX options.
DISPLAY OPDATA	Has sysplex scope for the PREFIX operand and the MONITOR operand (except for SPACE and DSNAME).
DISPLAY PFK	Has sysplex scope only when you specify CN=.

Table 1-2. MVS System Commands With Sysplex Scope (continued)

Command	Conditions
DISPLAY R	Has sysplex scope, but the output might be different on different consoles, because the output of DISPLAY R is dependent on the routing criteria for the console specified by CN=. If you do not specify CN=, the routing criteria of the console issuing the command is used. If you issue the command in a program (by using the MGCRE macro) the console you specify in the macro is used. If you specify a console ID of 0, all retained messages are included in the command response.
DISPLAY WLM	All
DISPLAY XCF,ARMSTATUS	Has sysplex scope provided all systems are using the same ARM couple data set.
DISPLAY XCF,CF	Has sysplex scope provided all systems in the sysplex are connected to the same coupling facilities.
DISPLAY XCF,COUPLE	Has sysplex scope as long as all systems are using the same types of couple data sets, as specified on the TYPE parameter (SYSPLEX, ARM, CFRM, SFM, LOGR, and WLM.) If you do not specify the TYPE parameter, only system-specific data is displayed.
DISPLAY XCF,GROUP	All
DISPLAY XCF,POLICY	Has sysplex scope as long as all systems are using the same types of couple data sets, as specified on the TYPE parameter (ARM, CFRM, SFM, and LOGR.)
DISPLAY XCF,STRUCTURE	Has sysplex scope provided all systems in the sysplex are connected to the same coupling facilities.
DISPLAY XCF,SYSPLEX	All
MONITOR	Has sysplex scope only when you specify L=.
MOUNT	Has sysplex scope only when you issue the command against an automatically switchable tape device.
REPLY	All
RESET CN	Issue the command from the system where the console was active to avoid inconsistent sysplex results.
SEND	Has sysplex scope only when sending to consoles; does not have sysplex scope when sending to TSO users.
SET CNGRP	Has sysplex scope provided all systems are sharing the same parmlib data set.
SET DAE	Has sysplex scope only when all systems are sharing the same DAE data set and the same parmlib data set.
SETLOGR FORCE	Has sysplex scope when you use DELETE,LSName options.
SET GRSRNL	Has sysplex scope only when all systems are sharing the same parmlib data set.
SET SMS	Has sysplex scope when you are issuing the command to change the name of the ACDS or COMMDS. All systems in the sysplex must be in the same SMS complex, and using the same parmlib data set. If you are issuing the command to start or restart SMS on a system, only the system on which you issue the command is affected.

Table 1-2. MVS System Commands With Sysplex Scope (continued)

Command	Conditions
SETSMS	Has sysplex scope only if you are changing the SCDS, ACDS, or COMMDS, and only if all systems in the sysplex are in the same SMS complex.
SETXCF FORCE	Has sysplex scope only when all systems are connected to the same coupling facility.
SETXCF COUPLE	Has sysplex scope only when you specify PSWITCH, ACOUPLE, or PCOUPLE, and all systems have access to the specified couple data set.
SETXCF STARTISTOP	Have sysplex scope only when you specify POLICY or REBUILD.
STOPMN	Has sysplex scope only when you specify L=.
UNLOAD	Has sysplex scope only when you issue the command against an automatically switchable tape device.
VARY CN	Has sysplex scope unless all of the following are true: <ul style="list-style-type: none"> • You issue VARY CN(<i>conspec</i>),ONLINE without specifying SYSTEM=. • You do not specify SYSTEM= in the CONSOLxx parmlib member that defines this console. • The console has never been active in the sysplex.
VARY SMS, STORGRPIVOLUME	Has sysplex scope under these conditions only: <ul style="list-style-type: none"> • You specify (<i>storgrp/volume,ALL</i>) and all systems in the sysplex are in the same SMS complex. • You specify (<i>storgrp/volume system</i>) where <i>system</i> is a system group, and the system group exactly matches the sysplex (that is, none of the systems in the sysplex is explicitly defined to SMS).
VARY XCF	All
VARY WLM	All

Sharing System Commands

MVS allows two or more systems in a multisystem environment to share commands while retaining unique values in those commands. When two or more systems share commands, you can view a multisystem environment as a *single system image* from which you can perform operations for several different systems.

This section explains how to share system commands in a multisystem environment, using:

- **System symbols**, which represent unique values in shared commands
- **Wildcards**, which identify multiple resource names in commands.

Using System Symbols in Commands

System symbols represent the values in shared commands that are unique on different systems. Each system defines its own values to system symbols, and replaces the system symbols with those values when processing shared commands.

To use system symbols in system commands, first see the section that describes system symbols in *z/OS MVS Initialization and Tuning Reference* to understand the types of system symbols, the elements that comprise them, and the general rules

for using them. Second, see the section on sharing system commands in *z/OS MVS Planning: Operations* for information about planning to share commands. Then read the rest of this section.

Display Static System Symbols: You can enter the DISPLAY SYMBOLS command to display the static system symbols and associated substitution texts that are in effect for a system. See "Displaying Static System Symbols" on page 4-235 for more information.

Know the Rules for Using System Symbols: The system enforces the following rules when you use system symbols in system commands. They apply *in addition* to the general rules for system symbols that are described in *z/OS MVS Initialization and Tuning Reference*.

1. Substitution in a command begins after the command name. This means that you cannot use symbolic variables to resolve to a command prefix or to a command name. The command "&Asyspref &mycmd" would result in an error message, for example.
2. If the issuing console has command association (CMDSYS) to another system, the issuing system first transports the command to the associated system. Substitution of any symbolic variables takes place on the receiving system.
3. If a command has a prefix defined with the command prefix facility (CPF), the issuing system first transports the command to the system defined for that prefix. Substitution of any symbolic variables takes place on the receiving system.
4. After echoing and logging a command, the system examines the command name. Certain commands receive special treatment:
 - The system will not perform substitution for symbolics in a VARY CN(*),ACTIVATE command.
 - A DUMPDS command will not undergo substitution. The DUMPDS command processor handles its own substitutions, at the time when it actually takes a dump.
 - For security reasons, the LOGON command does not support symbolic substitution.
 - For a REPLY command, substitution of any symbolic variables in the reply text takes place on the system originally issuing the WTOR.

However, if the WTOR is synchronous (SYNCH = YES was specified, and the synchronous WTO/R service displays the WTOR), the system does not perform substitution of the reply text.

But, if the system issues the WTOR early during the initial program load (IPL), that is, while the nucleus initialization program (NIP) is still in use:

- The system performs substitution after it processes the requested symbolics it reads from the parmlib. This means that the system *will* substitute symbolic variables in replies to WTOs it issues **after** issuing the IEA347A SPECIFY MASTER CATALOG PARAMETER message.
- The system *will not* issue message IEE295I for NIP-time replies that are changed by symbolic substitution. Message IEE600I will reflect the changed text.
- For a ROUTE command, the system issuing the command performs the substitutions up through the specification of the destination system(s). Each destination system completes the substitution of the text for the command.

For example, if you code the command

```
RO T=&T1,&SYSGRP1,F JOB&SYSCLONE,parms
```

the system issuing that ROUTE command will substitute the variables &T1 and &SYSGRP1

and each system in the system group that &SYSGRP1 names will issue the command

```
F JOB&SYSCLONE,parms
```

and each of those receiving systems will substitute its own value for &SYSCLONE. See “Using System Symbols in ROUTE Commands.”

- You cannot use symbolic variables on an “L=” operand to aggregate the command response when sending a command to more than one system. The system will not substitute for the “L=” operand.
 - For commands other than REPLY and ROUTE, the system issuing the command performs the substitution for the text after the command name, including comments.
5. You cannot use system symbols in commands that control batch jobs. Consider converting batch jobs to started tasks, which *can* specify system symbols.
 6. If substitution results in changing any command text, the system logs the “new” text again and issues message IEE295I.

The system makes the original (pre-substitution) command text available to the command installation exits and the subsystem interface (SSI). However, current programs, if not modified, will see the substituted text.

When the system calls the command installation exits or SSI, if those exits make any change to the command text, the system logs them again and issues message IEE295I. However, it does not perform substitution again. It frees the original command text, which means that it is no longer available in the system.

Cautions in Using System Symbols: The preceding rules mean that some forms of command input will probably not produce the results you want:

1. Symbolic variables before or in a command name remain unsubstituted. The system will process the command with the “&variable;” in the text, and probably generate a “COMMAND INVALID” error message.
2. If a command exit changes the text and adds a new symbolic variable, the system executes the command before substituting for the variable.
3. The following considerations apply when a command affects systems other than the one issuing it:
 - Except for REPLY, the substitution will reflect the issuing system. For example, if
`SYSVAR1 = (1,2)`

on the system issuing the following VARY command, but

```
SYSVAR1 = (3,4)
```

on a system with the console “consname” attached, the command

```
VARY CN(consname),ROUT=&SYSVAR1
```

would result in the console “consname” receiving codes 1 and 2. If this (unlikely) command is what you want, you should ROUTE it to the system with consname attached.

- The same logic applies to commands that accept the “L=name-a” parameter, that is, where you want the command output messages directed to a console (and display area) other than the one issuing the commands. Substitution of

symbolic variables in commands occur on the systems where the commands are issued, not where the “L=” console is attached.

- Do not use symbolic variables in the “L=” parameter on the ROUTE command. See the ROUTE command description in this manual.
- Understand the implications of using system symbols in commands that flow through several systems in a multisystem environment. See “Sharing Commands That Flow Through Multiple Systems” in *z/OS MVS Planning: Operations* for more information.

Determine Where to Use System Symbols: System symbols offer the greatest advantage when two or more systems require different resources. This section provides examples of how to specify system symbols when naming resources in system commands.

Data Sets:

Assume that you want to display, on all systems in a sysplex, the local page data sets that fit the following naming convention:

SY&SYSCLONE..PAGE.LOCAL

Instead of entering a different command to display the unique page data sets on each system, you could enter the following command to display all the data sets that fit the naming convention:

ROUTE *ALL,D ASM,PAGE=SY&SYSCLONE..PAGE.LOCAL

When each system processes the command, it substitutes the text that it has defined for the &SYSCLONE system symbol. For example, if a sysplex consists of two systems named SYS1 and SYS2, accepting the default value for &SYSCLONE produces the following data sets:

D ASM,PAGE=SYS1.PAGE.LOCAL on system SYS1
D ASM,PAGE=SYS2.PAGE.LOCAL on system SYS2

Jobs:

When specifying system symbols in the source JCL for job names, first determine if the jobs run as batch jobs or started tasks. If a job is a started task, you can specify system symbols in the source JCL. If a job runs in batch, you cannot specify system symbols in the source JCL; consider changing the job to run as a started task.

Then, if a started task is to have multiple instances, determine if you want the started task to have a different name for each instance. If each instance of a task has a different name, your installation can easily identify the system on which each instance runs.

For started tasks, you can also specify system symbols on the JOBNAME parameter on the START command that starts the task. For more information about using system symbols in START commands, see “START Command” on page 4-575.

Using Wildcards in Commands

Wildcards allow you to use a single specification to indicate a number of resources whose names match the wildcard pattern.

System commands use three kinds of wildcards:

- **Multiple-character trailing asterisk (*):** The * indicates zero, one, or more characters, up to the maximum length of the string. This * must be at the end and cannot appear alone. For example, ABC* matches ABC or ABCVWXYZ or ABC1 or ABCZZZ. Use this wildcard in:
 - CANCEL
 - DISPLAY
 - MODIFY
 - SETPROG
 - SLIP parameters, as indicated in their descriptions
 - STOP
- **Multiple-character asterisk (*) within the value:** The * indicates zero, one, or more characters, up to the maximum length of the string. This * can be in any position and can appear alone to indicate all values. For example:
 - A*BC matches ABC or ACBC or AWXYZBC or A3BC
 - * matches all values
 - *BC matches BC or WXYZBC or ZZZBC

Use this wildcard in the JOBLIST and DSPNAME parameters of the SLIP command.

- **Single-character question mark (?):** The ? indicates any single character. The ? can be in any position. For example:
 - A?C matches ABC or A1C
 - ABC?E?? matches ABCXEYZ or ABC1E23
 - ?BC matches ABC and ZBC

Use this wildcard in SLIP parameters, as indicated in their descriptions.

In some SLIP command parameters, you can use more than one type of wildcard. For example:

- A?C* matches ABC or AXCYZ or A5CZ2
- A*C? matches ABCD or AZZZC1 or A123CZ or ACD

You can use wild cards to reduce the number of system commands needed for a task. For example, you can enter one command to display information about all jobs and started tasks beginning with the characters XYZ:

DISPLAY A,XYZ*

Setting the Time and Changing the System Parameters

Using the SET command, you can set the local time and date and change some system parameters. See “SET Command” on page 4-382.

Using the System Restart Function

You can use the system restart function to:

- Restart the system after you have entered a QUIESCE command. (See “Quiescing the System” on page 1-39.)
- Restart the system from a restartable wait state that is specified in *z/OS MVS System Codes*.
- Restart the system when it behaves abnormally and you cannot terminate the suspected unit of work with the CANCEL or FORCE commands. A system behaving abnormally may be one that enters a nonvalid wait state or a disabled loop. A nonvalid wait state exists when the wait state code in the PSW (IC) is not listed in *z/OS MVS System Codes* and is not in the range of wait state codes (FF0-FFE) reserved for other authorized applications. Symptoms of a disabled loop are:

- Nonproductive processing occurs and the PSW (IC) frequently displays the same addresses.
- All I/O and external interrupts are masked off for the system.

System Restart Procedure

To initiate the system restart function press the RESTART key on the hardware operator's console or specify one of several restart actions on an operator frame. Refer to the hardware manual for your system for more detailed information about your configuration. If the system has been quiesced or is in a valid restartable wait state, the system restarts and continues processing the interrupted unit of work. If the system had not been quiesced or is not in a valid restartable wait state then, depending upon your system configuration, the system displays either message IEA502A or BLW004A.

If the system does not recover as a result of your restart actions, follow your installation's procedures for recording system problems. When you have recorded the system information, consult with your system programmer before taking further action.

Responding To IEA502A

Reply reason code '0' when you suspect that a unit of work is causing a wait state that is not valid or a disabled loop and you cannot terminate the suspected unit of work by using the CANCEL or FORCE commands.

1. The system displays message IEA500A and waits for operator response. IEA500A supplies information about the unit of work in progress.
2. Reply ABEND to abnormally terminate the interrupted program and invoke the necessary recovery routines if the information describes the unit of work you suspect has a problem.
3. Reply RESUME to end further restart processing and allow the interrupted work to continue if the information does not describe the unit of work that you suspect has a problem.

Repeat this process of invoking restart with REASON 0 until you interrupt the work you suspect. Only then should you reply ABEND to abnormally terminate the current work.

Note: The system terminates the work in progress without displaying any information about it if you request the restart function with REASON 0 on a processor that cannot communicate with an operator.

Reply reason code '1' when you suspect a system problem that is not related to the work currently in progress. The system diagnoses and repairs some problems that might be causing it to behave abnormally. Among its actions, the system:

- Makes itself dispatchable.
- Checks the number of message buffers. The system notifies you if the maximum number of buffers has been exceeded.
- Checks system activity. The system notifies you if there are no batch jobs or time-sharing users.
- Restarts I/O on all channel paths.
- Checks and repairs critical data areas.

Note: Using reason code '1' might cause the system to immediately terminate some address spaces. Use reason code '1' only under the direction of a system programmer.

Normally, the system notifies you of anything it diagnoses or repairs when you request the restart function with reason code '1'. You only get this information on a processor that can communicate with an operator.

Responding To BLW004A

The system displays message BLW004A and waits for operator response. BLW004A supplies information about the unit of work in progress.

1. Reply ABEND to abnormally terminate the interrupted program and invoke the necessary recovery routines if the information describes the unit of work you suspect has a problem.
Repeat this process of invoking restart procedure replying to BLW004A until you interrupt the work that has the problem.
2. Reply RESUME to end further restart processing and allow the interrupted work to continue if the message indicates that there are no batch jobs or time-sharing users.
3. Reply RESUME to end further restart processing and allow the interrupted work to continue if the message indicates that the WTO buffer limit has been exceeded.
4. Reply REPAIR if you suspect a system problem that is not related to the work currently in progress. The system diagnoses and repairs some problems that might be causing the abnormal behavior.

Note: Replying REPAIR might cause the system to immediately terminate some address spaces. Reply REPAIR only at the direction of the system programmer.

Activating a Workload Management Service Policy

Important

Beginning with z/OS V1R3, WLM compatibility mode is no longer available. Accordingly, the information below that pertains specifically to WLM compatibility mode is no longer valid. It has been left here for reference purposes, and for use on backlevel systems.

You can use the VARY WLM command to activate a named service policy for a sysplex. The service policy must be defined in the workload management service definition and must have been previously installed on the WLM couple data set.

You can also activate a workload management service policy by using the online ISPF administrative application. Refer to *z/OS MVS Planning: Workload Management* for more information or see your service administrator.

This command activates the named service policy on all systems in the sysplex, regardless of the workload management mode in effect. However, only systems operating in workload management goal mode will manage towards that service policy. If there is an active service policy on a system running in compatibility mode, and you use the MODIFY command to switch that system into goal mode, workload management uses the service policy you activated.

For complete information on how to use the VARY command to activate a workload management service policy, see “Activating a Service Policy” on page 4-662.

Switching Workload Management Modes

Important

Beginning with z/OS V1R3, WLM compatibility mode is no longer available. Accordingly, you can no longer switch from one mode to another, as described below. The information has been left here for reference purposes, and for use on backlevel systems.

You can use the MODIFY WLM command to switch the workload management mode in effect on a system. This command switches the mode of the system where you issue the command.

Before switching into goal mode on any system in the sysplex, your service definition should be complete and installed in the workload management couple data set, and a service policy should have been activated.

For complete information on how to use the VARY command to activate a workload management service policy, see “Activating a Service Policy” on page 4-662.

Controlling Time-Sharing

Time-sharing allows programmers at remote terminals to develop, test, and execute programs without the turnaround delays that occur when they submit jobs to a computer center. With time-sharing, a large number of jobs can share the resources of a system concurrently, and remote terminal users can exercise primary control over the execution of their jobs. Therefore, we can define time-sharing as the shared, conversational, and concurrent use of a computing system by a number of users at remote terminals.

Time-sharing in z/OS is provided by TSO/E. For more information about TSO/E see *z/OS TSO/E User’s Guide*.

You can display information about logged-on time-sharing users by using the DISPLAY command. You can keep track of terminal users logging on and off the system by using the MONITOR command. In response to the MONITOR command, the system displays the user id for each LOGON and LOGOFF. To stop the system’s monitoring of terminal use, issue the STOPMN command.

To communicate with time-sharing users you can use the SEND command to:

- Send messages to specific users or all users who are receiving messages
- Send messages to specific users or to all users logging on to the system
- Save messages in the broadcast data set
- List messages in the broadcast data set
- Delete messages from the broadcast data set

The broadcast data set, SYS1.BRODCAST, has mail and notices sections.

Controlling Jobs

A job is the basic unit of work for the system. Job control language (JCL) identifies a job to an operating system and describes the job's resource requirements. The JOB JCL statement identifies a job's beginning and contains such information as:

- Job name
- Job account number
- Job class
- Job priority.

Using job-related commands, you can start, stop, or cancel a job. You can also modify a job's parameters and performance group and restart a job that has failed. There are two kinds of jobs in the system: queued jobs and jobs that are selected on demand. Queued jobs are managed by JES. Jobs that are selected on demand (referred to as demand-selected) are created as the result of START, MOUNT, and LOGON commands.

Starting a Job

Using the START command, you can start jobs from the console. You can also use the START command to cause the JES internal reader facility to read a job from a tape or direct access volume.

Stopping a Job

Using the STOP command, you can stop a job if the programmer has coded a stop routine in the program.

Cancelling a Job

Using the CANCEL and FORCE commands, you can cancel a job that is executing. If the job is not currently executing, use a subsystem command to cancel it.

Passing Information to a Job

Use the MODIFY command to pass information to a job. This information may be used by the currently running program. Note that you can only pass information that is already defined in the currently running program.

Note to Programmers: For more information, see the section on communicating with a program using EXTRACT or QEDIT in *z/OS MVS Programming: Authorized Assembler Services Guide*.

Restarting a Job

Once a job is executing, it might end abnormally because of a hardware, programming, or system error. This might happen any time during program execution. Valuable machine time would be lost if an abnormal end occurred during one of the last job steps of a multistep program or in the middle of a long job step, and execution had to start again at the first job step. There are two ways of avoiding this problem: automatic restart and deferred restart.

For JES2 jobs and JES3 jobs, the checkpoint/restart feature of the system allows a job that ends abnormally to restart either at the beginning of a job step or at a checkpoint within the current step. The programmer submitting the job provides for an automatic restart or a deferred restart.

Automatic Restart

If the programmer submitting the job has provided for an automatic restart and the job ends abnormally, you receive the following system message:

```
* id IEF225D SHOULD jobname.stepname.procedure checkid RESTART
```

This message allows you to prevent repeated restarts at the same checkpoint or job step.

When this message appears, use the REPLY command to respond YES, HOLD, or NO, as follows:

- Reply YES if the restart is to be performed at a specific checkpoint or job step for the first time. (If it is a job step restart and the step to be restarted used a card input data set that was not part of the SYSIN stream, you must return to the appropriate hoppers all cards read by the job step before it ended abnormally. If it is a checkpoint restart, follow the programmer's instructions for replacing the input cards.)
- Reply HOLD if you want to defer the restart: for example, to permit another job to run first. You must issue the appropriate subsystem command when you are ready to restart the job. Also, if you want, you can cancel the job. However, cancelling the job can cause unrecoverable paging space or the failure of certain data sets to be deleted if the job was using virtual I/O.
- Reply NO if a restart at a specific checkpoint or job step has been requested repeatedly. When your reply is NO, and the programmer wants a restart to be performed, he must resubmit the job for a deferred restart.

If the programmer specifies VIRTUAL=REAL (V=R), the job is processed entirely in central storage; it is not paged out. For a V=R job, the restart might be delayed while the system waits for the allocation of storage. If another job is using the required storage, you get no message, only a delay. Enter the DISPLAY A,L command to see if a system task or another job is using the storage required by the job with a V=R region. You can then stop or cancel the conflicting task or job.

Note: Any operator commands in the input stream of the job step being restarted are not executed.

Deferred Restart

If the programmer submitting the job has provided for a deferred restart and the job ends abnormally, the programmer must resubmit the job for the deferred restart. To restart the job, the programmer must provide a restart deck for submission to the system through the system input reader. The JCL statements to be included in the restart deck are described in detail in *z/OS MVS JCL User's Guide*.

If you change the device configuration of your system after a job ends abnormally, restart the job carefully. For example, enough devices must be available to satisfy the needs of the job step being restarted. The system under which a step restart is run need not be the same as it was for the job's original execution. However, a checkpoint restart should be run under the original system unless the alternate system can meet the following restrictions:

- The job entry subsystem is the same.
- The release number is the same.
- The link pack area modules in use at the checkpoint reside in the same storage locations.
- An area of storage identical to the original area is available to a V=R job.

If the required storage is not available, the system cancels the restart and you receive the following message:

IEF209I VIRTUAL STORAGE UNAVAILABLE FOR jobname.stepname.procedure

Required storage might not be available for one of the following reasons:

- The link pack area expands into the required storage. This expansion can occur if an IPL has been performed between the original execution of the job and the restart. If it does occur, contact your system programmer for a respecification of the system parameters and reIPL using the new values.
- The system storage area expands into the required storage. When this expansion occurs, contact your system programmer for a respecification of the SQA and CSA system parameter and reIPL using the new values.

When a job restarts correctly, you receive two messages: IEF006I JOB RESTARTING and IEF008I JOB RESTARTED. If, for V=R jobs, these messages do not appear, enter DISPLAY A,L to see if a system task or other job is using the required storage. You can then stop or cancel the conflicting job. The system might ask you to mount data volumes other than those required at the beginning of the job. In addition, any card input data sets that have been used by the failing job step must again be made available to the system.

For more information on deferred restart, see *z/OS DFMSdfp Checkpoint/Restart*.

Controlling Started Tasks

A started task, like a job, is a basic unit of work for the system. However, started tasks differ from jobs in that started tasks are always demand-selected; that is, the operator or a program must take action to initiate a started task.

There are several ways to initiate started tasks:

- With a START command, described in Chapter 4 of this book.
- Via TSO/E logons. For information on using TSO/E logons, refer to the TSO/E publications.
- With ASCRE (address space create) macros in programs. For information on how to use the ASCRE macro, refer to *z/OS MVS Programming: Extended Addressability Guide*.

Both the START command and the ASCRE macro create an address space. A START command and an ASCRE macro started via a START command each will look for a program that has a procedure in SYS1.PROCLIB; that program will be the first to run in the ASCRE-created address space. Essentially, using ASCRE is similar to a started task.

For a started task, the system:

- Locates the JCL that starts the task
- Defines the task's address space
- Processes the JCL.

For a started task, operators can do the following:

Task	For information, refer to:
Cancel the started task	"CANCEL Command" on page 4-19
Display status about the started task	"DISPLAY Command" on page 4-97

Task	For information, refer to:
Force the started task	"FORCE Command" on page 4-290
Modify the started task	"MODIFY Command" on page 4-309
Name the started task	"START Command" "Starting a System Task from a Console" on page 4-575 (JOBNAME= parameter)
Start the started task	"START Command" "Starting a System Task from a Console" on page 4-575
Stop the started task	"STOP Command" on page 4-590

Controlling System Information Recording

The system records information that is later used for billing, accounting, or diagnostics. Among the facilities that record system information are:

- System management facilities (SMF)
- System trace
- The generalized trace facility (GTF)
- Master trace
- Component trace
- The logrec recording medium

The system also records information in the system log and/or the operations log. See *z/OS MVS Planning: Operations* for more information.

In addition to these facilities, JES2 and JES3 have their own event trace facilities. These trace facilities are described in detail in *z/OS JES2 Commands* and *z/OS JES3 Commands*.

System Management Facilities

System management facilities (SMF) consists of system routines and optional user-written exit routines that collect, format, and record system and job-related information.

The information gathered by SMF and user-written exit routines is recorded on direct access volumes in one of the SMF data sets. These data sets, called primary and secondary data sets, must be online at system initialization. At that time, SMF uses the primary data set as the active recording data set unless it is full. If the primary data set is full, SMF checks each data set in the order it is listed until it finds one that is not full. SMF then uses this data set as the active recording data set and requests that the operator dump all data sets that are not empty.

When the active recording data set becomes full, SMF automatically switches recording from the active SMF data set to an empty secondary SMF data set, passes control to the SMF dump exit, IEFU29, and issues a message to indicate that the data set needs to be dumped. Use the SMF dump program, IFASMFDP, to dump the full SMF data set and to reset the status of the dumped data set to empty so that it can be used again for recording.

Error Recovery

If an I/O error occurs while SMF is writing to one of the SMF data sets, you receive a message and SMF switches to one of the empty secondary data sets.

Switching the SMF Data Sets

To prepare an SMF data set for dumping *before* it becomes full, the operator normally uses the SWITCH SMF command to switch from the current data set to another data set. For the switch to be successful, there must be an inactive data set that is empty. Therefore, use the DISPLAY SMF command to verify that there is at least one alternate data set before issuing the SWITCH or HALT command.

The HALT EOD command also prepares an SMF data set for dumping, but use it only when you intend to quiesce the system in preparation to shut down. Do not use HALT when you intend to keep the system running. HALT EOD will close the system log and stop SMF recording.

Restarting SMF

Because SMF runs in its own address space, you can restart SMF with the SET SMF command. When you enter that command, this message appears:

IEE980I SMF IS BEING RESTARTED

When the restart is complete and recording starts, the following message appears:

IEE360I SMF NOW RECORDING ON SYS1.MANx

If the SET SMF command abends while updating the SMF parameters, it might be necessary to terminate the SMF address space and restart SMF. If the system programmer determines that it is necessary to terminate the address space, issue: FORCE SMF,ARM

To restart SMF after the SMF address space terminates, issue the SET SMF command again, specifying a SMFPRMxx parmlib member containing different parameters.

System Trace

System trace is a part of the operating system that records, for diagnostic purposes, events that occur during system initialization and operation. To record events, system trace provides three types of tracing: address space, branch, and explicit tracing. System trace can be used between subsystem initialization and the start of the generalized trace facility (GTF). For information on controlling system trace, see “TRACE Command” on page 4-603.

The Generalized Trace Facility

The generalized trace facility (GTF), like system trace, gathers information used to determine and diagnose problems that occur during system operation. Unlike system trace, however, GTF can be tailored to record very specific system and user program events. For information about starting and stopping GTF, see “START Command” on page 4-575 and “STOP Command” on page 4-590. For information about using GTF, see *z/OS MVS Diagnosis: Tools and Service Aids*.

Master Trace

Master trace is a diagnostic aid that maintains a trace table of console messages in virtual storage. When master trace is active, the master trace table is embedded in dumps that have the TRT option or contain the master scheduler’s private address space. Master trace can eliminate the need to submit a portion of the system log to IBM if there are problems in message processing. It also can ensure that the messages accompanying a dump are the ones that correspond to the problem. The

TRACE command controls master trace. For a more detailed description of master trace, see *z/OS MVS Diagnosis: Tools and Service Aids*.

Component Trace

Component trace is a diagnostic aid that system programmers can use to trace the action of certain system components. Component trace enables the programmer to use the TRACE command to start and stop component trace. The components that use the component trace command must first invoke the define component trace service and define the name of the component requesting the service and the name of the start/stop routine that will get control when the TRACE operator command is issued.

Logrec Recording Medium

When an error occurs, the system records information about the error in either the logrec data set or a sysplex-wide logrec log stream. The diagnostic information provides a history of all hardware failures, selected software errors, and selected system conditions.

Use the records in the logrec data set or the logrec log stream as a companion to dump data. The information in the records will point the system programmer in the right direction while supplying symptom data about the failure.

For more information about log streams, see *z/OS MVS Programming: Assembler Services Guide*. For more information about initializing a logrec data set or setting up a logrec log stream, see *z/OS MVS Diagnosis: Tools and Service Aids*.

Controlling Automatic Tape Switching

In a sysplex, there are MVS operational considerations for two types of tape devices:

- A **dedicated tape device** is varied online to one system at a time. For a second system to use that same device, an operator issues VARY commands (first VARY OFFLINE, then VARY ONLINE) to make the device available to the second system.
- An **automatically switchable tape device** can be online to more than one system at a time. For one system to use an automatically switchable tape device, then another system to use the same device, an operator does not have to issue any VARY commands. In many ways, automatically switchable tape devices are similar to JES3-managed devices. They require that the systems in the sysplex communicate with each other.

Through system commands, the operations staff plays a key role in setting up and maintaining automatic tape switching (that is, using automatically switchable tape devices). For example, a device is automatically switchable after the following operational actions are taken:

1. The device is defined as automatically switchable.
The VARY AUTOSWITCH command, as described in “Defining Automatically Switchable Devices” on page 1-28, turns the AUTOSWITCH attribute on and off.
2. The device is varied online through the VARY ONLINE command.
3. Before z/OS Release 2, the connection between participating systems and the coupling facility is active and an IEFAUTOS structure is defined in the active coupling facility resource management (CFRM) policy.

Systems in a sysplex stored the status of online automatically switchable tape devices in IEFAUTOS.

With z/OS Release 2 and higher, the ATS STAR design improves the availability and system management characteristics of the previous automatic tape switching function. ATS STAR does not use the IEFAUTOS structure but instead uses global resource serialization and XCF services to maintain serialization when it allocates shared tape devices. To maximize the performance of the ATS STAR function, use the global resource serialization STAR configuration rather than the ring configuration.

Defining Automatically Switchable Devices

To define a device as automatically switchable, the device must be in a varied-offline state. Use the following command:

```
VARY device,AUTOSWITCH,ON
```

The detailed description of this command is in “Defining a Tape Device as Automatically Switchable” on page 4-632.

The AUTOSWITCH definition lasts for the duration of the IPL. Only if the device has been defined through HCD does the definition persist longer than the duration of the IPL. If HCD turns the attribute on, and the VARY AS command turns the attribute off, the attribute will be on again at the next rel IPL.

The ESCON manager and the IEEVARYD programmable interface can also set the AUTOSWITCH attribute on and off.

Displaying Information About Automatically Switchable Devices

The DISPLAY U,,AUTOSWITCH command summarizes the status of automatically switchable devices. The display includes the following information:

- The name of the system to which the device is allocated
- The name of the job
- Volume serial number, if one is mounted and the device is allocated.

If a device is offline to the issuing system, the display shows “OFFLINE” in the status field and the display provides no other information about the device.

The following example shows information that appears in response to DISPLAY U,,AUTOSWITCH. Ten devices are defined automatically switchable in the sysplex. Four of those devices (identified by “A” in STATUS column) are allocated to jobs running on SYS5 and SYS6; two of the devices (identified by “OFFLINE” in the STATUS column) are varied offline to the issuing system; and the status of the other four devices is not known.

- d u,,as							*E	SYS=ALLOC5
UNIT	TYPE	STATUS	SYSTEM	FRAME	LAST	F		VOLSTATE
05A8	348S	A	SYS5	TAPE02	0012			PUB/REMOV
05A9	348S	A	SYS5	TAPE02	0012			PUB/REMOV
05AA	3480	OFFLINE						
05AB	3480						/REMOV	
05AC	3480						/REMOV	
05B8	349S	A -CA	SYS6	TAPE01	012E			PUB/REMOV
05B9	349S	A	SYS6	TAPE01	012E			PUB/REMOV
05BA	3490	OFFLINE						
05BB	3490						/REMOV	
05BC	3490						/REMOV	

Figure 1-2. Example of a Successful Response to a DISPLAY AUTOSWITCH Command

The syntax of the DISPLAY U,,AUTOSWITCH command is in “Displaying Device Status and Allocation” on page 4-239.

If you want to find out the status of a device that is assigned to a nonparticipating system, issue the DISPLAY U,,, command on each system that could have varied the device online, including the participating systems.

Interacting with System Functions

Most resource allocation, error recovery, and system monitoring functions in MVS are automatic. Sometimes, however, the system requests your assistance, takes certain actions that you must understand and/or correct, or issues messages that make you aware of internal processing. So that you can plan your actions carefully and respond appropriately to system messages, you need to know how to interact with the following system functions:

- Device allocation
- Hot I/O detection
- Device boxing

Device Allocation

Device allocation is the assignment of input/output devices and volumes to job steps. Requests for device allocation come from data definition (DD) statements and dynamic device allocation requests.

Data definition (DD) statements can be entered into the system by:

- Job input to the JES reader
- Jobs submitted through the TSO SUBMIT command
- Started tasks
- The MOUNT command
- TSO LOGONs
- APPC transactions

Dynamic device allocation/unallocation requests, in contrast, originate within executing programs.

While performing device allocations, the system might ask you to:

- Mount or dismount volumes
- Make decisions (for example, to bring a device online immediately or to wait)

To control the amount of work you have to do related to device allocation, you might want to restrict device allocation requests.

To control device allocation requests from data definition (DD) statements, you might restrict each of the forms of input for these statements (for example, by holding the reader, or by setting a maximum LOGON count). Because they originate within executing programs, however, you cannot control dynamic device allocation/unallocation requests.

Device Assignment

Operationally, the assignment of devices is influenced by:

- The online/offline status of the device. Generally, to be allocated to job steps, devices must be online. Exceptions are (1) when the online test executive program (OLTEP) or a similar testing program is running and (2) when teleprocessing devices are allocated. You can bring offline devices online with the VARY command or in response to the allocation recovery message, IEF238D.
- The MOUNT attribute. The MOUNT attribute, which applies only to tape or DASD devices, is influenced by the MOUNT and UNLOAD system commands, and, during initialization, by entries in the VATLSTxx parmlib member. Allocation requests that can be satisfied by mounted devices are processed quickly and without your intervention.
- The USE attribute. A parameter of the MOUNT command, the USE attribute affects the type of data sets that can be allocated on a tape or DASD volume. The USE attribute can also be set during initialization by entries in the VATLSTxx member of parmlib. Having a proper mix of volumes with various USE attributes reduces the amount of volume mounting.

The information from data definition (DD) statements determines the input/output resources to assign to a job or job step and the volumes that are required. If a requested volume is not mounted, the system issues a mount message asking you to mount a specific volume or scratch volume. If you mount the wrong volume, the system finds out as soon as it reads the volume label. The system unloads the volume and repeats the mount message.

When you know that several jobs are going to need a volume, use the MOUNT command to reserve that volume on a device. Allocation processing is faster when the required volume is reserved rather than removable. The system does not demount volumes reserved by a MOUNT command until you issue an UNLOAD command.

Note: Do not use the MOUNT command for devices managed by JES3. See *z/OS JES3 Commands*.

Never mount a blank tape volume unless specifically directed to do so because the system scans the entire volume for a tape label and this scanning wastes time. If an unlabeled tape is needed, write a tapemark to avoid unnecessary scanning. After you mount the tape volume and ready the drive, the system reads the volume label. If an incorrect volume is mounted, the system unloads the incorrect volume and repeats the mounting message.

Notes:

1. Occasionally, you receive two mount messages for the same volume, one starting with IEF and the other with IEC. Treat the two messages as though they were one. The second is a reminder.

2. When referring to I/O devices in the *devnum* parameter of system commands, use the unique 3-digit or 4-digit device number for each device. You can precede the device number with a slash (/). The slash is optional on many commands, but required for 4-digit device numbers on some commands, such as MOUNT and START.
3. Your installation can define symbolic group names of one to eight characters to be used by programmers in data definition (DD) statements. The number of devices associated with a symbolic name can range from one to the total number of devices in your installation. The symbolic name allows the devices to be grouped according to the attributes your installation considers significant. Do not use these symbolic names in system commands.
4. Make sure there are sufficient work volumes available to satisfy requests for temporary data sets at peak loads. A shortage of work volumes can cause the system to request additional scratch volumes. Balance work volumes across channel paths to increase system efficiency.

Automatic Volume Recognition

Automatic volume recognition (AVR) allows you to mount labeled volumes on unused drives not managed by JES3. The system recognizes and remembers these volumes, and assigns the drives to later job steps as required.

Hot I/O Detection

Hot I/O refers to the repeated I/O interruptions that result from hardware malfunctions. Because it can cause the system to loop or to fill the system queue area with I/O control blocks, hot I/O needs to be detected quickly and corrected.

When the number of repeated interruptions exceeds an installation-defined threshold value, the system assumes there is a hot I/O condition. If your installation has set up hot I/O recovery defaults that the system can use, the system issues message IOS109E and attempts to recover from the hot I/O condition. (See the IECIOSxx parmlib member in *z/OS MVS Initialization and Tuning Reference*.) If your installation has not set up hot I/O recovery defaults, the system issues one of the following messages, if possible, or loads one of the following restartable wait states and prompts you to take action:

IOS118A or IOS111D – HOT NON-RESERVED DIRECT ACCESS DEVICE
 (Wait state 111)
 IOS119A or IOS112D – HOT RESERVED DIRECT ACCESS DEVICE
 (Wait state 112)
 IOS117A or IOS110D – HOT NON-DIRECT ACCESS DEVICE
 (Wait state 110)

When you take action, try to solve the problem at the lowest possible level. That is, try to correct the problem at the device first and then at the control unit. You could power the device off and on. If that does not help, you could reset the control unit if the affected device is not a direct access device. If these actions do not correct the problem, you might have to physically disconnect the device or control unit.

Whatever action you take, tell the system what you are doing by responding to the prompting message or restartable wait state. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *z/OS MVS System Messages, Vol 9 (IGF-IWM)* for information about IOS messages, and *z/OS MVS System Codes* for a detailed explanation of the restartable wait states and your response to them.

| “Hot I/O” on page 1-51 describes how z/OS handles a hot I/O condition.

Device Boxing

In certain error recovery situations and in response to certain commands, the MVS system can “box” an I/O device. Once a device enters a boxed state, the system:

- Immediately terminates I/O in progress on the device
- Rejects future I/O requests (by a user or by the system) to the device as permanent I/O errors
- Rejects any attempts to allocate the device
- Puts the device in pending-offline status

The system boxes a device:

- When it detects hot I/O on the device and the device cannot be recovered
- When, because of a channel path error, it takes the last path to the device offline
- When, because of a channel path error, it releases a reserve or assign on the device
- When it releases an unconditional reserve for the device
- When you issue a VARY OFFLINE command with the FORCE option for the device
- When you issue a CONFIG OFFLINE command with the FORCE option for a channel path, and the command releases a hardware reserve or assign, or removes the last path to the device

Notes:

1. Because you might release a reserve or assign on a device and cause a data integrity exposure, be sure to use the VARY OFFLINE and CONFIG OFFLINE commands with FORCE only in emergency situations.
2. When you fix whatever caused the system to box a device, you can take the device out of the boxed state at any time by issuing VARY device ONLINE. Once the VARY command takes effect, the device is again available for IOS and any subsequent allocations (i.e., an allocation done in another step or job, or another dynamic allocation). Note that after the VARY command takes effect, the device is not considered for the current allocation.
You can make a boxed alias unit control block (UCB) of a parallel access volume available using the DEVSERV, QPAVS command.
3. You cannot take a boxed device out of the boxed state by replying with the device name to the allocation recovery message, IEF238D.

Boxed Device - Operator Actions

Device *boxing* is used by the MVS system during error recovery as a means of maintaining data integrity and preventing data corruption. A device is also boxed if the operator issues the VARY devnum,OFFLINE,FORCE command. When a device is boxed, all outstanding I/O operations for the device are ended with *permanent error* status, and no new allocations to the device are allowed.

It is very important to understand that in the case of shared DASD, the boxed device is boxed only to the system that originated the boxing. The device is still accessible from other systems. **This may lead to incorrect (or incomplete) data on the DASD volume.** Such a situation must be reported to the owner of the data on the boxed-DASD.

- If the data-files are shared with other systems, it is recommended to put the device in offline status on all the sharing systems. Use VARY OFFLINE or OFFLINE,FORCE commands.

- After the data sets are checked and recovered, the DASD volume may be put back online.

A device that is boxed *and offline* can be brought back online with the VARY devnum,ONLINE command. This will enable the UCW and perform online processing to the device. Assuming that the error condition has been resolved, the device will come online. If the error condition still exists, the device may remain in the boxed state.

A device that is *allocated boxed* may be brought back online with the VARY devnum,ONLINE,UNCOND command, if account procedures allow. Note that in this case, if the boxed device is DASD, volume verification (that is, VOLSER checking) is **not** performed. In this case, the VOLSER information can be obtained by entering a VARY devnum,ONLINE command to the DASD device or then entering a MOUNT command.

A DASD device that was offline (either boxed or not boxed) has the VOLSER details obtained from the device through the VARY devnum,ONLINE command. The VOLSER information is placed in the UCB as part of the vary online operation, if the vary online is successful, that is, that no out-of-line situations exist, for example, it is not a duplicate volume.

Recovery for a Failing Alias Unit Control Block (UCB)

For a parallel access volume, the status of a base UCB affects the status of its alias UCBs. The status of an alias UCB, however, might not affect the status of its base UCB. When the system detects a problem with an alias UCB, a recovery action applies to the base and its alias UCBs. If an alias UCB becomes boxed, you can unbox the alias with the following DEVSEERV command:

```
DEVSEERV QPAVS,devn,UNBOX
```

Error messages that display in the following situations are the only indication that an alias UCB is boxed:

- The device is varied online
- A hardware change is activated
- The system is in recovery

Boxed Tapes under Tape Management System Control

If a boxed tape drive is controlled by a tape management system, the drive will remain in the A-BOX state. Unless the tape management system is taken down, the VARY devnum,ONLINE,UNCOND command must be used to return the tape drive to the online state.

Tape Boxed Due to Lost Assign

If tape CHPIDs, control unit, switches, or ESCON connections are incorrectly handled, a tape ASSIGN may be reset (lost). When this occurs on the last path to the tape drive, the MVS system will box the device.

If an ASSIGN lost condition occurs while a tape was loaded, the MVS system may not be able to unload the tape. If this happens, as indicated by the cartridge remaining in the drive after the job has completed, perform the following actions at the tape unit:

- Place the READY/NOT READY switch to the NOT READY position.
- Toggle the UNLOAD switch, and the tape should unload.

Printer Boxed Due to Lost Assign

If printer CHPIDs, control unit, switches, or ESCON connections are incorrectly handled, a printer ASSIGN may be reset (lost). When this occurs on the last path to the printer, the MVS system will box the device.

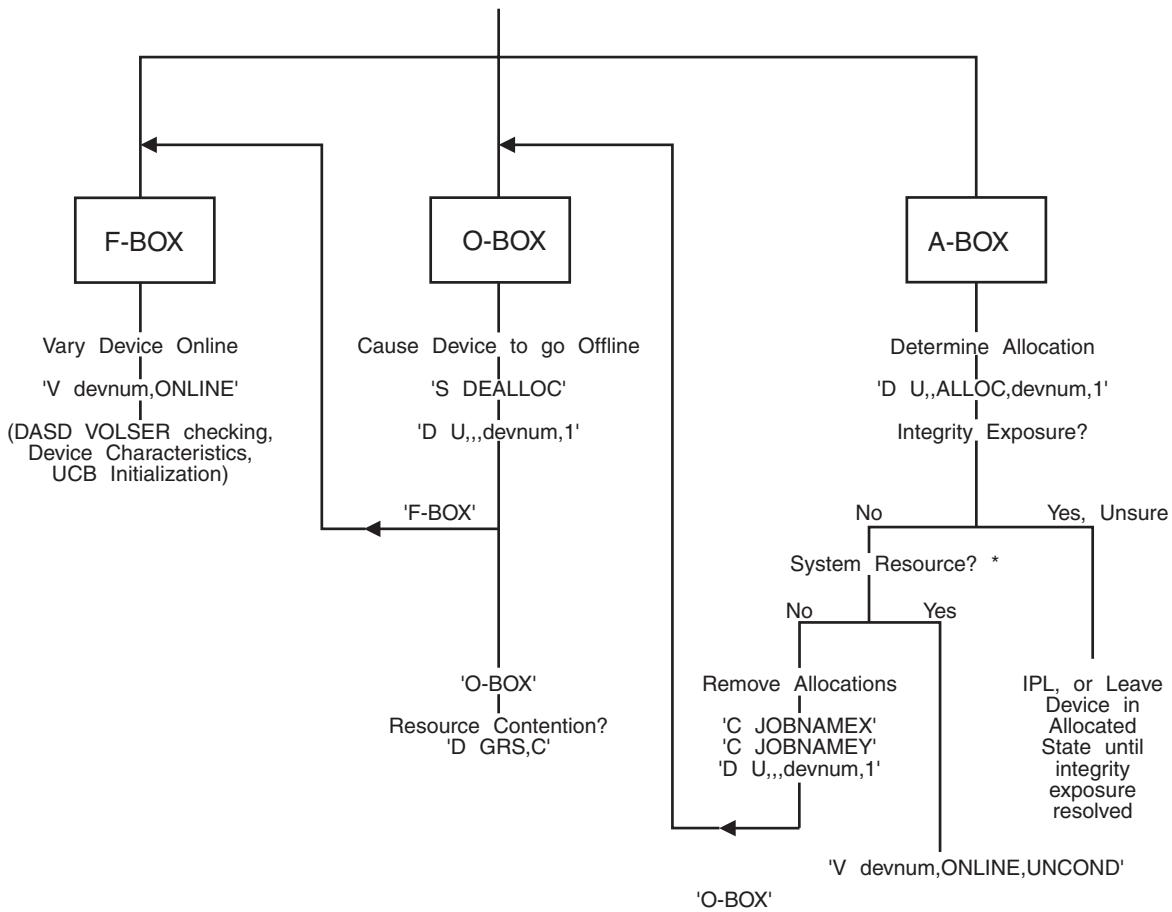
1. Locate the device boxed message to determine the reason for the BOX condition:

- I/O Error
- Lost Reserve or Lost Assign
- Lost Last Path (No Paths)
- UR Boxed
- Subchannel Recovery
- Vary Force
- Hot I/O

2. Correct the cause of the BOX condition before proceeding. (See next chart).

3. Recover boxed device:

Issue 'D U,,devnum,1' to Determine Device Status



Note: * System Resource - Any System Address Space (D A,ALL = *) or Subsystem

Figure 1-3. Boxed Device Recovery Procedure

The following procedures are recommended for use when a boxed condition is reported.

Table 1-3. Correcting Boxed Conditions

Cause	Operator Action
I/O Error	Correct the cause of the I/O error condition and then attempt to bring the device online
Lost Reserve or Assign	<p>For tape, if mount status=private, determine if any jobs were run or accesses made to the volume from any other system while it is in the boxed state.</p> <p>If yes, an integrity problem may exist and the device should not be varied online until the integrity of the volume can be assured.</p> <p>If no, you may attempt to un-BOX the device by varying it online.</p> <p>If the printer Assign was lost, it is possible the printer was assigned to another host. If so, first vary the printer offline from the other host and then vary online the printer to this host. Otherwise, attempt to vary the printer online to this host.</p>
Lost Last Path	Return the paths to an operational state and then vary the device online.
Subchannel Recovery	Identify and repair resource, then vary the device online
U/R 'Boxed'	Correct the cause of the I/O error condition and then attempt to bring the device online.
VARY FORCE command	Determine why the operator entered the VARY devnum,OFFLINE,FORCE command. Correct the condition on the system and vary the device back online.
Hot I/O	Identify and repair device, then vary the device back online.
Shared Tapes	Identify and repair device, then enter VARY devnum,ONLINE,UNCOND to unbox the device and bring it online

Command Flooding

Commands that run in the *MASTER* or CONSOLE address space are divided into six command classes. In each class, only 50 commands can execute at one time. Any additional commands in that class must wait for execution.

To manage the number of commands that are awaiting execution, the system operator can issue the CMDS command to display the status of commands, remove selected commands that are awaiting execution, or cancel commands that are executing. When a command is removed before execution, the command issuer receives message IEE065I COMMAND NOT EXECUTED, CMD=*command* instead of the usual command response message. When a command is canceled, the command is terminated with an ABEND code 422, reason code 00010301.

Class M1 Commands

Class M1 commands are commands that are attached in the *MASTER* address space, and are considered essential to clearing a backlog of other commands:

- DISPLAY GRS
- DISPLAY MPF
- DISPLAY SLIP
- DISPLAY XCF
- DUMP
- DUMPDS

- QUIESCE
- SET
- SETXCF
- SLIP
- VARY XCF

Class M2 Commands

Class M2 commands are ordinary attached commands that run in the *MASTER* address space:

- ACTIVATE
- CONFIG
- DEVSERV
- DISPLAY APPC
- DISPLAY ASCH
- DISPLAY ASM
- DISPLAY CEE
- DISPLAY CF
- DISPLAY CNGRP
- DISPLAY DLF
- DISPLAY DUMP
- DISPLAY ETR
- DISPLAY IOS
- DISPLAY IPLINFO
- DISPLAY LLA
- DISPLAY LOGGER
- DISPLAY LOGREC
- DISPLAY MATRIX
- DISPLAY MMS
- DISPLAY OMVS
- DISPLAY PARMLIB
- DISPLAY PROD
- DISPLAY PROG
- DISPLAY RTLS
- DISPLAY SMF
- DISPLAY SMS
- DISPLAY SSI
- DISPLAY SYMBOLS
- DISPLAY TCPIP
- DISPLAY TRACE
- DISPLAY U
- DISPLAY UNI
- DISPLAY WLM
- HALT EOD
- IOACTION
- LIBRARY

- LOGON (not MCS)
- MOUNT
- PAGEADD
- PAGEDEL
- RESET jobname
- SET UNI
- SETAPPC
- SETETR
- SETGRS
- SETIOS
- SETLOGR
- SETLOAD
- SETLOGRC
- SETOMVS
- SETPROG
- SETSMF
- SETSMS
- SETSSI
- START
- SWAP
- SWITCH SMF
- TRACE
- UNLOAD
- VARY GRS
- VARY ONLINE / OFFLINE
- VARY PATH
- VARY SMS
- VARY SWITCH
- VARY TCPIP
- VARY WLM

Class M3 Commands

Class M3 commands are ordinary attached commands that run in the *MASTER* address space. These commands can take a long time to execute, thus they require a command class different from Class M2:

- SEND

Class C1 Commands

Class C1 commands are those that are attached in the CONSOLE address space, and are considered essential to clearing a backlog of other commands:

- DISPLAY CONSOLES
- DISPLAY EMCS
- DISPLAY R
- LOGOFF
- LOGON (MCS)
- REPLY

- VARY CN
- VARY CONSOLE

Class C2 Commands

Class C2 commands are ordinary attached commands that run in the CONSOLE address space:

- CHNGDUMP
- CONTROL M
- DISPLAY A
- DISPLAY C,K
- DISPLAY JOBS
- DISPLAY OPDATA
- DISPLAY PFK
- DISPLAY TS
- RESET CN
- SETCON

Class C3 Commands

Class C3 commands are ordinary attached commands that run in the CONSOLE address space. These commands can take a long time to execute, thus they require a different command class than Class C2:

- ROUTE

Inline Commands

Inline commands are not attached, but execute under the SVC 34 issuer's task. These are not subject to the limits, and cannot be displayed, removed, or canceled, using the CMDS command:

- CANCEL
- CMDS
- CONTROL (except K M)
- DISPLAY NET
- DISPLAY T
- DISPLAY TP
- FORCE
- HALT NET
- HOLD TP
- LOG
- MODE
- MODIFY
- MONITOR
- RELEASE TP
- STOP
- STOPMN
- VARY NET
- WRITELOG

Responding to Failing Devices

Whenever a device fails, you can use the SWAP command to invoke dynamic device reconfiguration (DDR), which allows you to move or swap a demountable volume from the device.

Using the SWAP command, you can also turn on or off system-initiated swapping requests. When DDR is on, the system dynamically performs the swapping function whenever the originally-allocated device encounters device errors. DDR tells you to mount the volume on another available device. When the swapping function is turned off, you can invoke operator-initiated DDR by issuing the SWAP command and specifying the “from” and “to” device numbers. (See the SWAP command in Chapter 4.)

When swapping tape devices, the “from” and “to” devices should have the same density whenever possible. Swapping devices of unlike but compatible densities (for example, 1600 and 1600/6250) can cause the failure of jobs that are in device allocation at the time of the swap.

On JES3 systems, DDR interfaces with JES3 to ensure that the “to” device has not been assigned to another job or function. When the swap is complete, DDR notifies JES3.

The following devices are supported by DDR:

- 3400 series tape drives.
- 2501, 2540, 3505, 3525, 1403, and 3211 unit record devices. These devices are not swapped by system-initiated DDR; you must issue the SWAP command to swap these devices.
- 3330/3333 and 3340/3344 direct access devices. If you are using a 3348 Model 70F Data Module, make sure that the “to” 3340 device has the fixed-head feature installed. When swapping a 3340/3344 device with the fixed-head feature, be sure that the “to” device also has the fixed-head feature installed.

The following devices are not supported by DDR:

- Graphic or teleprocessing devices.
- Shared DASD devices, unless the device is swapped to itself.
- 3344 and 3350 fixed-head DASD devices (not supported by system-requested DDR).
- Any device holding a permanently-resident volume, such as a system residence or page data set volume.
- 3375 direct access storage device.
- 3380 direct access storage device.

Quiescing the System

Issuing the QUIESCE command causes the system to suspend the processing of all active jobs and to prevent the starting of any new ones. The system enters the MANUAL state, the MANUAL indicator is on, and no processing is being done. Quiescing the system does not affect any job step timings (for accounting purposes). Issue the QUIESCE command from any console with MASTER authority. You can continue processing by performing the restart function.

Do not issue a SYSTEM RESET after quiescing the system if you intend to issue a RESTART after the quiesce. Issuing a SYSTEM RESET will cause the system to enter an enabled wait state.

Stopping the System

When all processing (including subsystem processing) has finished, use the HALT command to ensure that all system statistics and data records in storage are collected for system recording facilities.

Recovery from Hardware Problems

Recovery is the attempt by the hardware, operating system, operator, automation, or any combination of these, to correct system malfunctions and return the system to a state in which it can do productive work. Recovery from some hardware errors is automatic; that is, the hardware recovers without any actions from the operating system or intervention by the operator or automation. Recovery from other hardware errors requires overt actions from the operating system, operator, and/or automation. For example, to keep the system in operation, the operator or the system can configure offline a failing unit, such as a storage element, a processor, or a channel path. The system continues processing, possibly with some degradation.

The process of recovery includes the following:

- Hardware to operating system communication and corrective actions
- Operator to operating system communication and recovery actions

Hardware Problems

This chapter describes the following categories of hardware malfunctions:

- Central processor (CPU) errors
- Service processor damage
- Storage errors
- Channel subsystem errors
- I/O device errors

For each of these categories, the discussion includes the effect on system operation and the recovery actions taken, if any. This chapter also presents some additional recovery actions.

Hardware to Operating System Recovery Actions

When CPU errors, service processor damage, storage errors, or channel subsystem errors occur, except for some I/O errors, the hardware notifies the operating system with a machine check interruption. Machine check interruptions fall into one of three classes depending on the severity of the error. The classes are:

- **Soft (or repressible) errors:** Least severe type. Generally these errors do not affect the operation of the task currently in control. Soft errors can be disabled (repressed) so that they do not cause a machine check interruption.
- **Hard errors:** Malfunctions that affect the processing of the current instruction or make incorrect the contents of hardware areas, such as registers.
- **Terminating error**
- s: Malfunctions that affect the operation of a CPU.

Hard and terminating errors are also referred to as *exigent* errors.

Information Provided with Machine Checks

When the hardware detects a failure, it stores the following information about the failure:

- The machine check interrupt code (MCIC), which contains:
 - Information about the severity of the error
 - The time of the error, in relation to the current instruction stream
 - An indication of whether the processor has successfully stored additional information about the error

The MCIC is the major interface between the hardware and the operating system, which uses the MCIC to determine what action to take.

- The save areas that contain the values of the general, floating point, control, and access registers, the CPU timer, and the clock comparator.
- The machine check old program status word (PSW), which contains the PSW at the time of error.
- The fixed logout area, which is implemented on only some processor complex models.

Reference Book

See the *Principles of Operation* for the format and content of the MCIC, register and timer save areas, extended interrupt information, machine check old PSW, and fixed logout area.

CPU Errors

CPU errors result from a malfunction of a hardware element, such as a timing facility, instruction-processing hardware, or microcode. When a CPU error occurs, the recovery processing has, in general, two stages depending on the severity and type of error:

1. When possible, the hardware retries the failing operation a certain number of times. If the retry works, the hardware may issue a recovery machine check interruption, which is repressible, so that the operating system can record the error in the logrec data set. After recording, the operating system returns control to the interrupted task.
2. If the error is too severe for hardware retry or the retries fail, the hardware issues either a hard or ending machine check interruption. The system determines the severity of the error and takes the appropriate action, which may range from ending the interrupted task to ending the entire system.

The next topics describe the following CPU errors:

- Soft CPU errors
- Hard CPU errors
- Ending CPU errors

Then the recovery actions of alternate CPU recovery (ACR) are described.

Soft CPU Errors

The CPU errors that can result in a soft machine check are:

- **System Recovery (SR):** A malfunction has occurred, but the hardware has successfully corrected or circumvented it.

- **Degradation (DG):** A continuous degradation of system performance has been detected.

The operating system does not inform the operator about the occurrence of soft machine checks until the threshold for a given type is reached. The default threshold set for an SR machine check is 50, and for a DG machine check it is 1. When a threshold for a type of machine check is reached, the system issues message IGF931E.

The MODE command allows the operator to change the threshold value for either SR or DG machine checks, and to specify what processing should be done when the threshold is reached.

- The operator can specify that at the threshold the CPU be disabled for that type of machine check, that is, be put in *quiet* mode.
- If the MODE command specifies RECORD=ALL for a particular type of machine check, the system does not enter quiet mode; it records all instances of the specified type of machine check in the logrec data set. The operating system issues message IGF931E when the number of machine checks reaches a multiple of the threshold. For example, if REPORT=3 is specified, message IGF931E appears after the third, sixth, ninth, twelfth machine checks, and so on.

Numerous IGF931E messages appearing on the console might indicate a performance degradation. In this case, the installation might want to configure offline the processor that is experiencing the errors. Hardware support personnel can repair the offline processor.

Hard CPU Errors

A hard machine check indicates that the current instruction could not complete. The system records the error in the logrec data set. Then the system either abnormally ends the interrupted task or retries the interrupted task at a predefined retry point. Even though the task may be ended, the system usually continues to run.

The CPU errors that cause hard machine checks are:

- **System Damage (SD):** A malfunction has caused the processor to lose control over the operation it was performing to the extent that the cause of the error cannot be determined.
- **Instruction Processing Damage (PD):** A malfunction has occurred in the processing of an instruction.
- **Invalid PSW or Registers (IV):** The hardware was unable to store the PSW or registers at the time of error, as indicated by validity bits in the MCIC. Any error, even a soft machine check, associated with these validity bits is treated as a hard machine check because the operating system does not have a valid address to use to resume operation. The error goes through recovery processing.
- **Timing Facility Damage:** Damage to the following has been detected:
 - TOD clock (TC)
 - Processor timer (PT)
 - Clock comparator (CC)
 - External Time Reference (ETR)

The four types of ETR-related machine checks are: primary synchronization damage, ETR attachment damage, switch to local, and ETR synchronization check.

To overcome the effects of numerous hard machine checks, the MODE command allows the operator to define machine check thresholds for each type. When

reached, the thresholds cause the failing processor to be configured offline by alternate CPU recovery (ACR). Thus, the operator can control whether, and to what extent, the system monitors the frequency of hard machine checks, and can define a separate threshold and time interval for each.

The default threshold value for most hard machine checks is 5. The default for PD machine checks is 16. The default for ETR machine checks is 5 in 300 seconds.

Terminating Errors on CPUs

A terminating machine check occurs when the operating system or the hardware considers a failure severe enough that a processor cannot continue operation.

In a uniprocessor (UP), the operating system enters a disabled non-restartable wait state, such as X'A01' or X'A26', and issues the following message:

IGF910W UNRECOVERABLE MACHINE FAILURE, RE-IPL SYSTEM

In a multiprocessor (MP), the action taken is as follows:

- If the hardware determines that a processor cannot continue operation, it places the processor in a check-stop state and attempts to signal the other processor(s) by issuing a malfunction alert (MFA) external interruption. The hardware issues an MFA when:
 - It cannot store the machine check logout data about the error.
 - It cannot load the machine check new PSW.
 - It is disabled for hard machine checks when a hard error is detected.
- If the operating system determines that a processor cannot continue operation, it attempts to signal the other processor(s) by issuing a Signal Processor (SIGP) instruction to cause an emergency-signal (EMS) external interruption. The operating system issues an SIGP instruction when:
 - The system is processing one machine check when another machine check occurs that cannot be handled.
 - A hard-machine-check threshold, which is an installation option established by entering the MODE command, has been reached.
 - Channel subsystem damage is detected.
 - The content of the MCIC is incorrect.

When a processor receives either an MFA or EMS external interruption for these conditions, the system receives control. The system, in turn, invokes ACR processing, which takes the malfunctioning processor offline and initiates recovery processing for that processor.

In a multiprocessor environment, an MFA or EMS is received by all the other online processors. On the first processor to receive the signal, the system tests and sets a flag before starting to process the error. When the other processors receive the interruption, the system sees that the error is already being processed and returns to the interrupted task.

Terminating Errors on Multiprocessors: In a multiprocessor, failure of some hardware elements may cause a terminating error on more than one CPU. It is possible that a terminating error may occur on a CPU while alternate CPU recovery (ACR) is still processing a terminating error on another CPU. In either case, the system puts the system into non-restartable wait state X'050'.

Alternate CPU Recovery (ACR)

ACR is a function that is initiated on an operative CPU when that CPU receives a signal that another CPU has had an ending error. ACR has two major functions:

- To configure offline the malfunctioning CPU
- To initiate the release of system resources held on the malfunctioning CPU

If the failing CPU has an Integrated Cryptographic Feature (ICRF), the ICRF is also taken offline.

ACR initiates the release of any resources held on the failing CPU by causing control to pass to the recovery routines for the work on the failing CPU. ACR allows the operating system to continue its normal operation on the remaining CPU(s), although the task that was interrupted by the error on the failing CPU might be ended.

When ACR is complete, it issues message IEA858E stating that ACR is complete and identifying the CPU that was configured offline. At this point, the operator can try to configure the failing CPU back online using a CONFIG CPU(x),ONLINE command. The configuration online might, or might not, be successful depending on the error that caused the CPU to be configured offline.

Some hardware malfunctions might cause a subsequent CONFIG CPU(x),ONLINE command to that CPU to fail, or might cause the problem to recur when the CPU is brought back online. In these cases, hardware support personnel need to service the CPU before it can be successfully brought back into the system.

However, if a CPU was configured offline because a threshold was reached or because of an operating system problem, a subsequent request to configure the CPU back online might work.

Service Processor Damage

Permanent Failure

When the system detects that the service processor is permanently, completely failing, the system receives a service processor damage machine check. The system also notifies subsystems about the damage.

For a permanent failure, the system issues the following message:

```
ISN000E THE PROCESSOR CONTROLLER HAS FAILED. SOME CRITICAL SYSTEM  
FUNCTIONS HAVE BEEN DISABLED. AN ORDERLY SHUTDOWN OF THE  
ENTIRE SYSTEM SHOULD IMMEDIATELY BE ATTEMPTED IN ORDER TO  
MINIMIZE THE IMPACT OF THIS FAILURE
```

After this message, the operator can optionally perform an orderly shutdown of the system. Processing can continue, but when a function of the service processor is required, the system may become inoperative. To recover, the operator then performs an initial microprogram load (IML).

Temporary Failure

If a service processor fails temporarily or partially and is in I/O Support Processor (IOSP) concurrent maintenance mode, the system continues operating but cannot perform certain functions.

For a temporary failure, a message with the prefix ARRP is issued to the operator.

In IOSP concurrent maintenance mode, certain functions of the operating system will not work or will work incompletely.

Storage Errors

The hardware detects and corrects storage errors where possible. The system is informed of the error by a machine check interrupt. The system invokes recovery routines.

If the storage error is detected during an I/O operation, however, the operation is ended with either a channel data check or a channel control check, depending on whether the error was encountered during data transfer or fetching of the channel control word (CCW) and indirect data address word (IDAW). No machine check interrupt is generated in this case. Error recovery procedures (ERPs) recover from this type of error.

Soft Storage Errors

The soft storage errors are system recovery (SR) errors with the **storage error corrected** flag set in the MCIC to indicate that the storage controller was able to repair the error.

When a **storage error corrected** (SC) condition occurs, along with **storage degradation** (DS), the system attempts to stop using the affected frame. This action eliminates performance degradation that would result from hardware correction of later occurrences of the same error. It also minimizes the chance that the same problem will later occur as a **storage error uncorrected**.

If the frame contains pageable data, the system moves that data to another frame, and the original frame is marked offline. If the data in the frame cannot be moved, the frame is marked **pending offline**, and is subsequently taken offline if the frame is released or if its contents are made pageable. Note that, before the system takes a frame offline, it tests the frame; if it has no errors, the frame is returned to available status.

The threshold for SR machine checks affects the ability of the system to deal with **storage error corrected** conditions. The default threshold is 50 SR machine checks. The operator can change the SR threshold with the MODE operator command. When the threshold is reached, the system disables SR machine checks. This action prevents a subsequent **storage error corrected** from being presented. The system then does not take any action to remove the affected frame.

Hard Storage Errors

This section deals with these types of hard storage errors:

- **Storage error uncorrected:** Indicates that the hardware could not repair a storage error.
- **Key in storage error uncorrected:** Indicates that the hardware could not repair a storage key that was in error.

When a hard storage error occurs, the operating system attempts recovery. For a storage key problem in a frame containing a virtual page, the operating system tries to reset the key. If the reset fails and the page is not fixed, the operating system moves the page to a new frame, setting the key in the new frame as required.

If recovery cannot repair the error, the operating system either takes the storage frame offline or marks it pending offline. *Pending offline* means that the operating system will take the frame offline when the frame becomes free.

A **storage error uncorrected** condition represents the potential loss of critical data. When this condition occurs with a PD machine check, the system in most cases

ends the affected unit of work. If the recovery routines complete successfully so that the affected storage frame is freed, the frame is marked offline and system processing continues. The recovery processing, however, could try to refer to the storage that originally caused the machine check, thus causing further errors. Such action could result in the PD threshold for machine checks being reached, thus taking a CPU offline.

The default threshold for PD machine checks is 16 in 5 minutes. The operator can change this threshold by means of the MODE operator command.

Effects of Storage Errors

Errors in critical areas of storage may cause the hardware system or the operating system to become inoperative. Those areas of storage and the effect of an error are as follows:

- **Hardware storage area (HSA):** An uncorrectable storage error in the HSA causes the system to enter a check-stop state. The system can be recovered by two actions:
 1. Power-on reset (POR) or a SYSML CLEAR service language command
 2. IPL
- **Nucleus:** A storage error in nucleus pages requires an IPL for recovery. If the IPL fails, recovery requires either a power-on reset or a SYSML CLEAR, followed by an IPL.
- **Link pack area (LPA), system queue area (SQA), and local SQA (LSQA):** A storage error in SQA could have the same effects as a nucleus storage error.

For a storage error in LPA, the operating system handles recovery. Normally, only the associated job is ended with the remainder of the system unaffected.

High Speed Buffer (Cache)

A processor cache error can result in the loss of the processor and possibly the system. The storage frame corresponding to any changed data in the cache is marked with an uncorrectable storage error. Because the cache might contain critical system data, recovery might require an IPL.

Channel Subsystem Errors

If the channel subsystem fails, the hardware generates a **channel subsystem damage** machine check interrupt. The resultant processing enters the entire system into non-restartable wait state X'A19' and issues message IOS019W.

Channel Report Words (CRWs)

When the channel subsystem detects an error, it does the following:

- Builds a CRW that describes the error
- Queues the CRW for retrieval by the operating system
- Generates a machine check interrupt with the **CRW pending** indicator set in the machine check interrupt code (MCIC)

The operating system records the CRW in a logrec data set error record. The CRW contains a code that indicates the source of the error: the channel path, the subchannel, channel configuration alert, or the monitoring facility.

Reference Book

See *Principles of Operation* for additional information on CRWs.

Channel Path Recovery

If the CRW indicates that a channel path caused the machine check, the system attempts to recover the channel path or route I/O down an alternate channel path. If multiple CRWs indicate errors on different channel paths, a failure in the hardware elements common to those channel paths may be indicated.

The channel path conditions fall into two categories:

- Expected: An expected channel path condition occurs as a result of a previous recovery action taken for an unexpected channel path error, and indicates the result of the action.
- Unexpected: An unexpected channel path error occurs with no warning.

The channel path conditions indicated in a CRW are:

- A terminating error condition on the channel path.
- A permanent error on the channel path; a system reset to the channel path has not been done.
- A permanent error on the channel path; a system reset to the channel path has been done.
- An initialized condition on a channel path, that is, an error recovered by the channel subsystem; a system reset to the channel path has been done.

Terminating Error: A terminating error condition is unexpected only; it is never the result of a previous recovery action. A terminating error condition indicates that the channel path is not permanently lost, but cannot be used until the error condition is reset. In this case, the system attempts to reset the channel path. The CRW that results from this reset is an expected CRW, and it will indicate whether the reset corrected the problem in the channel path.

For a failing channel path that has a device with an outstanding reserve, the system handles the condition in three different ways, depending on whether the device supports dynamic pathing, supports unconditional reserve, or does not support unconditional reserve. The system actions are:

- For a dynamic pathing device with multiple paths, no action is taken until the expected CRW is received.
- For a non-dynamic pathing device that supports unconditional reserve, such as a 3350 Direct Access Storage, the system issues an unconditional reserve command to the device to move the reserve to an alternate path.
- For a non-dynamic pathing device that does not support unconditional reserve, such as a 3330 Disk Storage, or for a reserved/assigned device with only one path, the system issues message IOS063E (or IOS062E) to request that the operator stop I/O to the shared devices (see Figure 1-4 on page 1-48 and the accompanying description). The system then tries to recover the channel path.

Note: Since stopping I/O to shared devices may require a certain level of multi-system disruption and coordination, users may wish to avoid this processing. Through the use of the TERMINAL BOX_LP(device_class1,...device_classN) statement in the IEClOSxx Parmlib member, users can cause devices in the specified device class to be BOXED rather than having to undergo multi-system disruption to recover the channel path to the device. For more information on the use of the TERMINAL statement in IEAIOSxx, see *z/OS MVS Initialization and Tuning Reference*.

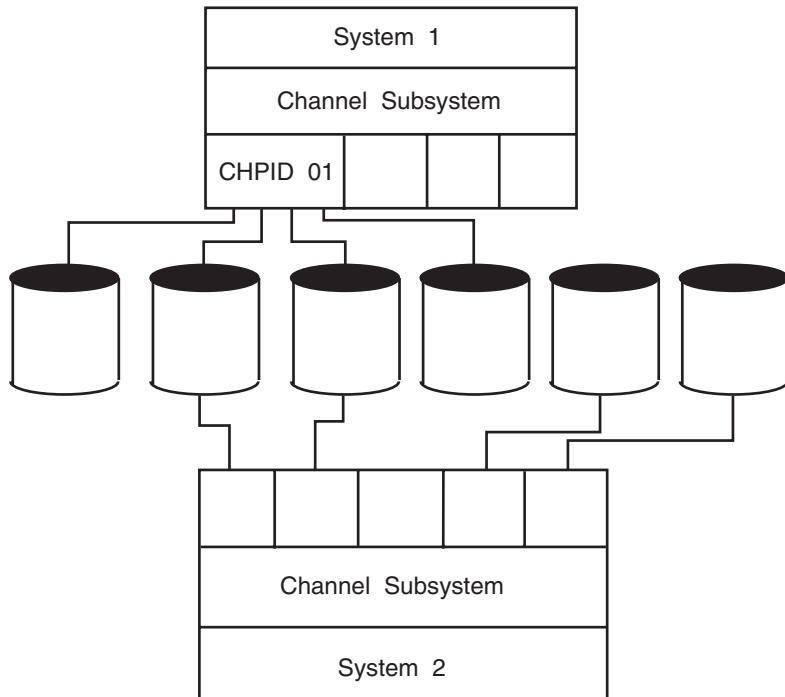


Figure 1-4. DASD Devices Shared Between Two Systems

Figure 1-4 shows two DASD devices that are shared between two systems. When system 1 encounters a channel path error on channel path 01, indicated by message IOS063E (or IOS62E), the operator should stop I/O to the shared devices from system 2 to maintain data integrity during recovery of the channel path.

Recovery Actions for a Channel Path Error with Shared DASD or Assignable Devices:

1. Identify which devices on channel path 01 on system 1 are shared with system 2.
2. Enter the IOACTION STOP command on system 2 to stop I/O on the shared devices. The device numbers **may** not be the same on both systems.
3. Restart system 1.
4. Wait for the system to issue message IOS204E (or IOS201E), indicating that channel path recovery is complete.
5. Enter the IOACTION RESUME command on system 2 to allow I/O to resume to the shared devices.

Notes:

1. Do not leave devices in the stopped state any longer than necessary to perform recovery. A shortage of SQA storage can result from stopping I/O for extended periods.
2. Before stopping a device, enter the D U,DASD,ALLOC,xxx command to determine which system resource's I/O will be affected. If any system oriented I/O will be stopped, the system could appear frozen. This situation will last until I/O resumes to the device.

Permanent Error Condition: A permanent error condition, whether expected following a Reset to a channel path in an ending condition or unexpected, results in the system taking that channel path offline. Any active I/O requests are retried on alternate paths if available. If the failing channel path was the last path to any devices, those devices are boxed. Boxing means:

- The operating system ends I/O to the device.
- Any new I/O request for the device causes a permanent I/O error.
- The operating system does not allocate the device.
- If the device was online, the operating system marks it pending offline. A pending offline device goes offline when the following occur, in this order:
 1. The device becomes no longer allocated to any job.
 2. The operating system allocates any device.

If the device was offline, it remains offline.

Initialized Condition: An initialized condition means that a previous recovery action has successfully recovered the channel path and the channel path is available for use. This condition can be expected only. The initialized condition indicates that the channel subsystem has been successful in recovering the channel path to a state where it is again usable.

For devices that support the Dynamic Path Selection (DPS) feature, such as the 3380 Direct Access Storage and 3480 Magnetic Tape Subsystem, DPS validation is called to restore dynamic pathing arrays for each DPS device attached to that channel path. For each non-dynamic pathing device that does not support unconditional reserve and that had an outstanding reserve on the failing channel path, a reserve command is issued to the device. Any previously active I/O requests are restarted.

Channel Path Alert Conditions

The operating system communicates with the operator when two other indicators are set in a CRW: **channel path temporary** and **configuration alert temporary**. In either case, the operating system performs no recovery processing.

- **Channel path temporary:** The operating system issues message IOS162A to inform the operator that the channel subsystem could not identify the device requesting service.
- **Configuration alert temporary:** The operating system issues message IOS163A to inform the operator that the channel subsystem could not associate a valid subchannel with the device requesting service.

Subchannel Recovery

If the CRW indicates that a subchannel caused the machine check, the operating system examines the error recovery code in the CRW. If the CRW indicates that the subchannel is available, the channel subsystem has recovered from a previous malfunction. I/O functions in progress and presentation of status by the device have not been affected. No program action is required.

If the CRW indicates that the subchannel is **installed parameter initialized**, the operating system determines if the device associated with the subchannel is still valid. If it is, the operating system enables the subchannel again. If, however, the device related to the subchannel is not valid, the operating system marks the device as unusable and issues message IOS151I.

Monitoring Facility Recovery

For a channel monitoring error, the operating system schedules a recovery routine.

I/O Device Errors

An error can occur in an I/O device. The following topics cover:

- “I/O Errors”
- “Missing Interrupts”
- “Hot I/O” on page 1-51
- “Recovery for Hung Devices” on page 1-53
- “Recovery for Failing Devices” on page 1-54
- “Shared Device Recovery” on page 1-55
- “3880/3380 Considerations” on page 1-57
- “DASD Maintenance and Recovery” on page 1-58
- “Recovery for a Failing Alias Unit Control Block (UCB)” on page 1-33

I/O Errors

Errors that are related to an I/O request are usually indicated in the status data provided with the I/O interrupt. These errors are:

- Device not operational on any path
- Device status errors, such as a unit check
- Subchannel status errors: interface control check, channel control check, and channel data check

The operating system processing of the interrupt may include:

- Invoking a driver exit
- Interfacing with attention routines and volume verification processing
- Invoking a device-dependent ERP for error recovery
- Processing an unconditional reserve
- Redriving the I/O request on a channel path other than the one that generated the interrupt
- Requesting an operator action by message IOS115A or by restartable wait state X'115'
- Issuing message IOS050I to inform the operator that a subchannel status error occurred

Missing Interrupts

At predefined intervals, the operating system checks devices of a specific type to determine if expected I/O interrupts have occurred. If an expected interrupt has not occurred across two of these checks, that interrupt is considered missing. The operating system then issues message IOS071I or IOS076E, writes a logrec data set error record, and tries to correct the problem. For recurring missing interrupts, the operating system issues message IOS075E together with message IOS076E or IOS077E to indicate the recurring condition on a particular device.

A feature of the IBM 3990-6 and 9340 attached devices allows MVS/ESA to automatically identify a system in a multisystem environment that is holding a reserve. After every start pending MIH condition, the system attempts to determine whether the device is not responding because of a reserve to another system. If the device is reserved to another system, message IOS431I is issued to identify the system by its central processor serial number. If the system holding the reserve is a member of the same sysplex as the system detecting the MIH condition, message IOS431I includes the system name and the LPAR ID, if there is one.

For JES2 systems, when the reserve is held by a system in the same sysplex, the system attempts to obtain information about the job causing the reserve by routing a D GRS,DEV=devnum command to that system. JES2 systems which have JES3

installed must have JES2 started with the NOJES3 option (CON=(xx,NOJES3) in order to identify the job holding the reserve. Message ISG020I identifies the jobs holding the reserve on the failing system. The installation can use this information to determine what to do.

Some causes of missing interrupts are:

- An idle unit control block (UCB) with I/O requests queued to it
- An outstanding I/O operation that should have completed
- An outstanding mount for a tape or disk

The intervals used by the operating system to determine whether an expected interrupt is missing varies from 15 seconds for DASD to 12 minutes for 3330 Disk Storage. An installation can define in the IECIOSxx parmlib member the time intervals for all devices in the I/O configuration. These intervals override the IBM-supplied defaults.

Notes:

1. During IOS recovery processing, the system will override your time interval specification and may issue MIH messages and MIH logrec error records at this IOS determined interval.
2. During IPL (if the device is defined to be ONLINE) or during the VARY ONLINE process, some devices may present their own MIH timeout values, via the primary/secondary MIH timing enhancement, contained in the self-describing data for the device. The primary MIH timeout value is used for most I/O commands; however, the secondary MIH timeout value may be used for special operations such as long-busy conditions for long running I/O operations. Any time a user specifically sets a device or device class to have an MIH timeout value that is different from the IBM-supplied default for the device class, the value will override the device-established primary MIH time value. This implies that if an MIH time value that is equal to the MIH default for the device class is explicitly requested, IOS will **not** override the device-established primary MIH time value. To override the device-established primary MIH time value, you must explicitly set a time value that is not equal to the MIH default for the device class.

Note that overriding the device-supplied primary MIH timeout value may adversely affect MIH recovery processing for the device or device class.

Please refer to the specific device's reference manuals to determine if the device supports self-describing MIH time values.

Reference Book

See *z/OS MVS Initialization and Tuning Reference* for the IECIOSxx member.

Note: If there are missing interrupts on the devices that contain the system residence (SYSRES) or the page volumes, the operator may not receive any message, because the needed operating system routines are pageable. The operator can learn about the missing interrupts by initiating restart reason 1.

Hot I/O

A hot I/O condition occurs when a device, control unit, or channel path causes continuous unsolicited I/O interrupts. The operating system attempts to recover from a hot I/O condition so that a reIPL is not required. For diagnostic purposes, the operating system indicates all hot I/O incidents in logrec data set error records.

The operating system first tries recovery at the device level by issuing the Clear Subchannel (CSCH) instruction in an attempt to clear the hot I/O condition. If the condition is cleared, processing continues normally. If the condition persists, the next recovery action is determined by one of the following:

- The parameters the installation defined in the IECIOSxx parmlib member for hot I/O recovery
- Operator response to the appropriate hot I/O message or restartable wait state for the class of device:
 - Message IOS117A, (IOS110D, or wait state X'110') for non-DASD, non-dynamic pathing device
 - Message IOS118A, (IOS111D, or wait state X'111') for DASD or dynamic pathing device that is not reserved
 - Message IOS119A, (IOS112D, or wait state X'112') for DASD or dynamic pathing device that is reserved

Because IPLs related to hot I/O are generally caused by incorrect operator actions, an installation should use the IECIOSxx parmlib member to make hot I/O recovery more automatic and reduce the need for immediate operator intervention. The following example parameters, when defined in the IECIOSxx parmlib member, tell the operating system how to handle automatic recovery from hot I/O.

Reference Books

See *z/OS MVS Initialization and Tuning Reference* for the IECIOSxx parmlib member.

Example IECIOSxx Parameters for Hot I/O Recovery: The following examples show how to specify the hot I/O recovery parameters in the IECIOSxx parmlib member. The values shown are also the IBM default values.

HOTIO DVTHRSH=100

Specifies 100 repeated interrupts as the threshold for the operating system recognizing the condition.

HOTIO DFLT110=(BOX,)

For a non-DASD, non-dynamic pathing device. Box the device on the first occurrence of this condition. On recursion, prompt the operator.

HOTIO DFLT111=(CHPK,BOX)

For a DASD or dynamic pathing device that is not reserved. Attempt channel path recovery for the device on first occurrence of this condition. On recursion, box the device.

HOTIO DFLT112=(CHPK,OPER)

For a DASD or dynamic pathing device that is reserved. Attempt channel path recovery for the device on first occurrence of this condition. On recursion, prompt the operator.

Hot I/O Recommendations: Although it does require more operator intervention, additional experience has shown that a higher level of availability can be achieved by specifying hot I/O recovery options (CHPK,OPER) for all three classes of devices: DFLT110, DFLT111, and DFLT112. These options will allow the possibility of automatic recovery before the system requests operator involvement.

Operator involvement might require that the message be issued using the Disabled Console Communication Facility (DCCF). This will occur when a DASD device is

attached on the channel path that is undergoing Hot I/O recovery. Unless the installation is prepared to deal with messages issued in DCCF mode, IBM recommends that operator involvement not be requested by the Hot I/O actions.

For non-DASD devices (DFLT110), CHPK,OPER would allow one automated recovery attempt and then request direction from the operator, depending on how critical the device is. If the device is non-critical to the operation of the system, such as a printer, the operator could then reply **BOX**. If the device is critical, such as a 3705 Communication Controller, and is properly equipped with multiple paths, such as through a type three channel adapter on the 3705, the operator could reply **CHPF**. Note that the operator should ensure that all critical devices on the same channel path (CHP) have multiple paths before replying **CHPF**.

You can choose recovery options other than the defaults or the general recommendations, because of considerations unique to the installation. For example, if all DASDs are configured with multiple paths, each through a different CHP, you might consider specifying **CHPK,CHPF** for both DFLT111 and DFLT112. This will allow one attempt to recover without loss of resources, then an attempt to recover by removing the CHP attached to the failing device or control unit, but without losing a critical device. If the installation contains DASD devices with only one path, the considerations are the same as for non-DASD.

Note: Since CHPK processing may require a certain level of multi-system disruption and coordination (to stop sharing processors), the CHPK option may not be suitable for all device classes in a particular Hot I/O device grouping. For instance, CHPK processing may be suitable for reserved DASD devices, but may not be suitable for assigned single-path tape devices. CHPK can be avoided on a device class basis by using the HOTIO BOX_LP(device_class1,...device_classN) parameter in the IECIOSxx parmlib member. Using the BOX_LP parameter forces a device to be BOXED for Hot I/O conditions that cause CHPK processing to occur.

Recovery for Hung Devices

When a device appears to be hung, an operator can consider varying the device offline to try releasing the hang condition at the device. However, the VARY OFFLINE command may obtain resources critical to the system and may attempt to issue I/O. Depending on the condition that caused the device to hang in the first place, this may cause the VARY OFFLINE command to also hang.

To avoid this problem, the operator can choose to use the VARY OFFLINE command with the FORCE parameter to mark the device offline and boxed. The FORCE parameter will not obtain resources or issue I/O, so it will complete regardless of any hardware problems with the device. Since the device is boxed, all I/O will be posted back to the I/O issuer in permanent error, which should cause all system resources previously held to be released.

Note: See “Boxed Device - Operator Actions” on page 1-32 for more information on boxing devices.

Reference Book

See *z/OS MVS System Commands* for information on the FORCE option found under the VARY command.

Recovery for Failing Devices

When a device fails, operators can enter the SWAP command to perform dynamic device reconfiguration (DDR). DDR allows the operator to move or swap a demountable volume from a device. When DDR is active, the system dynamically requests the swapping whenever a device encounters device errors. DDR tells the operator to mount the volume on another available device.

Operators can invoke DDR by issuing the SWAP command and specifying the *from* and *to* device numbers.

Reference Book

See *z/OS MVS System Commands* for the SWAP command.

When swapping tape devices, the *from* and *to* devices should have the same density, whenever possible. Swapping devices of unlike but compatible densities can fail the jobs in device allocation at the time of the swap.

On JES3 systems, DDR checks with JES3 to ensure that the *to* device has not been assigned to another job or function. When the swap is complete, DDR notifies JES3.

When a data check occurs in an IBM 3495 Tape Library Dataserver, the system cleans the tape device and retries the failing operation. If the error persists, the system initiates a swap to another eligible system-managed tape library device without involving the operator. The system will try to swap to up to five other devices. If these efforts fail, it issues an error message to the operator and fails the job.

The following devices are supported by DDR:

- 3400 series Magnetic Tape Units.
- Unit record devices. These devices are not swapped by system-initiated DDR; the operator must enter the SWAP command to swap these devices.
 - 1403 Printer
 - 2501 Card Reader
 - 2540
 - 3211 Printer
 - 3505 Card Reader
 - 3525 Card Punch
- Direct access devices. When using a 3348 Model 70F Data Module, operators must ensure that the *to* 3340 has the fixed-head feature installed. When swapping a 3340/3344 with the fixed-head feature, be sure that the *to* device also has the fixed-head feature installed.
 - 3330 Disk Storage
 - 3333 Disk Storage and Control
 - 3340 Direct Access Storage
 - 3344 Direct Access Storage

The following devices are not supported by DDR:

- 3344 and 3350 Direct Access Storage with fixed-head are not supported by system-requested DDR.
- 3375 Direct Access Storage.

- 3380 Direct Access Storage.
- Shared DASD devices, unless the device is swapped to itself.
- Any device holding a permanently-resident volume, such as a system residence or page data set volume.
- Graphic or teleprocessing devices.

Shared Device Recovery

When a system, for example system A, is sharing devices with other systems, events on any one system can affect the ability of any or all the systems to access the shared devices. For example, if one of the sharing systems has an allegiance to a shared device, an I/O operation from system A to that device will receive a **device busy** condition. In this case, the I/O operation is held in system A's channel subsystem until the other system ends its allegiance. At that time, system A's I/O to that device can then be processed normally.

However, a problem either on the system that has the allegiance or in the I/O hardware could result in the allegiance not being freed. This could prevent the processing of any pending I/O operations from any of the sharing systems to the device(s) affected by the allegiance. The indication to any sharing systems that had an I/O operation hung by such a condition would normally be message IOS071I, indicating a **start pending** to the device.

There are a number of other conditions that can cause message IOS071I for a shared device:

- Poor performance of programs using the device
- Contention for the device
- Long reserves
- Application errors
- Operator errors

Operator Actions: If message IOS071I occurs for a shared device, the operator should check the operating condition of the sharing systems. If any system is not operational, it could be the cause of the start-pending because the system could still hold an allegiance that it had when it became non-operational. Operator action depends on the condition of the non-operational system:

- If the system is in the check-stop state or in a non-restartable wait state, the operator should immediately initiate an interface reset to that device from the system console of the non-operational system. If the interface reset fails to release the device, the operator should issue a system reset from the system console of the non-operational system.

These actions should be taken before trying to recover the non-operational system, to allow the other operational systems that are sharing I/O with the non-operational system to continue with as little disruption as possible.

- If the system is in the stopped state, the operator should try to determine why it is stopped and, if appropriate, start it again.
- If the system issued message IOS431I, take the actions described in the operator response for that message.

Unconditional Reserve/Alternate Path Recovery: Alternate path recovery permits recovery from control unit or channel-path failures that cause a DASD or string of DASDs to be no longer accessible to the system. Alternate path recovery is performed only after the operating system guarantees ownership of the device by ensuring that the device is reserved to this system.

When the operating system can guarantee ownership of the device, alternate path recovery is performed. This consists of issuing an unconditional reserve (UR) CCW on a non-failing, online path to the device. If the UR CCW is successful, the operating system issues message IOS428I. If the UR CCW fails or if there are no alternate paths to the device, the operating system issues message IOS429I and boxes the device.

If ownership of the device cannot be established, the operating system issues message IOS427A to determine which recovery action is to be performed.

To maintain volume data integrity during a shared DASD recovery process, the operator must stop I/O from the sharing systems until the recovery process is complete. Prior to MVS/SP 4.2.2, the operator stops I/O to a device only by one of the following:

- Cancelling all jobs allocated to the device and varying the device offline on all sharing systems.
- Stopping the sharing processors.

With MVS/SP 4.2.2 and all subsequent releases, the operator should use the IOACTION STOP command to stop I/O to the device with less disruption to system processing.

See Figure 1-5 on page 1-57 for an illustration of three systems sharing a device. Use the following recovery scenario when system 1 issues message IOS427A, indicating a device failure.

1. Determine the device number on each sharing system.
2. To stop I/O requests for the device, enter IOACTION STOP,DEV=devnum on system 2 and system 3.
If the IOACTION STOP command fails because the device is reserved or reserve pending, and repeated attempts to stop the I/O using the command continue to fail, then end the reserving task or stop the system.
3. Reply UR to message IOS427A and wait for recovery completion messages on system 1.
4. After the recovery completion messages, enter IOACTION RESUME,DEV=devnum or IOACTION RESUME,ALL on system 2 and system 3 to return I/O processing to normal.

Notes:

1. Do not leave devices in the stopped state any longer than necessary to perform recovery. A shortage of SQA storage can result from stopping I/O for extended periods.
2. Before stopping a device, enter the D U,DASD,ALLOC,devnum command to determine which system resource's I/O will be affected. If any system oriented I/O will be stopped, the system could appear frozen. This situation will last until I/O resumes to the device.

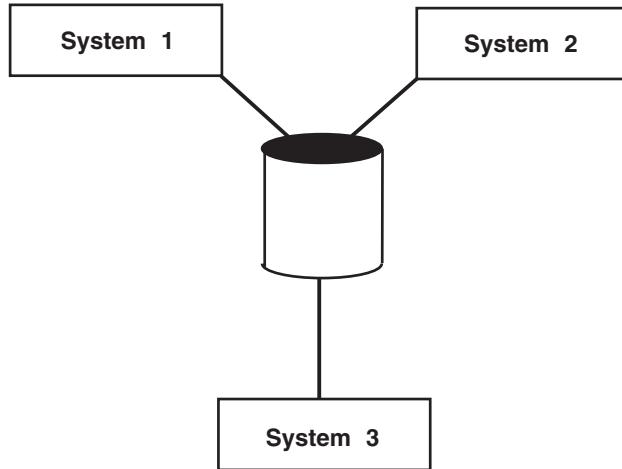


Figure 1-5. DASD Device Shared Between Three Systems

3880/3380 Considerations

The 3380 Direct Access Storage Model AA4 with the 3880 Storage Control is designed to allow concurrent maintenance at the storage director (SD) level. Prior to attempting concurrent maintenance, all paths from all processor complexes through the failing SD to the devices must be varied offline. Failure to vary all paths offline may result in various error symptoms, including interface control checks, path inoperative conditions, and out-of-sync conditions between the 3380 array and the operating system.

Prior to returning a repaired SD to the system, an IML of the SD or a power down-up sequence must be performed to establish a correct copy of the dynamic pathing support array in the 3380 for the repaired SD. The operator should enter VARY PATH ONLINE commands for all paths to all devices through the repaired SD.

Out-of-Sync Conditions: The enable/disable switch on the 3880 A box should NEVER be set to **disable** when any paths to the device are online. Setting the switch to **disable** could cause an **out-of-sync** condition between the array and the operating system. This out-of-sync condition can occur whenever the dynamic path group information maintained in the 3880 A box is reset without notification to the operating system. Any of these operator actions could cause an out-of-sync condition:

- IMLing the 3880 Storage Control
- Disabling the 3880 interface switch
- Disabling the 3380 interface switch

In addition, certain 3880/3380 hardware failures can affect the arrays.

Recovery from an Out-of-Sync Condition: Array out-of-sync conditions may be indicated by missing interrupts or path-inoperative I/O errors. The system provides automatic detection and recovery through the dynamic pathing validation support. This code detects potential out-of-sync conditions, such as missing interrupts, and then validates the physical path group information. If the dynamic pathing validation

code finds a mismatch between the hardware and software path group information, it invokes recovery to rebuild the dynamic path selection arrays.

Dynamic Pathing Support (DPS) Out-of-Sync Recovery: DPS validation is invoked because of an error condition usually symptomatic of a DPS array out-of-sync condition. MIH-start-pending messages are the most common symptom. The system detects and DPS validation repairs the DPS arrays for only one device at a time.

Usually more than one device is affected, because the cause is typically related to:

- An action on the 3880 SD, such as an IML or interface disable/enable
- An action on the channel path, such as an IFRST

Depending on the device activity and the rate of RESERVE macros per volume, the system could take several minutes to several hours to repair the DPS arrays for all affected devices. While the arrays are out-of-sync, performance is impacted because of the loss of the dynamic path reconnect function. In addition, the installation is exposed to a failure or operator action on another path that might then result in undetected loss of reserves (a data integrity exposure) or boxing of the device.

It is therefore recommended that whenever a DPS array out-of-sync condition is detected for one device, as indicated by message IEA452I, the operator should enter a VARY PATH ONLINE command to all devices associated with the suspected component.

For example, if message IEA452I indicates that the DPS arrays for device 100 through 10F on channel path 22 have to be repaired, the operator would then enter a command to vary the path through CHP 22 for all associated devices that have a path already online.

```
VARY PATH((100-10F),22),ONLINE
```

DASD Maintenance and Recovery

DASD can experience failures such as defective disk surfaces, drives, and actuators. When these failures occur, data becomes inaccessible to the operating system and could be lost. To prevent the loss of the data, an installation should consider using the following to monitor possible error conditions and correct any before they cause outages.

- The Environmental Record Editing and Printing (EREP) System Exception Report
- Device Support Facilities (ICKDSF)

When a DASD error does occur, such as a defective track, an installation can use Data Facility Data Set Services (DFDSS) to retrieve the data from the defective areas and copy it to a back-up DASD.

Reference Books

See the following:

- *IBM Disk Storage Management Guide: Error Handling*
- *ICKDSF User's Guide*
- *EREP User's Guide*
- *DFDSS User's Guide*

Additional Recovery Actions

This section includes these topics:

- Recovery by CPU restart

Recovery by CPU Restart

The operator can initiate recovery from some system incidents, such as loops and uncoded wait states, by issuing a restart to the processor that has the problem. The restart reason that is entered as part of the restart process directs the system to perform one of two recovery actions:

Restart reason 0

The system tries to display message IEA500A on the first two consoles defined as master or alternate and locally attached to the system issuing the message.

The message identifies the current unit of work on the target processor. The operator can reply either:

- RESUME to allow the current unit of work to continue
- ABEND to end that unit of work with an abend X'071'

If the operating system cannot communicate with the either console to issue message IEA500A, it ends the current unit of work with an abend X'071'.

Restart reason 1

The operating system:

- Interrupts the current unit of work on the target processor
- Detects and attempts to repair errors in critical system areas
- Writes a logrec data set error record for completion code X'071' with reason code 4 when repair actions were taken
- Reports the results of some of the actions taken in message IEA501I
- Returns control to the interrupted unit of work

Note: Restart of the CPU in a restartable wait state ignores the restart reason.

Reference Book

See *z/OS MVS System Commands* for information about restart.

Chapter 2. System Configuration

Reconfiguration

Reconfiguration is the process of adding hardware units to, or removing hardware units from, a configuration. Units can be either:

- **Online:** Units in use by a system are called *online*. When both physically and logically online, a unit is available to be used by the system.
- **Offline:** Units not in use by a system are called *offline*. When either physically or logically offline, a unit is not available to be used by the system.

An installation can use reconfiguration to:

- Adapt a system to changing work loads by configuring units online or offline as required.
- Perform maintenance on a part of a complex while the other part continues normal operation.
- Possibly recover system operation by configuring failing units offline.

Hardware unit or units may be put offline before initialization of the system or systems. To do this, an operator can deselect through the system console such units as central processors (CPU), storage elements, or channel paths. Note that an operator should never deselect a unit during system operation, because the operating system is not notified of the removal.

During system operation, the operating system configures failing units offline with or without any intervention. When intervention is needed, the operator:

- Can enter a CONFIG command that identifies the CONFIGxx member of Parmlib that specifies the reconfiguration. (See *z/OS MVS Initialization and Tuning Reference*.)
- Can enter a CONFIG or VARY command that directly configures units online or offline. (See *z/OS MVS System Commands*.)

For maintenance of hardware, prepare reconfiguration actions to configure hardware units offline for repair and back online so the system can use them. Also, hardware units and channel paths are reconfigured offline then online during reconfiguration of partitionable systems.

Dynamic I/O Configuration

Dynamic I/O configuration lets you change your I/O configuration without causing a system outage. In other words, you can select a new I/O configuration definition without performing a power-on-reset (POR) of the hardware or an initial program load (IPL) of the MVS system. Using I/O definition files (IODFs) created through hardware configuration definition (HCD), dynamic I/O configuration allows you to add, delete, or modify the definitions of channel paths, control units, and I/O devices to the software and hardware I/O configurations. You can change the I/O configuration definitions to both software and hardware or software only. (See *z/OS HCD Planning*.)

Dynamic I/O configuration has the following benefits:

- Increases system availability by allowing you to change the I/O configuration while MVS is running, thus eliminating the POR and IPL for selecting a new or changed I/O configuration definition.

- Allows you to make I/O configuration changes when your installation needs them rather than wait for a scheduled outage to make the changes.
- Minimizes the need to over-define an I/O configuration by logically defining hardware devices that do not physically exist.

Logical and Physical Reconfiguration

Logical reconfiguration allows or prevents use of a resource by the operating system. Physical reconfiguration allows or prevents use of a resource by the hardware. An operator can enter a CONFIG command on the console with master authority to logically and physically reconfigure the following hardware units, if applicable for the particular processor complex:

- CPUs
- Central storage elements
- Central storage increments
- Channel paths
- I/O devices

Also, when an operator issues a CONFIG CPU command to reconfigure a processor, the system logically reconfigures any Integrated Cryptographic Feature (ICRF) attached to the processor.

Note: Physical reconfiguration may not be supported for all hardware units by all processor models. (See *Functional Characteristics*).

Reconfiguration Support According to Processor Types

The reconfiguration functions supported by MVS depend on the configuration of the processor complex, as follows.

Uniprocessor (UP)

Depending on the processor type, an installation can configure offline some or all of the following:

- Central storage elements
- Central storage increments
- Channel paths
- I/O devices

In a UP system, the purpose of reconfiguration is to configure offline failing units to allow the system to continue operation.

Multiprocessor (MP)

Depending on the processor type, an installation can configure offline some or all of the following:

- Central processors, including any associated ICRFs
- Central storage elements
- Central storage increments
- Channel paths
- I/O devices

Reconfiguring a Central Processor

When configuring a central processor offline, the operating system stops dispatching work to the processor and takes it logically offline. Then the system stops the processor removes it physically from the configuration, if physical central processor reconfiguration is supported by the machine.

Actions to Reconfigure a Central Processor Offline

1. Enter a CONFIG CPU(x),OFFLINE command on a console with master authority. The system responds on the console on which the command was entered.
The operating system rejects a CONFIG CPU(x),OFFLINE command when:
 - The target processor is the only online processor.
 - The target processor is the only processor with an operative timer.
 - Alternate CPU recovery (ACR) occurs during OFFLINE processing.
 - Any active jobs have processor (CPU) affinity with the target processor.
2. If you enter a CONFIG command for a central processor and currently scheduled jobs have CPU affinity for that processor, the system issues message IEE718I to list the jobs. Do the following:
 - a. Prevent the operating system from scheduling any additional jobs, by replying YES to message IEE718D.
Note that replying YES to message IEE718D leaves the target central processor unavailable for affinity job scheduling. If you want to restore the central processor to its original state, enter CONFIG CPU(x),ONLINE to restore the original central processor status and make the target central processor available for affinity job scheduling.
 - b. Either wait for the active jobs to complete or cancel them.
 - c. Reenter the CONFIG CPU(x),OFFLINE command.

Reconfiguring a Central Processor with an ICRF

The operator cannot directly reconfigure an ICRF. When the operator uses the CONFIG command to reconfigure a central processor with an associated ICRF, the system changes the online/offline status of both the central processor and the associated ICRF as follows:

- When the operator enters a CONFIG CPU(x), ONLINE and Integrated Cryptographic Service Facility/MVS (ICSF/MVS) is active in the processor complex, the system brings the ICRF online.
- When the operator enters a CONFIG CPU(x), OFFLINE command the system takes the ICRF offline.
- When the operator enters a CONFIG CPU(x), ONLINE and ICSF/MVS is not active, the ICRF is not brought online until ICSF/MVS is started.

Actions to Bring Online a Central Processor and its ICRF

Central processor x (CPU x) is to be brought online. The processor has an associated ICRF and ICSF/MVS is installed and active in the processor complex.

Enter one of the following commands on a console with master authority; the system responds on the console on which the command was entered:

CONFIG CPU(x)

CONFIG CPU(x), ONLINE

When the command completes, the system issues the following messages:

IEE504I CPU(x), ONLINE
IEE504I CRYPTO(x), ONLINE

Removing the Last ICRF

When a CONFIG command is entered to remove a central processor associated with the last online ICRF, the system issues the following messages:

```
IEE109I CONFIG CPU(x), OFFLINE COMMAND WOULD REMOVE LAST CRYPTO  
IEE325D REPLY U TO CONTINUE CONFIG COMMAND. REPLY C TO CANCEL
```

After *U* is replied to message IEE325D, the system takes the central processor and the ICRF offline.

From that point on, the system abnormally ends any jobs that request cryptographic services using ICSF/MVS. This applies to new jobs and to jobs running in the processor complex at the time the system took the ICRF offline.

Actions to Take Offline a Central Processor and its ICRF

Central processor x is to be taken offline. The processor has an associated ICRF.

Enter the following command on a console with master authority; the system responds on the console on which the command was entered:

```
CONFIG CPU(x), OFFLINE
```

When the command completes, the system issues the following messages:

```
IEE505I CPU(x), OFFLINE  
IEE505I CRYPTO(x), OFFLINE
```

Reconfiguring Central Storage

To place storage offline and then bring it online to be used again, commands can reconfigure central storage increments and central storage elements.

Under PR/SM, storage may be reconfigured between logical partitions. See *PR/SM Planning Guide* for a description of storage reconfiguration between logical partitions.

Physical View of Central Storage

Central storage is physically divided into storage elements (SE). The central storage elements are composed of storage increments. Each storage increment may be composed of two sub-increments:

- One sub-increments contains the even-numbered frames of the increment, such as, 0-kilobyte, 8-kilobyte, 16-kilobyte, and so on.
- The other contains the odd-numbered frames, such as, 4-kilobyte, 12-kilobyte, 20-kilobyte and so on.

The sub-increments of an increment may reside in different storage elements, as follows:

SE0

SE2

SE1

SE3

30-32M Even Frames Reconfigurable
28-30M Even Frames Preferred
26-28M Even Frames Preferred
24-26M Even Frames Preferred

30-32M Odd Frames Reconfigurable
28-30M Odd Frames Preferred
26-28M Odd Frames Preferred
24-26M Odd Frames Preferred

62-64M Even Frames HSA and Preferred
60-62M Even Frames Reconfigurable
58-60M Even Frames Reconfigurable
56-58M Even Frames Reconfigurable

62-64M Odd Frames HSA and Preferred
60-62M Odd Frames Reconfigurable
58-60M Odd Frames Reconfigurable
56-58M Odd Frames Reconfigurable

6-8M Even Frames SQA and Preferred
4-6M Even Frames Preferred
2-4M Even Frames Preferred
0-2M Even Frames V=R and Preferred

6-8M Odd Frames Preferred
4-6M Odd Frames Preferred
2-4M Odd Frames Preferred
0-2M Odd Frames V=R and Preferred

38-40M Even Frames Reconfigurable
36-38M Even Frames Reconfigurable
34-36M Even Frames Reconfigurable
32-34M Even Frames Reconfigurable

38-40M Odd Frames Reconfigurable
36-38M Odd Frames Reconfigurable
34-36M Odd Frames Reconfigurable
32-34M Odd Frames Reconfigurable

Storage Increments and Sub-increment Sizes

The following central storage increments cannot be configured offline:

- The increment containing absolute address 0
- The highest addressable increment available at IPL
- Any increment containing preferred central storage frames

A storage element can be configured offline only if:

- It contains only non-preferred storage frames
- The preferred storage sub-increments in this storage element can be moved to another storage element containing reconfigurable storage sub-increments.

Processor Complex	Storage Element ID	Storage Element Size (megabytes)	Increment Size (megabytes)	Sub-increment Size (megabytes)	Maximum Storage Size (megabytes)
ES/3090 Model 110J	1	32	2	1	64
ES/3090 Model 100S	1	32	2	1	64
ES/3090 Model 120J	1	32	4	2	64
ES/3090 Model 120S	1	32	2	1	64
ES/3090 Model 150J	1	32	4	2	64
ES/3090 Model 150JH	1	32 - 64, then 128	4	2	128
ES/3090 Model 150S	1	32	2	1	64
ES/3090 Model 170J	1	32 - 64, then 128	4	2	128
ES/3090 Model 170JH	1	32 - 64, then 128	4	2	128
ES/3090 Model 170S	1	32	2	1	64
ES/3090 Model 180J	1	32 - 64, then 128	4	2	128
ES/3090 Model 180S	1	32 - 64, then 128	4	2	128
ES/3090 Model 200	0, 1	32	2	1	64
ES/3090 Model 200E	0, 1	32, 64	4	2	128
ES/3090 Model 200J	0, 1	64 - 128; then 128 - 256	4	2	256
ES/3090 Model 200S	0, 1	64 - 128; then 128 - 256	4	2	256
ES/3090 Model 250J	1, 3	32	4	2	128
ES/3090 Model 250JH	1, 3	On each side: 32 - 64, then 64 - 128	4	2	256
ES/3090 Model 250S	1, 3	64	4	2	128
ES/3090 Model 280E	0, 1, 2, 3	16, 32	4	2	128
ES/3090 Model 280J	1, 3	On each side: 32 - 64, then 64 - 128	4	2	256
ES/3090 Model 280S	1, 3	64 - 128, then 128 - 256	4	2	256
ES/3090 Model 300E	0, 1	32, 64	4	2	128
ES/3090 Model 300J	0, 1	64 - 128, then 128 - 256	4	2	256
ES/3090 Model 300S	0, 1	64 - 128, then 128 - 256	4	2	256
ES/3090 Model 380J	0, 1, 3	On each side: 64 - 128, then 128 - 256	4	2	512
ES/3090 Model 380S	0, 1, 3	128 - 256, then 256 - 512	4	2	512
ES/3090 Model 400	0, 1, 2, 3	32	2	1	128
ES/3090 Model 400E	0, 1, 2, 3	32, 64	4	2	256

Processor Complex	Storage Element ID	Storage Element Size (megabytes)	Increment Size (megabytes)	Sub-increment Size (megabytes)	Maximum Storage Size (megabytes)
ES/3090 Model 400J	0, 1, 2, 3	On each side: 64 - 128, then 128 - 256	4	2	512
ES/3090 Model 400S	0, 1, 2, 3	128 - 256, then 256 - 512	4	2	512
ES/3090 Model 500E	0, 1, 2, 3	32, 64	4	2	256
ES/3090 Model 500J	0, 1, 2, 3	On each side: 64 - 128, then 128 - 256	4	2	512
ES/3090 Model 500S	0, 1, 2, 3	128 - 256, then 256 - 512	4	2	512
ES/3090 Model 600E	0, 1, 2, 3	32, 64	4	2	256
ES/3090 Model 600J	0, 1, 2, 3	On each side: 64 - 128, then 128 - 256	4	2	512
ES/3090 Model 600S	0, 1, 2, 3	128 - 256, then 256 - 512	4	2	512
ES/9000 Model 620	0, 1, 2, 3	128, 192, 256, 384, 512	4	2	512
ES/9000 Model 720	0, 1, 2, 3	128, 192, 256, 384, 512	4	2	512
ES/9000 Model 820	0, 1, 2, 3	256, 384, 512, 768, 1024	4	2	1024
ES/9000 Model 900	0, 1, 2, 3	512, 768, 1024	4	2	1024

Specifying the RSU Parameter

To prepare for reconfiguration of central storage, specify an RSU parameter in the IEASYSxx parmlib member or as a parameter during system initialization. The RSU parameter specifies the number of storage units in central storage that the operating system should keep available for reconfiguration.

Reference Book

See *z/OS MVS Initialization and Tuning Reference* for the RSU parameter.

The RSU storage increments are *reconfigurable* or *non-preferred*. The remaining central storage increments are *non-reconfigurable* or *preferred*.

The default RSU value is 0. If the installation does not specify an RSU value, the default means that the operating system designates ALL installed central storage as preferred. With RSU=0, the system cannot be reconfigured.

Reconfigurable Storage Increments: The central storage that is shared by all processors in a configuration is logically divided into storage increments. For the following example, the installation specified RSU=8 to make half of the storage

increments reconfigurable.

60M	Hardware storage area (HSA) and preferred
56M	Reconfigurable
52M	Reconfigurable
48M	Reconfigurable
44M	Reconfigurable
40M	Reconfigurable
36M	Reconfigurable
32M	Reconfigurable
28M	Reconfigurable
24M	Preferred
20M	Preferred
16M	Preferred
12M	System queue area (SQA) and Preferred
8M	Preferred
4M	Preferred
0M	V=R and Preferred

The HSA cannot be in a storage increment that will be reconfigured. The HSA resides in the five highest megabytes of central storage.

RSU Example: Assume that a processor complex has 128 megabytes of central storage and that the storage increment size is 4 megabytes. Such a system has 32 storage increments.

If the system initializes one side with RSU=16, the operating system allocates as reconfigurable the 16 storage increments (64 megabytes) of the offline side.

RSU Parameter Specification: The RSU values recommended for the least system overhead and maximum capability for reconfiguration are as follows:

Processor	Recommended RSU Value
Uniprocessor	RSU=0
Non-partitionable multiprocessor	RSU=0
Partitionable processor in single-image mode	RSU= $\frac{\text{Total megabytes of installed central storage}}{2 * \text{Central storage increment size in megabytes}}$
Partitionable processor in physically partitioned mode	RSU= $\frac{\text{Total megabytes of installed central storage}}{2 * \text{Central storage increment size in megabytes}}$

Assignment of Storage Frames: Storage containing fixed pages cannot be reconfigured. The operating system assigns storage frames as follows:

Storage	Assignment
Non-preferred and preferred	Normal page allocation requests
Non-preferred and preferred	Short-term page fixes
Preferred	Long-term page fixes for non-swappable jobs

However, if a long-term page fix for a non-swappable job requires storage but the preferred storage units are full, the operating system may convert some non-preferred storage to preferred storage. If so, the amount of storage available for reconfiguration will be less than that specified in the RSU parameter. The operating system issues message IAR005I to notify the operator.

If the operator then tries to configure storage offline in preparation for partitioning, the system tries to free enough central storage to support the request. Central storage and the address ranges assigned to that storage cannot be configured offline either logically or physically until the required amount of storage is available.

The operating system normally tries to assign requests for long-term fixed pages to preferred storage frames when the requesting job was initiated as non-swappable. However, an authorized job can be initiated as swappable and, while running, issue a SYSEVENT macro to make itself non-swappable for a short period of time. The job may request long-term fixed pages that are assigned to non-preferred storage. Usually this request is not a problem because the job shortly makes itself swappable again. The system can free the storage that backs the long-term fixed pages when the job is swapped out when the storage is required for storage reconfiguration.

However, a long-running job may make itself non-swappable for long times and also request short-term fixed pages that cannot be freed until the job ends normally. Some of these requests may be satisfied from non-preferred storage. Because the frames cannot be freed by paging them out or by swapping out the job, storage reconfiguration may not be possible.

To resolve this problem, specify such jobs in the program properties table (PPT) in the SCHEDEXX parmlib member.

Reference Book

See *z/OS MVS Initialization and Tuning Reference* for the SCHEDEXX member.

Actions to Reconfigure Central Storage:

Enter at the console with master authority or in a CONFIGxx parmlib member:

1. CONFIG STOR(E=id),OFFLINE command to configure a central storage element offline
2. CONFIG STOR(E=id),ONLINE command to configure a central storage element online

The system responds on the console on which the command was entered or on the master console for the system processing the CONFIGxx member.

When configuring a central storage element offline, the system may issue message IEE575A to indicate that central storage configuration is waiting to complete. The system may cancel the message in less than a minute. The system may issue and cancel the message several times.

If the message remains on the display for a long time, it indicates that the operating system cannot find sufficient reconfigurable central storage to satisfy the configuration request.

If the system displays message IEE575A for a long time:

1. Enter a DISPLAY M=STOR command to identify the job using the central storage that cannot be freed.
2. Do one of the following:
 - a. Cancel any jobs that are using the storage to allow the storage configuration to complete.
 - b. Reply C to end the storage configuration process.

Any central storage already configured offline remains offline. If desired, bring this central storage element back online by entering a CONFIG STOR(E=x),ONLINE command.

Reconfiguring Channel Paths

Channel paths may be reconfigured for partitioning and merging. Under PR/SM, channel paths may be reconfigured to move a channel path from one logical partition to another.

Reconfigure channel paths carefully. Reconfiguration can remove connections to DASDs or MCS consoles that are critical to the operation of MVS. If this occurs, system operation will fail.

Actions to Reconfigure Channel Paths

To configure channel paths, enter the following:

- To configure channel paths individually offline or online:

```
CONFIG CHP(x),OFFLINE  
CONFIG CHP(x),ONLINE
```

- To configure all channel paths owned by a side of a partitionable processor complex:

```
CONFIG CHP(ALL,x),OFFLINE  
CONFIG CHP(ALL,x),ONLINE
```

These commands are particularly useful when partitioning or merging the complex.

- To configure a range of channel paths:

```
CONFIG CHP(x-y),OFFLINE  
CONFIG CHP(x-y),ONLINE
```

The system determines which devices are connected to a channel path and if that path is the last path to a device. To configure offline the last path to a device, enter a CONFIG command with one of the following operands:

- UNCOND operand: To configure offline the last path to an unallocated, online device
- FORCE operand: To configure offline the last path to a device regardless of the state of the device.

To make sure that FORCE is intended, the operating system issues message IEE800D to ask whether to continue or stop the CONFIG command processing.

Enter all the CONFIG commands at a console with master authority. The system responds on the console on which the command was entered.

Reconfiguring I/O Devices

To reconfigure I/O devices, enter a VARY ONLINE/OFFLINE command at the console with master authority. For a VARY OFFLINE command for a device that is currently in use, the operating system marks the device ‘pending offline’. The operating system makes no further allocations to the device unless the volume mounted on the device is specifically requested.

Because vary offline processing cannot complete until a device is unallocated, either wait until the jobs using the device complete or cancel them.

Before reconfiguring a complex from single-image mode to physically-partitioned mode, complete or cancel any tape volume mounts for devices to be reconfigured. Then enter a CONFIG CHP(ALL,n),OFFLINE,UNCOND command.

If a tape mount is pending when the CONFIG command is entered, the tape drive(s) might not start after they are mounted and the system has been partitioned. Another way to avoid this problem is to enter a VARY device online command for the tape drive(s).

Reconfiguring a Coupling Facility

A coupling facility can be reconfigured to perform maintenance or to upgrade its level of control code. Before reconfiguring a coupling facility, you should have a plan in place for its orderly shutdown, during which time structures that are in the coupling facility can be either relocated to another coupling facility or deallocated

(with a possible data loss). Applications using the coupling facility for its structure(s) should have documented procedures for how to relocate or deallocate them.

Refer to *Parallel Sysplex Configuration Assistant* at <http://www.ibm.com/s390/pso/psotool> for help with the configuration of a coupling facility.

Chapter 3. System Console Operations

The tasks of starting, running, and stopping an MVS system involve:

1. Operating the system itself—that is, controlling the system software and most installation hardware (including processors, channel paths, and I/O devices)
2. Operating the MCS (multiple-console support) and SMCS (SNA multiple-console support) consoles

“Console Characteristics and Operations” describes the physical characteristics and techniques for operating the various consoles that MVS supports as operators’ consoles. It describes the characteristics and operations that you **cannot** control, including those operations that are common to all operator’s consoles.

“Defining and Changing Console Characteristics” on page 3-20 continues the console descriptions of “Console Characteristics and Operations” by describing the console characteristics that you **can** control. It describes the commands that operators and system programmers can use to tailor the consoles and console operations to the installation’s requirements.

Console Characteristics and Operations

General Characteristics of Display Consoles

Many different input and output (I/O) devices can function as consoles in an MVS system. Three logical conditions determine how or if the devices function. A device can be:

1. Online: If allocated, the system assigns functions with these two limitations:
 - The device must be capable of performing the function.
 - The device cannot be assigned as a console because it is allocated to some other function.If unallocated, the device can be assigned as a console.
2. Offline: The device is generally unavailable for the system to use.
3. Console: The system can use the device to send messages to you, and you can use the device to issue system commands (if the device has input capability), but you cannot use the device for other input/output purposes.

You can use a device as a multiple console support (MCS) if the device number for the console on a CONSOLE statement, in the CONSOLxx parmlib member, is the same as the device number specified in the IODF. If you use MVSCP, the device number specified for each console in CONSOLxx — the CONSOLE DEVNUM(devnum) statement — must correspond to a device number specified on an IODEVICE statement. SMCS consoles are also defined in CONSOLxx, but are not specified in HCD.

Subsystem Use of Consoles

Many different devices can function as consoles in an MVS system if they are specified as consoles in a CONSOLxx parmlib member. If the console is allocated to a subsystem — CONSOLE DEVNUM(SUBSYSTEM) — there is no corresponding device definition in the IODF. You should familiarize yourself with subsystem consoles if your configuration includes them; some of them can affect MVS operations in important ways. It is called a subsystem-allocatable console and is defined to the subsystem.

For a subsystem-allocatable console, the definition
CONSOLE DEVNUM(SUBSYSTEM)

must appear in the CONSOLOxx parmlib member.

Multiple-Console Configuration

You can divide the functions and message traffic of the system among a number of consoles. These consoles make up a **multiple-console configuration** controlled and serviced by MCS.

A multiple-console configuration for a system or sysplex consists of up to 99 consoles. These consoles can have different levels of authority. For more information, see *z/OS MVS Planning: Operations*.

Any console with master console authority allows you to:

- Enter all operator commands
- Change the definition of the hardcopy message set or assign the hardcopy medium

Other MCS and SMCS consoles are used for specific types of operator-system communication when it is more convenient to have a console located away from the processor. An MCS or SMCS console might, for example, be located close to tape or disk drives or remote teleprocessing devices to make it easier for the operator in that area to see which magnetic tapes or disk packs to mount. An MCS or SMCS console without master authority cannot enter all commands (see “System Commands Grouped According to System Command Authority” on page 3-24), and can receive only those messages that are specifically routed to that console.

Your installation might further limit how you can use a console by assigning an operating use that prevents the console from accepting commands.

A console you use both to issue commands and receive messages is in **full-capability** operating mode. A console that only receives status displays is in **status display** mode. A console that only monitors system activities and assists in system security is in **message stream** mode. Both message stream and status display consoles do not accept commands.

The different console modes help limit the number of consoles that operators can use to issue commands, and yet provide operators the information that they need to do their work.

At IPL, the system looks to the CONSOLOxx member of parmlib to determine the operating modes of the consoles. It also looks for the following attributes:

- System command groups — the categories of commands that the system accepts from that console
- Message routing codes — the messages the console receives, determined by routing code
- Message levels — the messages the console receives, determined by message level
- Hardcopy medium — the system log (SYSLOG) or operations log (OPERLOG) that receives the hardcopy message set
- PFK definitions — the commands that console's PFKs issue

Features on Display Consoles

MCS display consoles can operate in full-capability, status display, or message stream mode. SMCS only operate in full-capability mode. Each one has a keyboard to enter commands and responses to messages and to signal the system that you are entering information. Each one also has a cursor, which appears on the screen as a movable point of light (either an underscore, a horizontal bar, or a vertical bar). The cursor points out the position on the screen that the system will examine for your next action. This action might be positioning a typed character, entering a command, requesting message deletion, or requesting a display. Special keys located on the console keyboard control cursor movement.

A display console can also have some or all of the following features:

Selector Pen

The selector pen is a light-sensitive device that is available on some display consoles. When you put the pen over specific areas of the display console screen, it senses the light from the screen and signals the system. The system then determines the screen location over which you have put the pen and takes appropriate action. The action the system takes might involve entering operator commands, deleting messages from the screen, cancelling processes, or presenting displays.

Audible Alarm

An audible alarm is available on display consoles. The system sounds this alarm when certain changes in conditions occur, such as when you enter an invalid CONTROL command. WTO macros with descriptor codes of 1, 2, or 11, and all WTOR macros will cause the audible alarm to sound on operator consoles so-equipped.

Program Function Keyboard

The program function keyboard is an input device that is available on some display consoles. You can define each key on the program function keyboard to enter one or more operator commands; you can enter a command or a group of commands by pressing one key.

Extended Highlighting

Extended highlighting refers to blinking, reverse video, and underscored presentation of messages that require operator action.

Color

Four or more colors are available on some devices, with certain colors identifying certain kinds of messages that require action.

Intensity

Some messages that require operator action appear brighter.

Display Screen Areas

The operating mode of the console controls the appearance of a display screen. Figure 3-1 on page 3-5 illustrates the differences among the three different kinds of consoles. The display screens can have these functional areas:

Message Area

This area contains system and problem program messages and copies of certain operator commands. The size of the message area depends on the console.

Display Areas

These areas contain formatted, multiple-line displays of information about some part of the system. The displays are written to the console in response to certain commands, such as the DISPLAY command. The default on consoles in full-capability mode is one display area, the default on consoles in status display mode is two display areas. For consoles operating in full-capability mode, unless a status display is requested, the display area is used for general messages.

PFK Display Line

This line contains a display of program function key (PFK) numbers that you use when entering commands with the selector pen. This line is available on a 3277 model 2.

Instruction Line

This line contains console control messages. For example, if you make an error entering a CONTROL command, an error message appears in the instruction line.

Entry Area

This area contains one or two lines that you use to enter commands and reply to messages.

Warning Line

This line warns you of conditions that could require action. For example, a warning message appears in this line when the message area is full and one or more messages are waiting to appear. The warning line is not available on output-only consoles in status display operating mode.

Operator Information Area

This line, the bottom-most line on the screen, is separated from the rest of the screen by a horizontal line. The operator information area, which is not controlled by MCS or SMCS, contains messages and symbols that keep you informed of the operational status of the terminal. It is not available on some terminals.

Figure 3-1 shows the screens on consoles in the three different operating modes. You can change the display areas on the consoles in full-capability mode and status display mode. The screen on the console in message stream mode always appears as in the figure.

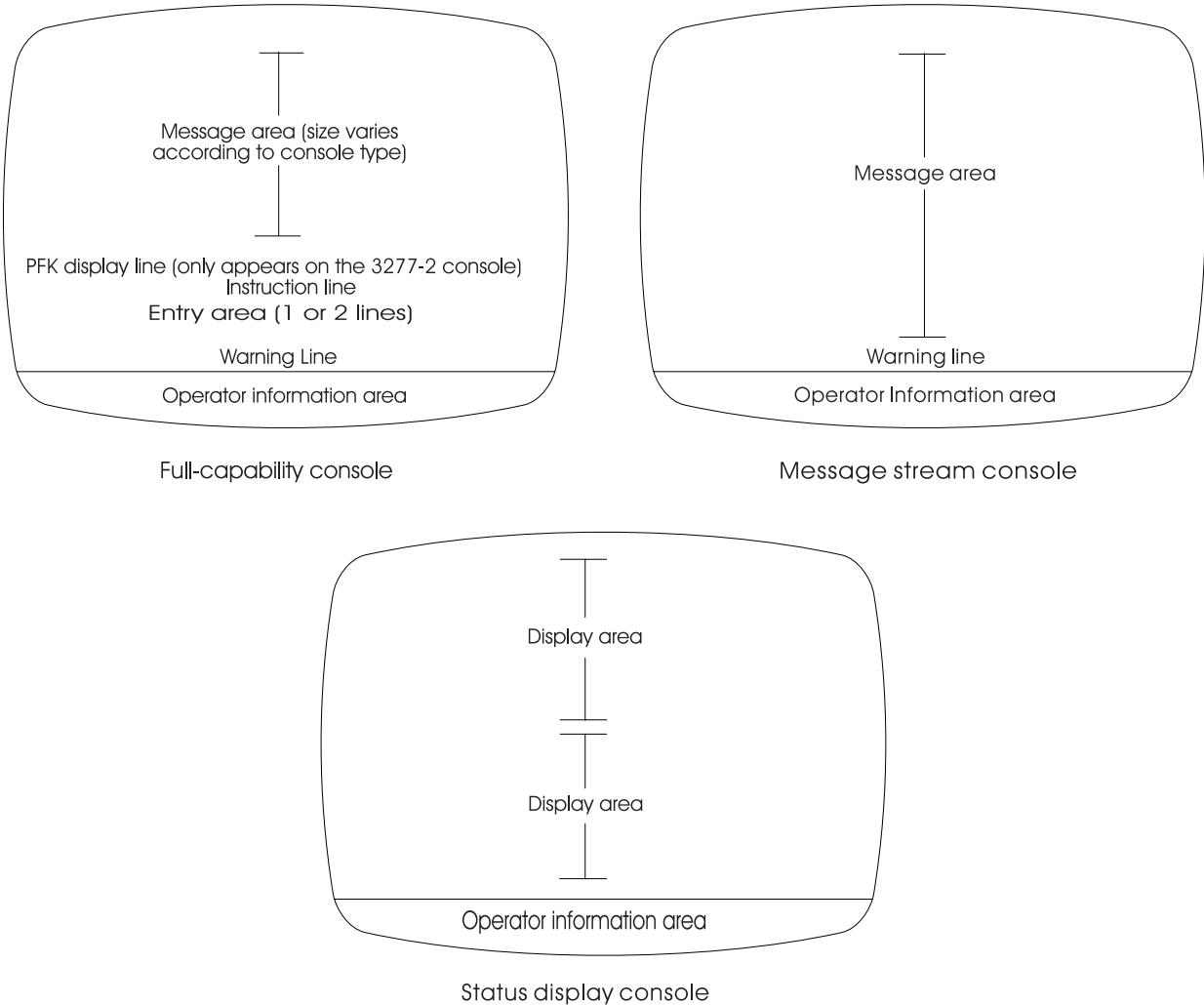


Figure 3-1. Comparison of the Display Screens of Full-Capability and Output-Only Display Consoles

L= Operand

Commands that manage consoles and console traffic use the L= operand to modify the screen area. For example, use the L= operand to delete messages or to delete lines from the screen area.

Commands that direct output use the L= operand to direct the output to an out-of-line area that is defined to the console. If there is no out-of-line area defined to the console, or if the area ID specified is z, the message is displayed inline.

For more information on the syntax and use of the L= operand for specific commands, see the description of the specific command in this book.

For a discussion of the L= operand in a sysplex, see *z/OS MVS Planning: Operations*.

Special Screen Characters

The system uses five special screen characters to indicate the status of certain screen messages. These special characters appear in position three, four, or five of the lines in the message area:

- A vertical line (|) in position three indicates that required action has been taken for the message and the system has deleted the message.
- A horizontal bar (-) in position three indicates that the message is for information only and requires no action from you.
- An asterisk (*) in position four indicates that the message is a system message that requires action from you.
- An at sign (@) in position four indicates that the message is a problem program message that requires action from you.
- A plus sign (+) in position five indicates that the message is a problem program message that requires no action from you.

Messages Sent to Display Consoles

The MVS system and any program running under the MVS system can issue messages. A displayed message can appear by itself or with information about the message. Each message consists of:

- An identifier, which is a three-letter prefix to identify the system component that produced the message and a message serial number to identify the individual message. The identifier may contain other information.
- A message text to provide information, describe an error, or request an operator action.

Messages sent to your consoles can appear in one of the following formats:

f message

or

hh.mm.ss sysname jobident f message

Fields that are always present in a message are:

f A blank, which means that no action is required, or a special screen character. See “Special Screen Characters” on page 3-6.

message Message identifier and text

Fields that you might chose to add to a message are:

jobident Job name or job id for the task that issued the message.

sysname Name of the system that issued the message

hh.mm.ss Time stamp, given as the hour (00-23), minute (00-59), second (00-59)

To add any combination of job identification, system name, and time stamp to all console messages, see “Controlling the Format of Messages” on page 3-32. For more information about console messages, use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

Operations on Display Consoles in Full-Capability Mode

Although some of the procedures for operating and controlling display consoles involve special functions and conditions, most console procedures are quite general. These general procedures are described in this topic and include:

- How to perform basic keyboard actions
- How to enter commands with the keyboard
- How to enter commands with program function keyboard
- How to enter commands with the selector pen
- How to change information in the entry area

Performing Basic Keyboard Actions

While the basic operating procedures are similar for all types of display consoles, the physical characteristics of each console require you to perform certain actions (such as, the ENTER, CANCEL, cursor detect, and selector pen detect actions) in different ways. The descriptions of operating procedures later in this section refer to these actions.

To perform the ENTER action, press the ENTER key.

To perform the CANCEL action, on a 3278 or 3279 display console, hold down the ALT key and press the PA2 key. On all other display consoles, press the CANCEL (PA2) key.

The cancel action:

- Erases the entry area
- Moves the cursor to the first position in the entry area
- Rewrites the message area and the instruction line
- Removes deletable-message indicators (if any are displayed)
- Removes message line numbers (if line numbers are displayed)

To perform a CURSOR DETECT action, position the cursor under the desired character and press the ENTER key.

To perform a SELECTOR PEN DETECT action, on 3277, 3278, or 3279 display consoles, any of which has a selector pen, place the selector pen over the desired indicator. Then, press the pen against the screen.

To retrieve the previous command, press the PA1 key.

How to Enter Commands

You can enter commands with the keyboard, the program function keys, or the selector pen (together with the PFK display line).

Entering Commands with the Keyboard

To enter commands with the keyboard through display consoles, use the following procedures. Use the same procedures to reply to WTOR messages:

1. Move the cursor to the first position in the entry area.
2. Type in the command.
3. Enter the command by performing the ENTER action.

Moving the Cursor: Move the cursor to the first position in the entry area by one of the following methods:

- Press the cursor control keys.
- Press the tab key, the back-tab key, or the new line key.

- Press the ENTER key when the cursor is in the entry area or under the ENTER indicator in the instruction line. Pressing the ENTER key passes any data in the entry area to the system.
- Perform a cancel action. This action might also change the display.

Typing the Command: Type in the command just as you would on a typewriter. As you type each character, the corresponding character appears in the entry area, and the cursor advances to the next character position. When you reach the end of the first line of a two-line entry area, the cursor advances automatically to the first character position of the next line, so that you can continue the command. The maximum number of characters that you can enter is 126.

You have the option of entering one command or several commands. When you wish to enter more than one command, use the MVS command delimiter. The MVS command delimiter is defined during system initialization. When the MVS command delimiter has not been defined during system initialization, you cannot enter more than one command at a time.

Most commands can be entered in either lowercase or uppercase. The system converts the commands to uppercase, if required. However, information within a command that is contained within single quotes (for example, a reply to a WTOR message) is not converted to uppercase by the system. If the system requires the information within the single quotes in uppercase, be sure to type it in uppercase when you enter the command. When an MVS command delimiter has been defined during system initialization, you cannot use the defined delimiter within single quotes.

Entering the Command: When you enter the command, the cursor must be in the entry area or under the ENTER indicator in the instruction line, but it need not be at the end of the command. Pressing the ENTER key or selecting the ENTER indicator causes the command to be read and processed by the system. Commands other than the CONTROL command disappear from the entry area and reappear in the message area. If the message area is full, the command may not appear immediately; to have it displayed, you may have to delete some messages.

The PA1 Key

Each time you press the PA1 key, you see a command that you entered previously. The maximum number of times you can press the PA1 key to see previous commands is specified by your installation with the RBUF option on the CONSOLxx parmlib member. If you exceed this maximum, you see the same commands again.

Correcting Command Entry Errors: If you make errors entering a CONTROL command, the audible alarm sounds, and the command appears in the entry area. The location of the cursor indicates the error:

- If the error is an invalid operand, the cursor appears under the invalid operand:
CONTROL X,N
- If the error is an invalid erase request, the cursor appears under the first invalid request.
CONTROL E,31,19
- If the CONTROL command exceeds 126 characters, the cursor appears at location 127 in the entry area.

To correct any of these errors, use the procedures described under “Changing Information in the Entry Area” on page 3-13.

If the system detects an error in a command other than a CONTROL command, it writes the command in the message area with an error message. Follow the procedures indicated for the error message in the *MVS System Messages* books.

Entering Commands with Program Function Keys

The program function keyboard is a group of keys called PFKs. They are located on or near the operator console keyboard. PFKs are used as a shortcut for entering commands. Some PFKs have commands defined for them at IPL. The definitions might be those in a PFK table that your system programmer assigned to the console, or the PFKs might have the defaults assigned by IBM. You can redefine the PFK commands; see “Defining PFKs using PFK Tables” and “Defining PFKs Using the CONTROL Command” on page 3-36 in “Defining and Changing Console Characteristics” on page 3-20.

Each PFK can be either **conversational** or **nonconversational**. The commands associated with a conversational PFK appear in the entry area one at a time when you press the key. You can change them before entering them. Commands associated with a nonconversational PFK are entered immediately when you press the key.

If your system programmer does not define and activate a PFK table for your PFKs, IBM supplies default definitions in sample IEESPFK.

Identifying PFK Definition Errors: When the system tries to execute an invalid CONTROL N,PFK command, the audible alarm sounds, and the command appears in the entry area. The location of the cursor indicates the error:

- If the cursor is positioned under the first letter of a keyword (CMD, KEY, PFK, or CON), that keyword or its trailing equal sign is incorrect.
- If the cursor is positioned under the number of the PFK being defined, that number is either not a numeric character or not the number of a PFK that was designated for command entry in the PFK table, or it is the number of a PFK you are trying to associate with a list of key numbers when it is already part of a list of key numbers.
- If the cursor is positioned under a number following the KEY operand, the key number indicated is either a non-numeric character, the number of the PFK that is being defined, the number of a PFK that has already been defined as a list of key numbers, or the number of a PFK that has no command associated with it in a PFK table.

To correct these errors, follow the procedures described under “Changing Information in the Entry Area” on page 3-13.

Checking the Commands Defined for Each PFK: Use the DISPLAY PFK command to determine the commands defined for a console’s PFKs, the PFK definitions in a specific PFK table, or the PFKs in effect for a specific console. The display can appear in the message area or can be routed to a display area or to another console. Unless you specify another console, the definitions always refer to the console on which you issue the command.

Table 3-1. Checking the Commands Defined for Each PFK

If you want to know	Use this command
The names of all available PFK tables	DISPLAY PFK, TABLE
The PFKs in effect at your console	DISPLAY PFK

Table 3-1. Checking the Commands Defined for Each PFK (continued)

If you want to know	Use this command
The definitions in a specific PFK table	DISPLAY PFK, TABLE=nnnnnnnn, where nnnnnnnn is the name of the table
The PFK definitions in effect for a specific console	DISPLAY PFK, CN=name, where name is the console name

"Summary of the PFK Definitions for the Cluster" later in this chapter shows the complete output of the DISPLAY PFK, TABLE=nnnnnnnn command.

Example 1

To display the commands associated with the PFKs on the console on which you issue the command, enter:

```
DISPLAY PFK
```

In response to this command, the following message usually appears in the message area:

```
IEE235I hh:mm:ss PFK DISPLAY  
PFK DEFINITIONS FOR CONSOLE nnnnnnnn TABLE - MASTCMDS IN PFKTAB02  
KEY# CON -----DEFINITION-----
```

The definitions for each key appear under the headings; *nnnnnnnn* identifies the console on which the command is issued.

If no PFKs are defined for the console named *CON04*, the following message appears in the message area instead:

```
IEE235I hh:mm:ss PFK DISPLAY  
NO PFK DEFINITIONS FOR CON04
```

Example 2

To determine the definitions in effect for the PFKs on CON04, enter:

```
DISPLAY PFK, CN=CON04
```

In response to this command, a message such as the following might appear in the message area:

```
IEE235I hh:mm:ss PFK DISPLAY  
PFK DEFINITIONS FOR CON04 TABLE - MASTCMDS IN PFKTABJC
```

where the PFK table in effect for console CON04 is MASTCMDS in the PFKTABJC parmlib member.

The definition for each key appears under the headings. If, however, no PFKs are defined for the console, the following message appears:

```
IEE235I hh:mm:ss PFK DISPLAY  
NO PFK DEFINITIONS FOR CONSOLE CON04
```

Entering Commands Assigned to PFKs in Conversational Mode: In conversational mode, the system causes commands assigned to PFKs to appear in the entry area. You can change and then enter them, enter them unchanged, or cancel them. The cursor appears under the third character of the command or where designated with an underscore when the PFK was assigned a command.

You can change or complete the command by positioning the cursor under the first character you want to change, typing in the change, and performing an ENTER action.

To enter commands in conversational mode,

1. Press the PFK associated with the command that you want to enter, causing the first command associated with the key to appear in the entry area.
2. According to your requirements:
 - Enter the command by performing an ENTER action. The next command associated with the PFK (if any) then appears in the entry area.
 - Change the command from the keyboard, then enter the command. (See “Changing Information in the Entry Area” on page 3-13.)
 - Cancel the command that appears in the entry area by performing a CANCEL action. The next command associated with the PFK (if any) then appears in the entry area.
 - Cancel the request initiated by the first press of the PFK by pressing any PFK while the command is still in the entry area.

The result of cancelling a request in this way is shown in the following example. In the example, PFK 1 is assigned the commands START PGM1 and START PGM2.

PFK pressed	Result
PFK 1	START PGM1 command is displayed
Any PFK	START PGM1 command is canceled, and a blank line is displayed
PFK1	START PGM2 command is displayed

Altering a command in the entry area works only for the command entry in progress; the system retains the original definition for future use of the PFK. To redefine a PFK, use the procedures described in “Defining and Changing Console Characteristics” on page 3-20 under “Defining Commands Using the CONTROL Command.”

Entering Commands Assigned to PFKs in Nonconversational Mode: Press the PFK associated with the commands that you want to enter. All of the commands are entered in the order in which they were associated with the key, just as if you had typed each command and performed the ENTER action.

Notes:

1. PFKs that are defined as conversational function in the conversational mode even though the console is in nonconversational mode. Use these keys as if you were in conversational mode, as described earlier under “Entering Commands Assigned to PFKs in Conversational Mode” on page 3-10.
2. Although the commands are entered in order, their execution may overlap. Therefore, assign commands requiring sequential execution in conversational mode.

Responses to PFK Errors: If you press a PFK that is not designated for command entry, the following message appears in the instruction line:

IEE721I PFK nn NOT SUPPORTED

If you press a PFK that has been designated for command entry but for which no command has been defined, the following message appears in the instruction line:

Displaying the PFK Numbers on 3277 Model 2 Consoles: You can display the PFK numbers on 3277-2 consoles and then point to them with the selector pen. Pointing to a number has the same effect as pressing that key. To display the PFK numbers, use the CONTROL D,PFK command. To erase the numbers in the PFK line, use the CONTROL E,PFK command.

Example

To request a display in the PFK display line (this line is located immediately above the instruction line), enter:

CONTROL D,PFK

In response to this command, a display similar to the following appears in the PFK display line:

1 2 3 4 5 6 7 8 9 10 11 12

Only those numbers that have been designated for PFK command entry appear in the display. Once you have requested this display, you can leave it on the screen; the PFK display line is not used for any other purpose, even when the key numbers are not displayed. To erase the display, enter:

CONTROL E,PFK

Entering Commands with the Selector Pen

Use the selector pen to enter commands that appear in the entry area. The commands can be in the entry area either because you typed them there or because you pressed a PFK that is in conversational mode. The PFK numbers available for selector pen command entry are defined in the active PFK table or are IBM defaults.

On a 3277 model 2, the selector pen can be used with the PFK display line to enter commands. The numbers appearing in the display line represent PFK numbers, and selecting a number with the selector pen has the same effect as pressing a PFK.

In **nonconversational mode**, all of the commands associated with a PFK are entered in the order in which they were associated with the key number. All commands (except CONTROL commands) appear in the message area when screen space is available. No commands appear in the entry area.

To enter commands on the 3277 model 2 in nonconversational mode:

1. Display the PFK numbers in the PFK display line by entering the CONTROL D,PFK command.
2. Select the PFK number associated with the command(s) you want to enter.
3. Press the selector pen against the screen over the selected number. The command is automatically entered.

To select commands on the 3277 model 2 in conversational mode, follow the same three steps. The system does not automatically enter the command; rather, the first command associated with the PFK number appears in the entry area. To enter the command, follow the steps described in the next section.

Entering Commands with the Selector Pen in Conversational Mode: In conversational mode, each command associated with a PFK number is presented in the entry area, one command at a time, where you can enter it as is, change it

and enter it, or cancel it. Changing a command in the entry area works only for the command entry in progress; the system retains the original definition for that PFK.

To enter commands with the selector pen in conversational mode:

1. Enter the command by performing the ENTER action or by selecting ENTER. The next command associated with the PFK (if any) then appears in the entry area.
2. Change the command from the keyboard before entering it as described later in this chapter under "Changing Information in the Entry Area" on page 3-13.
3. Cancel the command in the entry area by performing a CANCEL action. The next command (if any) then appears in the entry area.
4. Cancel the request initiated by the first selection of the PFK number by pressing the selector pen against the screen over any other PFK number while a command associated with the first key number is still in the entry area.

Changing Information in the Entry Area

You can change information in the entry area to correct a typing error or to change a command during conversational command entry or message deletion. You might not need to completely retype a command to correct or change it. (Both conversational command entry and message deletion are described in this section.) You can blank out the entry area without entering a command to the system.

Pressing the PA1 key displays a command that you entered previously. When you see that command, you can make corrections or changes (as described in this section) and press the Enter key to issue the command.

Substituting Characters: If you make a mistake when typing in the entry area move the cursor to the first character you want to change and type the correct characters.

Example

If you type in the following reply to a WTOR message:

R 22,'DISPLAY REQUESTED'_

and then note (before performing the enter action) that you have typed the word DISPLAY incorrectly, you can move the cursor under the L, and type PL. The reply then reads:

R 22,'DISPLAY REQUESTED'

In the same example, if you decide that the correct response is NO, moving the cursor under the D in DISPLAY and typing NO leaves the following in the entry area:

R 22,'NO'PLAY REQUESTED'

To correct this situation, move the cursor under the P and press the ERASE EOF key. This key erases the remainder of the entry area (from the cursor to the last character position), leaving the following in the entry area:

R 22,'NO'_

Inserting Characters: To insert one or more characters within data in the entry area:

1. Position the cursor at the character position following the point where the missing data should appear.

2. Press the INS MODE key (the insert mode marker appears on the console).
3. Type in the missing data.
4. On some consoles, you must press the RESET key to return the keyboard to its normal input mode.

Example

To insert the console identifier 10 in the following command:

```
DISPLAY JOBS,L=CONSC
```

Move the cursor back to the C, press the INS MODE key, type in 10, and press the RESET key. The command then reads:

```
DISPLAY JOBS,L=CONS10C
```

Note that the characters to the right of the inserted characters shift to make room for the inserted characters. If required, characters shift to the second line of the entry area.

Deleting Characters: To delete a character, position the cursor at the character to be deleted and press the DEL key.

All characters that follow the deleted one shift to the left to fill the space formerly occupied by the deleted character. Delete one character at a time.

Example

To delete the extra S from the following command:

```
DISPLAY JOBS,L=CONSSOLEC
```

Position the cursor at either S and press the DEL key. The command then reads:

```
DISPLAY JOBS,L=CONSOLEC
```

Blanking the Entry Area

The ERASE INPUT Key

To remove all of the data that you have typed in the entry area without causing it to be passed to the system, press the ERASE INPUT key. This key erases the entry area and moves the cursor to the first position in the entry area.

Note: Do not use the ERASE INPUT key on the 3279 models 2A, 2C, and 3A. On these devices, the ERASE INPUT key blanks out the entry areas and all fields with data displayed in red.

The PA2(CANCEL) Key

To clear the entry area and restore the screen, press the PA2 key.

Handling Consoles in Error Conditions

Several types of errors can occur that directly affect the operation of display consoles. In some cases, the error becomes apparent by a sudden screen failure, the appearance of error messages, or the locking of the keyboard. In other cases, the error might not be immediately apparent. Errors can be caused by a programming problem (system error), a console malfunction (hardware error), or a hardware error not related to the console.

System Errors

When a system error occurs, one or more of the following can happen:

- The screen is blanked out, and then an error message appears in the message area
- An error message appears in the WARNING line.
- There is an abnormal lack of console activity.

Responding to an Error Message in the Message Area: An error message at the bottom of the message area indicates that a recoverable system error has occurred. Perform the action specified by the error message, and then perform a CANCEL action. This should restore the screen. It is good practice to review the messages at this time to make sure that no messages were lost during error recovery.

Responding to an Error Message in the WARNING Line: An error message in the WARNING line might indicate that an unrecoverable system error has occurred and that the system needs to be loaded again. If so, follow normal procedures for IPL, and notify your system programmer.

Responding to an Inactive Console: An inactive console condition is characterized by a lack of message traffic or a lack of system response to commands. The inactivity could be caused simply by a low level of system activity, or it could be the result of a problem in the message handling portion of the control program.

If an MCS or SMCS console appears inactive, check the system response by requesting a display of the time:

```
DISPLAY T
```

The system should respond within a few seconds with the time and date. If it does not, perform one of the following actions:

- Issue the CONTROL C,D command to cancel any status displays being presented on the inactive console.

If neither of these procedures returns the console to normal activity, assume that there is some other problem related to the console. Check for a console hardware error. If the system must be loaded again, follow normal procedures for IPL. Report the occurrence of this problem to your system programmer.

Console Hardware Errors

When a console hardware error occurs, one or more of the following can happen:

- Error messages are centered on the screen (the remainder of the screen is blank).
- The screen is blank (and no error message appears).
- The screen appears normal, but the keyboard is locked and you cannot enter commands.

Responding to Error Messages Centered on the Screen: If a console hardware error occurs, one of the following sets of messages can appear centered on the screen:

IEE170E RETRYABLE ERROR. RECENT ACTION MAY NEED TO BE REPEATED
IEE170E PRESS THE CANCEL KEY TO RESTORE THE SCREEN

-- or --

IEE171E CONDITIONAL ERROR. RECENT ACTION MAY NEED TO BE REPEATED
IEE171E PRESS CANCEL TO CONTINUE

Perform a CANCEL action. The CANCEL action should restore most of the screen, including messages displayed inline in the message area, the instruction line, and the warning line. The entry area and the PFK line, however, are blanked out, any out-of-line displays are erased, and the cursor is positioned to the first data entry position. Also, message numbering (if active) is terminated.

Note: If you do not perform a CANCEL action, the system rewrites the screen (same effect as CANCEL) after about 30 seconds. If a console hardware error results from keyboard input when you perform the CANCEL action, the system sees the error as a permanent I/O error.

Responding to a Blank Screen: If the console screen goes blank, the console has failed.

Note: It is normal for the screen of a 3277 to go blank for a few seconds if the back-tab key is pressed when the cursor is not in the entry area.

Responding to a Locked Keyboard: Sometimes the system is unable to blank out the screen. If you find that you cannot enter commands through a console that otherwise appears normal, try to restore the screen by performing a CANCEL action.

Note: Inhibited input, with or without keyboard locking, can also occur when the system abends or goes into a wait state, or when a problem occurs in the message handling portion of the control program. See the procedures described for an inactive console under "System Errors" on page 3-15.

Responding to Console Message Backups

The MVS system keeps some WTO and WTOR messages in buffers in virtual storage. The WTO buffers hold the messages that the system has not yet displayed at the eligible consoles; the WTOR buffers each hold one WTOR message that the system has already displayed but that an operator has not responded to. The maximum number of WTO and WTOR buffers are determined by the MLIM and RLIM parameters on the INIT statement in the CONSOLxx parmlib member. If these parameters are not coded, the system defaults (as described in *z/OS MVS Initialization and Tuning Reference*) are in effect.

To avoid WTO message buffer shortages, you can raise your WTO buffer limit (MLIM) and adjust message deletion specifications on your consoles. To avoid WTOR message buffer shortage, raise your WTOR buffer limit (RLIM) and reply to WTORs more frequently. Procedures for responding to WTO and WTOR buffers shortages follow in this section.

Responding to WTO Buffer Shortages: When WTO message buffer use reaches 80 percent of the limit specified at IPL, the system issues the following message:
IEA405E WTO BUFFER SHORTAGE - 80% FULL

The system also issues a DISPLAY CONSOLES,BACKLOG (D C,B) command to provide information helpful in determining the cause of the buffer shortage.

If the problem continues and WTO buffer use reaches its limit, the system issues the following action message:

IEA404A SEVERE WTO BUFFER SHORTAGE - 100% FULL

When MLIM is reached, the system obtains buffers in backup storage. When this backup storage is exhausted, the system issue the following message:

At this point, any new WTOs will be thrown away.

When the system notifies you that the WTO buffers are 80% full, determine the reason for the buffer shortage and correct the problem. Possible reasons are:

- A console is not ready and WTO messages are filling the console message buffers because:
 - An intervention required condition exists.
 - The console has been powered off.
 - Some part of the path to the device is not working; for example, an I/O interface is disabled.
 - One or more consoles may have their displays held.
- A console is not in roll mode, and messages are filling the console message buffers.
- A console is in roll or wrap mode but the update time is too long, and messages are filling the console message buffers.
- A buffer limit specified at IPL is too low to handle the message traffic in the system. (Either the value on the MLIM parameter in the CONSOLxx member is too low, or the system default for RLIM is too low.)
- A program is issuing messages at too rapid a rate and might be in a loop. When a job uses a high percentage of the WTO buffers, the system issues message CNZ3011I which identifies the jobname and the address space.

To determine the extent of the problem and the responsible console or consoles, examine the output from the DISPLAY CONSOLES,BACKLOG (D C,B) command. When messages are backed up for a console, it might be necessary to delete the queue of messages for the console using a CONTROL Q command. You might need to issue CONTROL Q several times to clear the console completely.

When there are too many messages from one job/address space, consider cancelling the job or jobs specified in message CNZ3011I. If cancelling a job would cause a serious impact, look at the messages the job is issuing. If the job seems to be in a loop, then activate an MPF member to suppress or delete the repeating message. Another option is to temporarily remove the message's routing code from all the consoles.

When a high number of buffers is in use for messages from another system in the sysplex, you can route a D C,B command to the other system to determine if a job on the other system is generating too many messages. You can protect your system from a runaway job on another system in the sysplex by using the V CN,DMSCOPE= command.

Figure 3-2 shows an example of the DISPLAY CONSOLES,BACKLOG output. The system displays information about all consoles, on this system only, that have any outstanding WTO messages. The output in the figure includes the following line:

```
MSG: CURR=1356 LIM=1500 RPLY:CURR=1 LIM=10 SYS=1 PFK=NONE
```

In this line, MSG: CURR=1356 LIM=1500 tells you the current use of WTO buffers and the specified limit. RPLY: CURR=1 LIM=1500 tells you the number of WTOR messages that have been displayed and are awaiting operator reply, and the specified limit. The line confirms that more than 80% of the specified WTO buffer limit is reached; 1356 WTO buffers are full and the specified limit is 1500. The display in Figure 3-2 on page 3-18 indicates, through NBUF, the number of buffers

queued to each console. In this example, CON12, with 1217 message buffers filled, is the source of the problem. The buffer limit of 1500 seems adequate, so CON12 is probably failing and causing undisplayed messages to fill the message buffers.

```
SY2 IEE889I 15.03.36 CONSOLE DISPLAY 198
MSG: Curr=1356 LIM=1500 RPLY:Curr=1 LIM=10 SYS=1 PFK=None
CONSOLE ID ----- SPECIFICATIONS -----
CON12 12 COND=A AUTH=MASTER NBUF=1217
03E0 AREA=Z,A MFORM=S
SY2 DEL=RD RTME=2 RNUM=5 SEG=10 CON=N
USE=FC LEVEL=ALL PFKTAB=*DEFAULT
ROUTCDE=ALL
CMDSYS=1
MSCOPE=*&ALL
MONITOR=JOBNAMES
WTO BUFFERS IN CONSOLE BACKUP STORAGE = 0
ADDRESS SPACE WTO BUFFER USAGE
ASID - 0019 JOBNAME = FLOODNUM NBUF = 520
MESSAGES COMING FROM OTHER SYSTEMS - WTO BUFFER USAGE
SYSTEM = 2 NBUF= 4
```

Figure 3-2. Example of DISPLAY CONSOLES,BACKLOG Command Output

If the buffer limit is not adequate, issue the CONTROL M,MLIM= command to increase the WTO buffer limit for the duration of the IPL. Your system programmer might code the MLIM parameter on the INIT statement in the CONSOLxx member to raise the WTO buffer limit for the next IPL.

When the number of buffers in use drops below 60% of the limit specified at IPL time, the system issues the following message:

IEA406I WTO BUFFER SHORTAGE RELIEVED

Notes:

1. All lines of an out-of-line multi-line status display that have not been presented occupy message buffers. Therefore, you should erase these displays when they are no longer needed.
2. The current buffer count can be larger than the specified limit. Even though the buffer count is greater than or equal to the limit, the system always gives a privileged task a buffer unless the storage available for buffers is exhausted.
3. The system does not use the MLIM and RLIM parameter values specified in the CONSOLxx parmlib member until either the hardcopy medium (SYSLOG or OPERLOG) becomes active or NIP processing is complete. After NIP processing, multiple consoles become active and buffer space becomes important.

Responding to WTOR Buffer Shortages: When WTOR message buffer use reaches 80 percent of the limit specified at IPL, the system issues the following message:

IEA230E WTOR BUFFER SHORTAGE - 80% FULL

If the problem continues and WTO buffer use reaches its limit, the system issues the following action message:

IEA231A WTOR BUFFER SHORTAGE CRITICAL - 100% FULL

When the system notifies you that the WTOR buffers are 80% full, you should reply to the WTOR messages that are outstanding. If any of the WTORs have rolled off

the screen (console roll mode is DEL=R), use the DISPLAY R,R command to retrieve the text of the outstanding requests.

To raise the limit of WTOR buffers for the duration of the IPL, issue the CONTROL M,RLIM command. If WTOR buffer use often reaches 80 percent of the limit, the limit for WTOR messages specified at IPL might be too low to handle the WTOR message traffic in the system. Your system programmer should code the RLIM parameter on the INIT statement in the CONSOLxx member to raise the WTOR buffer limit for the next IPL.

Processing MVS Messages at the System Console During System Recovery

During system recovery, MVS might try to communicate with you. Your installation might have defined the system console, or any other MCS console as members of a console group in CNGRPxx to receive synchronous messages. Synchronous messages are WTO or WTOR messages that can be issued during initialization or recovery situations. The operator must respond to the WTOR messages before the system will continue. In a sysplex, a console can display synchronous messages only if it is attached to the system that issues the message. If your installation has not specified a console group or a console is not active, the system that issues the message tries to select a console in the following order:

- The system console on the system that issues the message:
 - If the system hardware level supports the OPRMSG system console, MVS will display the message on the OPRMSG priority frame.
 - Otherwise, MVS displays a WTOR message on the SCPMSG or SYSMSG system console. When this action is successful, the alarm on the processor controller rings, and a message replaces the screen image on the system console. The message tells you that an MVS message is pending. To display this MVS message, enter one of the following commands on the system console:
 - F OPRMSG (on an ES/9000 processor)
or
 - F SCPMSF (on an ES/3090 processor)
or
 - F SYSMSG (on a 308x processor)

The MVS message on the system console does not time out; the message remains on the screen until you enter a reply. See *z/OS MVS Planning: Operations* for more information about consoles and console recovery.

Placing a Console in Offline Status

When an MCS or SMCS console or the system log must be bypassed for any reason, you must enter a VARY command to place the console offline. Command activity from the console is immediately suspended. If the console is a printer, messages continue to be displayed until all waiting messages have been issued.

The VARY command does not cause the functions of the bypassed console to be assigned to another console.

Interchanging Your Consoles on a Control Unit

If a device has been specified as a 3270 model X to hardware configuration definition (HCD), you can replace it with another device and redefine it through the HCD panels. For information about using HCD, see *z/OS HCD User's Guide*.

Defining and Changing Console Characteristics

This chapter describes:

- Using Operator Commands to Change CONSOLxx Statements
- Changing Console Characteristics
- Controlling System Messages and Commands
- Defining Program Function Keys (PFKs)
- Processing Hardcopy

When your system comes up, the definitions in certain members of SYS1.PARMLIB are in effect. After IPL, you can use CONTROL, MONITOR, SET, and VARY commands to change some of the definitions; however, the effect of the command lasts only for the duration of the IPL.

Using Operator Commands to Change CONSOLxx Statements

Several operator commands are available to modify the statements in the CONSOLxx parmlib member.

Potential Effects of Altering Console Attributes

Altering some console attributes might cause a message loss or SYNCHDEST loss. If a loss occurs, MVS issues a DISPLAY CONSOLE,HONLY command and message IEE889I. You need to understand that this can happen and can affect automation.

The potential for this situation to occur comes from using these commands:

VARY CN
VARY CONSOLE
CONTROL V,LEVEL

The CONSOLE Statement of CONSOLxx

A CONSOLE statement in the CONSOLxx parmlib member establishes the device as an MCS or SMCS console and defines certain console values or attributes. These values are specified by system programmers or are IBM defaults. After IPL, operators can use certain commands to change these attributes. The effects of these commands last only for the duration of the IPL; at the next IPL, the values will be those in the parmlib members or the IBM defaults.

CONSOLxx contains console definitions for the system or sysplex.

Table 3-2. Comparison of System Commands and CONSOLE Parameters in CONSOLxx

MVS Commands	CONSOLE Parameters with DEFAULT	Characteristic that the Parameter Affects
CONTROL A	AREA	Size of the out-of-line display areas
CONTROL N,PFK	PFKTAB	PFK table
CONTROL S,CON	CON(N)	Conversational or nonconversational mode
CONTROL S,DEL	DEL(RD)	Message deletion mode
CONTROL S,MFORM	MFORM(M)	Format in which the messages appear
CONTROL S,RNUM	RNUM(5)	Number of message lines included in one message roll
CONTROL S,RTME	RTME(2)	Number of seconds between message roll/wrap

Table 3-2. Comparison of System Commands and CONSOLE Parameters in CONSOLxx (continued)

MVS Commands	CONSOLE Parameters with DEFAULT	Characteristic that the Parameter Affects
CONTROL S,SEG	SEG	Number of lines in the message area that can be deleted by a CONTROL E,SEG command
CONTROL V,CMDSYS	CMDSYS	Systems where commands on a console can be directed for processing
CONTROL V,LEVEL	LEVEL	Message levels for the console
CONTROL V,USE	USE(FC)	Console operating mode
MONITOR	MONITOR	Monitoring of certain events
VARY CN,AMSCOPE VARY CN,DMSCOPE VARY CN,MSCOPE	MSCOPE	Systems that direct messages to a console
VARY CN,AUTH	AUTH(INFO)	Command groups
VARY CN,LOGON	LOGON	Defines the LOGON attribute
VARY CN,LU	LU	Defines the predefined LU for an SMCS console only
VARY CN,ROUT VARY CN,AROUT VARY CN,DROUT	ROUTCODE	Routing codes for the console

The INIT Statement in the CONSOLxx Member

The INIT statement contains initialization values for the system. You code only one INIT statement in the CONSOLxx member for all the consoles.

Table 3-3 describes each MVS command that has a corresponding parameter on the INIT statement in CONSOLxx, the parameter, and the characteristic that the command and parameter affect. The value in parentheses indicates the default.

Table 3-3. Comparison of System Commands and INIT Statements in CONSOLxx

MVS Command	Parameter on INIT Statement with default value	Characteristic that the Parameter Affects
CONTROL M,AMRF	AMRF(Y)	Establishes whether the action message retention facility is to be active
CONTROL M,APPLID	APPLID	Sets the APPLID used by SMCS on this system
CONTROL M,GENERIC	GENERIC	Sets the GENERIC used by SMCS for the entire sysplex
CONTROL M,MLIM	MLIM(1500)	Limits the number of buffers for WTO messages that the system has not yet displayed
CONTROL M,LOGLIM	LOGLIM(1000)	Limits the number of buffers for messages that the system sends to the system log

Table 3-3. Comparison of System Commands and INIT Statements in CONSOLxx (continued)

MVS Command	Parameter on INIT Statement with default value	Characteristic that the Parameter Affects
CONTROL M,RLIM	RLIM(10)	Limits the number of WTOR messages that the system has displayed but that the operator has not replied to
CONTROL M,UEXIT	UEXIT(Y)	Establishes whether the installation exit IEAVMXIT is to be active
MONITOR	MONITOR	Establishes how the system displays mount and demount messages in response to the MONITOR command
SET CNGRP	CNGRP(NO)	Specifies the CNGRPxx parmlib members that the system is to use
SET MMS	MMS(NO)	Specifies the MMSLSTxx parmlib member that holds the translation tables that are available for your system
SET MPF	MPF(NO)	Specifies the MPFLSTxx parmlib members that the system is to use
SET PFK	PFK(NONE)	Specifies the PFKTABxx parmlib member that holds the PFK tables that are available for your consoles
TRACE CT,PARM=	CTRACE(CTIOPS00)	Specifies the CTnOPSxx parmlib member that contains tracing options for the operations services (OPS) component

The HARDCOPY Statement in the CONSOLxx Member

Table 3-4 describes each VARY HARDCPY command operand, the corresponding parameter in CONSOLxx parmlib member, and the task the command and parameter performs. The value in parentheses indicates the default.

Table 3-4. Comparison of VARY HARDCPY Commands and HARDCOPY Statements in CONSOLxx

VARY HARDCPY Command Parameters	Parameters on HARDCOPY Statement	Description
SYSLOG or OPERLOG	DEVNUM	Establishes whether the hardcopy medium is SYSLOG or OPERLOG
ROUT	ROUTCODE	Establishes the routing codes for messages included in the hardcopy message set
NOCMDS, INCMDS, STCMDS, or CMDS	CMDLEVEL	Establishes whether the hardcopy message set includes operator commands, responses, or status displays

The HARDCOPY statement is optional; CONSOLxx contains only one statement for each system. If the HARDCOPY default is used, the system uses the following defaults:

- The hardcopy medium is SYSLOG.
- The system uses a minimum set of routing codes (1, 2, 3, 4, 7, 8, 10, and 42) to select messages for the hardcopy message set.
- CMDLEVEL(CMDS) is used to select the level of commands included in the hardcopy message set.

The DEFAULT statement in CONSOLxx: The system programmer uses the DEFAULT statement to control certain default values for MCS and SMCS consoles in the configuration. DEFAULT lets the system programmer specify console attributes that control the following for console configuration:

- Console security by specifying operator logon options
- Certain console screen functions for all consoles (ability for operators to hold moving or wrapping messages on the screen)
- Routing for messages without routing codes or other message queuing information, and routing for synchronous messages that bypass normal message queuing
- Determining the maximum value for operator REPLY ids.

Unlike values in CONSOLE and INIT, operators cannot change individual DEFAULT statement values. Operators must re-IPL the system with the CONSOLxx member that contains the new DEFAULT statement.

Displaying Information About Console Characteristics

To learn the current characteristics of the console, use the DISPLAY CONSOLES,A command. The output is message IEE889I, which contains information about the system's use of consoles as well as information about each console's characteristics. Figure 3-3 shows the output of the command. For a complete description of message IEE889I, use LookAt (see "Using LookAt to look up message explanations" on page xviii) or use the *MVS System Messages* books.

```
| SY2 IEE889I 14.51.23 CONSOLE DISPLAY 636 C
| MSG: Curr=1 LIM=1500 RPLY:Curr=0 LIM=99 Sys=SY2 PFK=None
| CONSOLE ID ----- SPECIFICATIONS -----
| SYSLOG COND=H AUTH=CMDS NBUF=N/A
|          ROUTCDE=ALL
|          C3E0SY1 01 COND=A AUTH=MASTER NBUF=N/A
|          03E0 AREA=Z MFORM=S
|          SY1   DEL=RD RTME=1 RNUM=19 SEG=9 CON=N
|          USE=FC LEVEL=ALL PFKTAB=*DEFAULT
|          ROUTCDE=ALL
|          CMDSYS=SY1
|          MSCOPE=*ALL
|          MONITOR=JOBNAMES,SESS
|          C3E0SY2 21 COND=A AUTH=MASTER NBUF=1
|          03E0 AREA=Z MFORM=S
|          SY2   DEL=RD RTME=1 RNUM=19 SEG=19 CON=N
|          USE=FC LEVEL=ALL PFKTAB=*DEFAULT
|          ROUTCDE=ALL
|          CMDSYS=SY2
|          MSCOPE=*ALL
```

Figure 3-3. Example of DISPLAY CONSOLES,A Command Output

Changing Console Characteristics

You can change the characteristics of MCS and SMCS consoles dynamically through MVS commands.

System Commands Grouped According to System Command Authority

If an MVS operator command is not RACF-protected (for example, if the RACF OPERCMDS class is not active, or if no OPERCMDS profile covers the command), the authority to issue the MVS command is granted based on the command group.

There are five command groups:

- Informational commands (INFO)
- System control commands (SYS)
- I/O control commands (IO)
- Console control commands (CONS)
- Master level authority commands (MASTER)

If RACF is used to control who can issue commands, the RACF OPERCMDS settings override the command group (AUTH) settings. For example, if the user has access to the correct OPERCMDS profile, a job submitted in a class with AUTH(INFO) will issue a MODIFY command. Similarly, if the user does not have access to the proper OPERCMDS profile, a job submitted in an AUTH(ALL) jobclass will be unable to issue a MODIFY command.

The commands in each group are shown in Table 3-5. The command groups are ordered from the lowest to the highest JES authority level, as described in *z/OS JES2 Commands* or *z/OS JES3 Commands*.

You can enter informational commands from any full-capability console. However, to enter system control, I/O control, or console control commands from a secondary console, that particular command group must be assigned to that console. If you enter a command at a console where it is not authorized, MVS rejects the command and sends an error message to the issuing console.

At a master authority console, you can enter all operator commands. Any console with AUTH(MASTER) in the CONSOLxx parmlib member has master authority.

Using RACF, the installation can allow the operators to log on to any MCS or SMCS console. IBM recommends logon for SMCS. The operator's RACF profile and group authority determines what commands can be issued from the console. For a list of MVS commands and their profile names, see *z/OS MVS Planning: Operations*.

Table 3-5. Command Groups Used to Determine Command Authority

Command Group	Commands	
INFO	CMDS DISPLAY CMDS SHOW CONTROL (See Note 3 on page 3-25) DEVSERV DISPLAY (See Note 1 on page 3-25) LOG LOGOFF LOGON MONITOR	REPLY (See Note 4 on page 3-25) ROUTE SEND STOPMN

Table 3-5. Command Groups Used to Determine Command Authority (continued)

Command Group	Commands	
SYS (system control)	ACTIVATE CANCEL CHNGDUMP DUMPDS HALT (See Note 2) HOLD LIBRARY MODE MODIFY PAGEADD PAGEDEL RELEASE RESET SET	SETAPPC SET GRSRNL SETETR SETIOS SETLOAD SETOMVS SETPROG SETRRS CANCEL SETRRS SHUTDOWN SETSMF SETSMS SLIP START STOP SWITCH SMF TRACE (with CT, ST, or STATUS) WRITELOG
IO (I/O control)	ASSIGN MOUNT SWAP UNLOAD	VARY {NET } (See Note 2) {OFFLINE} (See Note 5) {ONLINE } (See Note 5) {PATH } {name or [/]devnum}
CONS (console control)	CONTROL (See Note 3)	{OFFLINE} (See Note 5) {ONLINE } (See Note 5) {PATH } {name or [/]devnum }
MASTER (master control)	CMD\$ ABEND CMD\$ REMOVE CONFIG CONTROL (See Note 3) DUMP FORCE IOACTION QUIESCE RESET CN SETCON SETGRS SETLOGR SETLOGRC SETSSI SETXCF	TRACE (with MT) VARY {CN(...)[,AUTH=...]} {CN(...)[,LOGON=...]} {CN(...)[,LU=...]} {CONSOLE[,AUTH=...]} {GRS } {HARDCPY } {OFFLINE,FORCE } {XCF }
Notes:		
<ol style="list-style-type: none"> 1. CONS command group when message routing is specified. 2. HALT NET and VARY NET are related to the Virtual Telecommunications Access Method (VTAM) 3. CONTROL is in the INFO command group except when <ul style="list-style-type: none"> • Purging the message queues of any other full-capability MCS or SMCS console — MASTER. • Message routing is specified — CONS. • Changing or displaying the status of the action message retention facility — MASTER. • Changing or displaying the number of allowed message buffers — MASTER. • Changing or displaying the status of WTO user exit IEAVMXIT — MASTER. • In a sysplex, changing the maximum time to wait for aggregated command responses — MASTER. • Increasing the number of reply IDs — MASTER. 4. An operator can reply to any message that the console is eligible to receive. Any console with master authority can reply to any message. 5. VARY CN,OFFLINE and VARY CN,ONLINE require CONS. Without the CN keyword, VARY OFFLINE and VARY ONLINE require IO authority. 		

Changing the Authorization of a Console

You can change the system command groups that a console is authorized to enter.

You change the authorization of consoles by:

- **Using the VARY Command:**

The VARY CN,AUTH= command defines which system command groups may be entered through the consoles specified on the AUTH= keyword.

Example

To assign master level authority to a console named REMOTE, enter:

```
VARY CN(REMOTE),AUTH=MASTER
```

Enter this command through any console that has master authority. If you try to enter this command from a console without master authority, the command is rejected and a message appears to indicate that the switch did not take place.

The effect of this command lasts only for the duration of the IPL.

Defining Console Use

MCS consoles can operate in one of the following ways:

- Status Display Console
- Message Stream Console
- Full-capability Console

Note: In this book, the term output-only mode refers to status display mode and message stream mode.

Note: SMCS consoles are not permitted to be status display or message stream consoles. SMCS consoles may only be full-capability consoles.

Using a Status Display Console: A status display console has output capability only; it cannot be used to enter commands. The system uses the screen to receive status displays.

A console in status display mode provides a convenient area for displaying system status information and frees a full capacity console for use by other system messages.

You can divide the screen of the status display console into display areas, according to your needs.

Controlling Displays on Status Display Consoles: Because a status display console has no input capability, you must enter each request concerning the console on a separate full-capability console. Use the routing location operand with each command to designate the console and display area at which an action is to take place.

The routing location operand can be entered only from a console with CONS (console control) command group authority. Command group authority is described under "System Commands Grouped According to System Command Authority" on page 3-24.

Using a Message Stream Console: A message stream console has output capability only; it cannot be used to enter commands. The system uses the screen to present general messages.

A console in message stream mode provides an area for presentation of messages. The messages sent to a message stream console depend on the routing codes or message levels assigned to that console. Message stream consoles can provide system monitoring capabilities in tape or disk libraries, or can assist in system security.

Deleting Messages from Message Stream Consoles

When a console enters message stream mode, roll-deletable message deletion goes into effect automatically. (See “Defining Automatic Message Deletion” later in this section.) All messages except action messages are automatically removed from the screen.

Using a Full-Capability Console

A full-capability console has both input and output capability; the console can be used both to enter commands and to receive status displays and messages. There can be many full-capability consoles in the system or sysplex.

You can divide the screen on a full-capability console so that part of the screen receives general messages and the other part receives status displays. When a status display is not on the screen, MCS uses the status display area for general messages.

Changing Full-Capability to Message Stream or Status Display Mode

The screens of the message stream console and the status display console appear identical; they do not have any entry area. However, the screens of the consoles in message stream mode receive general messages and the screens of the status display consoles receive formatted status displays.

When you change a full-capability console to message stream or status display mode, the PFK display line, the instruction line, and the entry area are incorporated into the message area or the display area. Figure 3-4 shows the 3277 model 2, in message stream mode. Once a display console enters message stream or status display mode, it can accept no more input; you must use another console to enter commands. Examples at the end of this section illustrate how the display on a full-capability console changes to the display on a status display or message stream console.

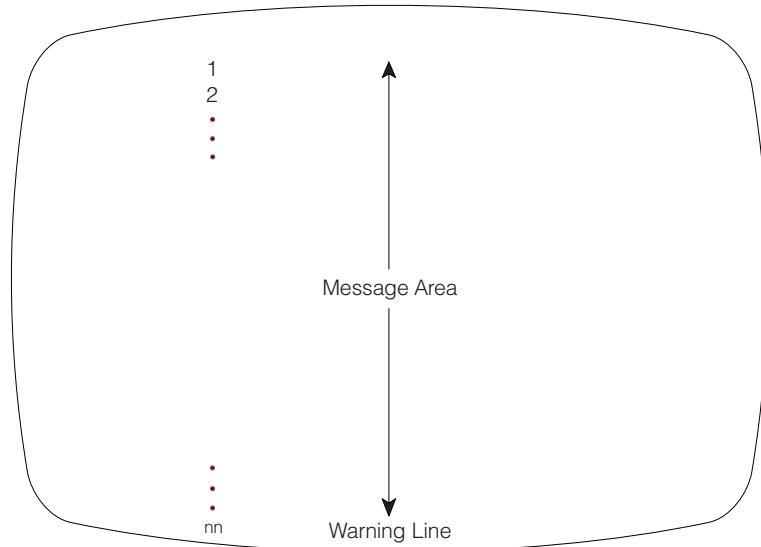


Figure 3-4. Format of a Console Screen in Message Stream Mode

The system gives you the following choices for operating mode for MCS consoles:

- FC** Full-capability
- MS** Message stream
- SD** Status display

SMCS consoles may only be FC (full-capability) mode consoles. The operating mode of an SMCS console cannot be changed.

If a console is an input/output device, the default operating mode is full-capability mode.

You can check the console operating mode by entering the CONTROL V,REF command. In response to this command, the specifications appear in the entry area. You can change the specifications using the procedures described under "Changing Information in the Entry Area" on page 3-13.

You define the operating mode of a console by:

- **Using the CONTROL Command:**

Use the USE operand on the CONTROL V command to change the operating mode of a console.

Example 1

To define the console with a console name of CON8 as a full-capability console, enter:

CONTROL V,USE=FC,L=CON8

The effect of this command lasts only for the duration of the IPL.

Note: When you use the CONTROL command to change the console operating mode, you might also have to change other console characteristics. If the new definition for the console operating mode is incompatible with other characteristics, the system rejects the CONTROL command.

Example 2

To change the console in Example 1 from full-capability mode to status display mode, enter:

```
CONTROL V,USE=SD,L=CON8
```

In response to this command, any information on the screen disappears, and the system reestablishes the display area specifications that were defined in the CONSOLxx parmlib member. If you were changing the console from full-capability mode to message stream mode, information on the screen would disappear and the message area would expand, as in Figure 3-4 on page 3-28.

Example 3

To return CON8 to full-capability mode, enter the following command from a full-capability console:

```
CONTROL V,USE=FC,L=CON8
```

In response to this command, the message area of the console with a console name of CON8 returns to its full-capability size, and the console specifications return to those established the last time the console was in full-capability mode for this IPL or those established in the CONSOLxx member.

The display area specifications also return to the specifications established the last time the console was in full-capability mode.

Controlling System Messages and Commands

Messages are the system's chief means of communication with you. Messages range from **informational**, which are important but do not require a response, to **immediate action**, which are not only important but require that you perform the requested action at once. The action might be required because the message issuer waits until the action is performed, or because taking the action as soon as possible can improve system performance. Less urgent, but still important, are the **eventual action** and **critical eventual action** messages. The message issuer is not waiting for you to perform the action, but a number of unanswered requests might degrade system performance.

The size of the screen's message area varies, depending on the type of display console. When the message area becomes full, you need to delete messages to make room for new ones. You can delete messages, or have the system do it for you automatically. (See "Deleting Messages from the Console Screen" later in this chapter.) Once an action message is deleted from the screen, you cannot see the entire message again unless the action message retention facility is active and you have issued a DISPLAY R command.

So that you do not have to delete messages too often, make sure that you manage message traffic carefully on all consoles. For example, if you find that a console screen often fills with action messages, think about:

- Adjusting routing codes and assigning message levels. Any console should receive only messages for which the operator of that console is directly responsible.
- Activating the action message retention facility so you can put the console in roll mode without losing action messages.

Defining Routing Codes for a Console

Most messages have one or more routing codes. The system uses these codes, decimal numbers from 1 to 128, to determine which console or consoles should receive a message. The system programmer assigns routing codes to the consoles attached to your system so that a specific message type is routed to the proper console. Table 3-6 lists the routing codes.

Routing codes do not appear with a message at a console; routing codes 1 through 28 do, however, appear on the system log. To determine the routing codes each console receives, use the DISPLAY CONSOLES,A command. Figure 3-3 on page 3-23 shows the display that appears in response to this command.

Table 3-6. Message Routing Codes

Code	Definition
1	Primary operator action
2	Primary operator information
3	Tape pool
4	Direct access pool
5	Tape library
6	Disk library
7	Unit record pool
8	Teleprocessing control
9	System security
10	System error/maintenance/system programmer information
11	Programmer information
12	Emulators
13-20	Reserved for customer use
21-28	Reserved for subsystem use
29	Disaster Recovery
30-40	Reserved for IBM
41	Information about JES3 job status
42	General information about JES2 or JES3
43-64	Reserved for JES2 or JES3
65-96	Messages associated with particular processors
97-128	Messages associated with particular devices

One way to limit the messages that arrive at a console is to assign a routing code or codes to a console. The console then receives only the messages that are appropriate.

To learn what the routing codes for a console are, enter the DISPLAY CONSOLES command. Figure 3-3 on page 3-23 shows the display that appears in response to this command.

You define routing codes for a console by:

- **Using the VARY Command:**

Use operands on the VARY command to add to the existing set (AROUT operand), subtract from the existing set (DROUT), or redefine the set (ROUT).

Example

To assign the routing codes 1, 2, 9, and 10 for a console named CON81D, enter:
VARY CN(CON81D),CONSOLE,ROUT=(1,2,9,10)

The effect of this command lasts only for the duration of the IPL.

Defining Message Levels for a Console

Assigning routing codes is one way to limit message traffic to a console. You can further reduce the number of messages that appear on a console by directing certain messages to consoles by message levels. The system differentiates among these kinds of message levels:

- Write-to-operator with reply (WTOR) messages, which demand an immediate reply.
- System failure and immediate action messages (descriptor codes 1 and 2), which indicate that a task is awaiting your action.
- Critical eventual action messages (descriptor code 11), which indicate a potential system problem.
- Eventual action messages (descriptor code 3), which do not require immediate attention.
- Broadcast messages, which are normally sent to every active console regardless of the routing code you assigned to the console.
- Informational messages, which generally indicate system status. (Most messages are informational.)

Assignment by message level means that a console can accept combinations of action, broadcast, and informational messages that the system sends to a console. You can choose among the following message level options:

- R** Write to operator (WTOR) messages are to appear
I Immediate action messages (descriptor codes 1 and 2) are to appear
CE Critical eventual action messages (descriptor code 11) are to appear
E Eventual action messages (descriptor code 3) are to appear
IN Informational messages are to appear
NB Broadcast messages are **not** to appear
ALL All messages, including broadcast messages, are to appear.

If the LEVEL parameter in the CONSOLxx member is not coded, the system sends all messages, including broadcast messages, to the console.

To display the routing codes and message levels for a console, issue the DISPLAY CONSOLES command. Figure 3-3 on page 3-23 shows the display that appears in response to this command.

To display the routing codes and message levels that appear only on the system log **and not on any console**, issue the DISPLAY CONSOLES,HONLY command.

You define the level of messages for a console by:

- **Using the CONTROL Command:**

Use the LEVEL operand on the CONTROL V command to assign message levels to a console.

Example 1

To direct only WTOR messages and immediate action messages to the console with console name CON06, enter:

CONTROL V,LEVEL(R,I),L=CON06

When you change message levels so that some informational or broadcast messages will not appear at any console, the system rejects the CONTROL V command. If you want to override this rejection, use the UNCOND operand.

These messages appear only in the hardcopy log, and on any extended MCS consoles that are receiving the hardcopy message set. The system displays this message to warn you of the message loss:

Example 2

To assign to the console with console name CON12 (and device number 81D) the informational messages directed to the tape libraries (routing code 5) and disk libraries (routing code 6), enter:

```
VARY 81D,CONSOLE,ROUT=(5,6)  
CONTROL V,LEVEL=IN,L=CON12
```

Controlling the Format of Messages

On a display console, a message can appear by itself or with information about the message, such as job and system identification and the time the message was issued. "Messages Sent to Display Consoles" on page 3-6 describes the format of messages and describes the optional information that the system can include with each message:

- J** The jobname/job id of its issuer
- S** The name of the system that issued the message
- T** A time stamp
- M** Only the message text displays
- X** Suppress system and job name of its issuer when S and/or J are specified

You request that additional information precede each message the system sends a console by:

- **Using the CONTROL Command:**

Use the MFORM operand on the CONTROL S command to change the format of messages.

Example

To request that the system add to all messages that appear at console CON2 a time stamp, the name of the system that issued the message, and the jobname or ID of its issuer, enter:

```
CONTROL S,MFORM=(J,T,S),L=CON2
```

The effect of this command lasts only for the duration of the IPL.

Controlling the Message Processing Facility (MPF)

The message processing facility (MPF) controls message processing. It controls the suppression and retention of messages, the installation exits that gain control when certain messages are issued, and message presentation (that is, the color, intensity and highlighting of messages) at certain consoles.

The operator can:

- See what MPF member or members are active with the DISPLAY command
- Change the active MPF member or members with the SET command.

For MPF to suppress messages, hardcopy processing must be active. The suppressed messages do not appear on any console; they do appear on the hardcopy log, and any extended MCS consoles that are receiving the hardcopy message set.

Message Presentation

Message presentation refers to the way the system uses color, intensity, and highlighting (including blinking, reverse-video, and underscoring) to identify messages that require action. The presentation depends on the type of device you are using.

Using the SET Command:

Enter the SET MPF command to change the MPFLSTxx member or members that the system is to use.

Example

To specify MPFLST03 and MPFLST06 as the MPF members for the system to use, enter:

```
SET MPF=(03,06)
```

The effect of this command lasts only for the duration of the IPL.

Displaying Information About Messages Awaiting Action: Many systems now handle so much work so quickly that you cannot always keep up with the messages that demand operator response. These messages roll off the screen before you can respond. The action message retention facility keeps these messages, including the WTORs and JES3 messages, so that you can see them at a later time. (While you are examining the messages that you missed, you might, of course, miss more messages. Experience with your system will help you determine how frequently you need to check for retained action messages.)

The DISPLAY R command allows you to display all outstanding action messages or a subset of these messages. For example, to display all outstanding action messages at your console, enter DISPLAY R,M. To display all the outstanding critical eventual-action messages (descriptor code 11), enter DISPLAY R,CE. See *z/OS MVS Planning: Operations* for use of the DISPLAY R command.

Controlling the Action Message Retention Facility

During its initialization, the system can start the action message retention facility (AMRF). When active, the facility retains in a buffer area all action messages (those messages with descriptor codes 1, 2, 3, and 11) except those specified by the installation in the active MPFLSTxx member.

If the first system IPLs and AMRF is active, then AMRF is active on every system that you subsequently IPL into the sysplex.

When you have performed the action required by a message displayed on the screen, the system deletes the message; or you can use the CONTROL C command to delete the message. You can remove action messages from the screen that require later action, then retrieve them in their entirety later by using the DISPLAY R command. Periodically, you should display the retained messages and delete the ones for which action has been taken so that the action message retention buffer does not fill up.

To change the messages that the action message retention facility is to retain, activate an MPFLSTxx member that contains the message retention options you want. The system default is to have the action message retention facility on.

To learn the status of the action message retention facility, issue the CONTROL M,REF command.

You change the status of the action message retention facility by:

- **Using the CONTROL Command**

Use the CONTROL M,AMRF command to turn the action message retention facility on or off.

Example

To deactivate the action message retention facility, enter:

```
CONTROL M,AMRF=N
```

Activating WTO and WTOR Installation Exit Routines

The system programmer at your installation codes installation exit routines that gain control when the system issues certain messages. A WTO installation exit can change routing codes, descriptor codes, and message texts, as well as perform other message processing; it can override MPF processing. Information about coding these installation exits appears in *z/OS MVS Installation Exits*.

The most effective message control involves coding and installing the installation exit IEAVMXIT, which can gain control when any WTO or WTOR message is issued.

To learn whether IEAVMXIT is active or not, issue the CONTROL M,REF command. The system displays (in the entry area) the status of the action message retention facility, the status of installation exit IEAVMXIT, and the limit of the number of WTO and WTOR buffers.

Your installation might have other exit routines to process messages. MPFLSTxx parmlib members contain the IDs of messages and the installation exits that process these messages. To activate processing by these installation exits, see “Controlling the Message Processing Facility (MPF)” on page 3-32.

You can activate the installation exit IEAVMXIT, if it is installed, by:

- Using the CONTROL Command:

Use the UEXIT operand on the CONTROL command to control whether the installation exit IEAVMXIT is active.

Example

To deactivate IEAVMXIT, enter:

```
CONTROL M,UEXIT=N
```

The effect of the command lasts only for the duration of the IPL.

Checking Message Processing, Retention, and Presentation Options

Issue the DISPLAY MPF,MSG command to see:

- Which messages are being suppressed by MPF
- Which action messages are not being retained by the action message retention facility
- Which installation exits receive control for selected messages
- The status of the general WTO installation exit IEAVMXIT
- Whether this message is automated by MPF
- The MPFLSTxx member that identifies the message ID, color attribute, or command installation exit definition
- A list of the subsystems receiving foreign messages and DOMs

Issue the DISPLAY MPF,COLOR command to see:

- What color, intensity, and highlighting capabilities are in effect

Issue the DISPLAY MPF command to see all of this information for the messages that are defined in the MPFLSTxx parmlib member.

Defining Program Function Keys (PFKs)

You can define program function keys for a console by activating a PFK table or by using the CONTROL N,PFK= command.

Defining PFKs Using PFK Tables

You define a console's PFKs by activating a PFK table — a table that your installation has defined. The PFK table resides, optionally with other PFK tables, in a PFKTABxx parmlib member. The entries in this table:

- Assign one or more commands to a PFK
The text of one or more commands are to be associated with a PFK. Later, when you press this PFK, the commands are entered into the system.
- Assign one or more other PFKs to a PFK
The commands associated with other PFKs are to be associated with one PFK.

Entries in the PFK table also determine whether conversational or nonconversational mode is to be in effect for a command defined to a PFK. In nonconversational mode, the commands associated with a key are entered immediately when you press the key. In conversational mode, pressing a PFK causes the command to appear in the entry area, but no enter action takes place. You can change, enter, or cancel the command according to your requirements.

In conversational mode, the cursor normally appears under the third non-blank character when the command is in the entry area. If you want the cursor to appear in a different location, when you define the command, type an underscore before the character under which the cursor is to appear. The system deletes the space occupied by the underscore in the actual command. For example, if you add the following entry to a PFK table:

```
PFK(5) CMD('D U,L=_XXX') CON(Y)
```

pressing PFK 5 causes the following to appear in the entry area:

D U,L=XXX

If you want an underscore to appear in the command, code two consecutive underscores. The system will treat them as a single underscore, and will not use them for cursor placement. Example:

If the PRKTAB table contains:

```
PFK(17) CMD('E _XXXXXXXX,SRVCLASS=BAT__HI'),CON(Y)
```

when you press PFK17, the entry area will contain:

E _XXXXXXXX,SRVCLASS=BAT_HI

with the cursor under the first X.

Selector pens also use the definitions in PFK tables.

You can use some MVS commands to display information about the PFKs at your console, or to change the PFKs that are available for your consoles. The following commands relate to the previous example:

- Display the PFK definitions in the PFK table named MVSCMDS.

```
DISPLAY PFK,TABLE=MVSCMDS
```

- List the names of all PFK tables in the active PFKTABxx member.

```
DISPLAY PFK,TABLE
```

- Assign the commands in the PFK table named JES2CMDS to the PFKs on your console.

```
CONTROL N,PFK=JES2CMDS
```

- Activate another PFKTABxx member, in this case PFKTAB02.

```
SET PFK=02
```

This command assumes that you have a PFK table in PFKTAB02 and that you want to replace MVSCMDS with another PFK table. (Other consoles might be using tables in the former PFKTABxx member. PFK definitions for these consoles are not affected by the action of this SET command.)

Defining PFKs Using the CONTROL Command

Use the CONTROL N,PFK= command to change the definition for PFKs. This command performs three tasks:

- Assigns one or more commands to a PFK
- Assigns one or more other PFKs to a PFK
- Assigns a PFK table to your console.

With the CONTROL N,PFK= command you can also determine whether conversational or nonconversational mode is to be in effect for the commands defined to the PFK. Nonconversational mode is the default. For example, if you define PFK 5 as follows:

```
CONTROL N,PFK=(5,CMD='D U,L=CON9A'),CON=N
```

pressing PFK 5 has the same effect as typing DISPLAY U,L=CON9A and pressing the ENTER key.

On the other hand, if you specify conversational mode by entering:

```
CONTROL N,PFK=(5,CMD='D U,L=CON9A'),CON=Y
```

pressing PFK 5 causes the command D U,L=CON9A to appear in the entry area but no enter action takes place. You can change, enter, or cancel the command according to your requirements.

The system does not accept PFK assignments that may result in an endless loop. Examples of commands that the system will not accept are:

- You **cannot** assign a PFK to itself. For example, the system does not accept CONTROL N,PFK=(9,KEY=9).
- If a PFK is being assigned a list of PFKs (that is, a key list), that PFK **cannot** appear in the key list for another PFK. For example, if PFK 5 is already associated with keys 3 and 4, the system does not accept CONTROL N,PFK=(6,KEY=5,8).
- If a PFK is already in a key list, you **cannot** assign a key list to that PFK. For example, if key 4 is associated with keys 5 and 6, the system does not accept CONTROL N,PFK=(5,KEY=7,8).

Remember that the assignment of the command to the PFK through the CONTROL command lasts only for the duration of the IPL.

Example 1

If PFK 3 is associated with commands SET OPT=PM and SEND 14,BRDCST, and PFK 4 is associated with the command START GTF,MODE=INT,BUF=387,TIME=YES,DEBUG=YES, you can associate all three of these commands with PFK 5 by entering:

```
CONTROL N,PFK=(5,KEY=3,4),CON=Y
```

The commands associated with PFK 5 are now:

```
SET OPT=PM  
SEND 14,BRDCST  
START GTF,MODE=INT,BUF=387,TIME=YES,DEBUG=YES
```

The system schedules the commands in that order, but might not execute them in that order.

Example 2

To remove a definition previously set for PFK 5, leaving PFK 5 undefined, enter:

```
CONTROL N,PFK=(5,CMD='')
```

The PFKTABxx and PFKs

The PFKTABxx parmlib members contain the **PFK tables** that have the definitions your installation has assigned to PFKs. To associate your console's PFKs with the definitions in a particular PFK table:

- The PFK parameter on the INIT statement in the active CONSOLxx member must identify the PFKTABxx member that contains the table.
- The PFKTAB parameter on the CONSOLE statement in CONSOLxx must identify the name of the PFK table.
- The particular table must contain entries; each entry supplies a command or commands associated with a PFK.

You use CONSOLxx and PFKTABxx members to set the PFK definitions at IPL. You can also change the PFK definitions for the duration of the IPL:

To change a PFK table:

1. Enter SET PFK=xx, if necessary, to change the PFKTABxx member in effect for the console. Other consoles using the former PFKTABxx member are not affected by the SET command you issue for your console.
2. Enter CONTROL N,PFK=nnnnnnnn to assign the PFK table that contains the PFK definitions you want to use for the console.

To change a PFK key:

- Enter CONTROL N,PFK=(nn1,CMD='...') to change a specific PFK key definition for the console where the command is entered.

During IPL, the system looks for the PFK parameter in CONSOLxx member. If the system does not find the PFK parameter, it issues the message:

```
IEA180I USING IBM DEFAULT DEFINITIONS. NO PFK TABLES REQUESTED
```

In this case, PFKs 1 through 8 have the defaults that IBM supplies. These defaults are shipped in sample IEESPFK.

To define PFKs for your consoles, see "Defining PFKs Using PFK Tables" on page 3-35.

Processing Hardcopy

Logging provides a permanent record of system activity. Your installation can record system messages and, optionally, commands and command responses, by using the system log (SYSLOG), the operations log (OPERLOG), or both. Your installation can also allow an extended MCS console to receive the same set of messages as SYSLOG and OPERLOG. The log that receives messages is called the hardcopy medium. The group of messages that is recorded is called the hardcopy message set.

The hardcopy message set is defined at system initialization and persists for the life of the system. See *z/OS MVS Planning: Operations* for the characteristics of the hardcopy message set.

The Hardcopy Message Set

Unless you specify otherwise, the hardcopy message set includes all messages, except those that are explicitly omitted through the WTO macro or installation exits. You can request that the hardcopy message set not include messages with certain routing codes. The minimum set of routing codes is 1, 2, 3, 4, 7, 8, 10, and 42. If you attempt to eliminate any of these, the system includes messages with these routing codes in the hardcopy message set anyway.

To see information about the kinds of messages that the system includes in the hardcopy message set, but does not send to any console, issue the DISPLAY CONSOLES,HONLY command.

Selecting Messages for the Hardcopy Message Set: You control which messages are included in the hardcopy message set by:

- **Using the VARY Command:**

Use the VARY,HARDCPY command to specify the routing codes of messages that are included in the hardcopy message set. You can add to the existing set (AROUT operand), subtract from the existing set (DROUT), or redefine the set (ROUT).

Example

To stop including all routing codes except the minimum set, enter:

VARY ,HARDCPY,DROUT=(5,6,9,11-41,43-128)

The system would give the same response if you entered the VARY,HARDCOPY,ROUT=NONE command.

The effect of this command lasts only for the duration of the IPL.

Selecting Commands and Command Responses for the Hardcopy Message Set: Unless you specify otherwise, the system includes all operator and system commands, responses, and status displays (including static and time-interval updated status displays) in the hardcopy message set. To request that some commands and command responses not be included in the hardcopy message set, the system gives you the following choices on the VARY ,HARDCPY command:

NOCMDS	The system does not include operator commands or their responses in the hardcopy message set.
INCMDS	The system includes all operator commands and their responses, excluding any status displays, in the hardcopy message set.
STCMDS	The system includes all operator commands and their responses in the hardcopy message set.

CMDS	The system includes all operator commands and their responses, including all status displays, in the hardcopy message set.
-------------	--

To see which commands and command responses the system includes in the hardcopy message set, issue the DISPLAY CONSOLES command. Figure 3-3 on page 3-23 shows the display that appears in response to this command.

Use of the TRACK command in z/OS releases

Because dynamic displays dealt exclusively with the TRACK command that has been removed from z/OS V1R7 and later, dynamic displays are no longer issued by z/OS. Multiline messages with descriptor codes 8, 9, 10 appear in the log with a hardcopy message of CMDS.

Note:

You control which commands and command responses are included in the hardcopy message set by:

- **Using the VARY Command:**

Use the VARY ,HARDCPY command to change the commands or the command responses that are included in the hardcopy message set.

Example

To request that the hardcopy message set include all operator commands and responses except status displays, enter:

VARY ,HARDCPY,INCMDS

The effect of this command lasts only for the duration of the IPL.

The Hardcopy Medium

You can specify whether the hardcopy medium is the system log (SYSLOG) or the operations log (OPERLOG). If you use SYSLOG as the hardcopy medium, start a writer that includes the system log message class (A for MVS, unless otherwise specified in your installation). The SYSLOG spool file is managed by JES and cannot be browsed or edited using normal means. The external writer will write it to an SMF-managed file.

The System Log: The system log (SYSLOG) is a direct access data set that stores messages and commands. It resides in the primary job entry subsystem's spool space. It can be used by application and system programmers (through the WTL macro) to record communications about programs and system functions. You can use the LOG command to add an entry to the system log.

Several kinds of information can appear in the system log:

- Job time, step time, and data from the JOB and EXEC statements of completed jobs entered by user-written routines
- Operating data entered by programs using a write to log (WTL) macro instruction
- Descriptions of unusual events that you enter using the LOG command
- The hardcopy message set

When MVS has JES3 as its job entry subsystem, the system log can record console activity. If used to record console activity, the system log is referred to in JES3 messages as DLOG.

In CONSLxx, you can use the HCFORMAT keyword on the HARDCOPY statement to specify whether hardcopy records should have a 2-digit or 4-digit year.

The system log is queued for printing when the number of messages recorded reaches a threshold specified at system initialization. You can force the system log data set to be queued for printing before the threshold is reached by issuing the WRITELOG command.

If the system log is defined as the only hardcopy medium and SYSLOG fails, hardcopy is suspended and the system issues message CNZ4201E. To avoid the loss of hardcopy, IBM recommends both SYSLOG and OPERLOG be defined as hardcopy.

The Operations Log: The operations log (OPERLOG) is an MVS system logger application that records and merges the hardcopy message set from each system in a sysplex that activates OPERLOG. Use OPERLOG as your hardcopy medium when you need a permanent log about operating conditions and maintenance for all systems in a sysplex.

For more information on OPERLOG, see *z/OS MVS Setting Up a Sysplex*.

Assigning the Hardcopy Medium: Assign the hardcopy medium by using the VARY command.

Use the HARDCPY operand on the VARY command to assign SYSLOG or OPERLOG as the hardcopy medium. You can assign both SYSLOG and OPERLOG as the hardcopy medium by issuing the command separately.

Example:

To specify the hardcopy medium as SYSLOG, issue:

```
VARY SYSLOG,HARDCPY
```

The effect of this command lasts only for the duration of the IPL.

To display information about the hardcopy medium, enter:

```
DISPLAY CONSOLES,HARDCOPY
```

The resulting display tells you the following information:

- Whether the hardcopy medium is SYSLOG, OPERLOG, or both
- The criteria that have been defined by the installation for selecting messages for the hardcopy message set
- The number of messages waiting to be placed on the hardcopy medium

Chapter 4. MVS System Commands Reference

This chapter describes the functions, syntax, and parameters of all the MVS base control program (BCP) system commands. You can use these commands to control both the system itself and multiple console support (MCS) or SNA multiple console support (SMCS) consoles.

Table 4-1 on page 4-2 sums up the MVS BCP system commands and their functions. The figure shows the operator command groups for each command and tells whether you can enter the command from the job stream, an MCS or SMCS console, or an extended MCS console session. An extended MCS console session is established either by the TSO/E CONSOLE command as an interactive TSO/E session or by a program issuing the MCSOPER macro so the program can receive messages and issue commands. See *z/OS TSO/E System Programming Command Reference* for information about the TSO/E CONSOLE command. See *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU* for information about the MCSOPER macro.

An installation can use RACF to control which consoles and commands operators can use. For more information, see *z/OS MVS Planning: Operations*.

Operator commands may contain the following characters:

- A to Z
- 0 to 9
- '# \$ & () * + , - . / ¢ < ! ! ; ¬ % _ > ? : @ " =

The system translates characters that are not valid into null characters (X'00').

You can enter operator commands in uppercase or lowercase. Unless enclosed in apostrophes, lowercase letters are converted to uppercase. Therefore, when a lowercase response is required, you must enclose the text in apostrophes. Also, when an apostrophe appears in the text of a command and the text is enclosed in apostrophes, you must enter two apostrophes in the text. For example, you would enter:

```
SEND 'Your job''s done'
```

You can enter system commands through a multiple console support (MCS) console, an SNA multiple console support (SMCS) console, an extended MCS (EMCS) console, or through the input stream (card reader). Table 4-1 on page 4-2 indicates from which types of consoles a command is accepted. Superscripts denote footnotes that can be found on the last page of the table. All examples in this book show the format for MCS and SMCS console entry.

Notes:

1. If you enter a system command through a card reader in a JES2 system, enter \$VS,'system command' when you enter the command between jobs, and enter //b system command when you enter the command within a job.
2. Do not use the JES backspace character within a system command.

Following the summary figure is a section on command syntax and format. The syntax rules are shown in "How to read syntax conventions" on page 4-12.

The rest of this chapter consists of a description of each command in more detail. The descriptions are in alphabetical order by command name. Each description lists the functions that the command performs followed by the command's syntax and

parameters. The syntax and parameters of complex commands follow subsets of the listed functions. Descriptions of the parameters and keywords appear in the order in which they appear in the syntax.

Table 4-1. System Command Summary

Command (Abbr)	Function	Acceptable From	Command Group
ACTIVATE	Build the interface to and invoke the hardware configuration definition (HCD) application program interface.	MCS, SMCS or extended MCS console ⁴	SYS
CANCEL (C)	Cancel a MOUNT command Cancel a time-sharing user Cancel a cataloged procedure Cancel a job in execution Cancel a started catalog procedure Cancel an external writer allocation Cancel the writing of a SYSOUT data set by an external writer session Cancel a running APPC/MVS transaction program Cancel a z/OS UNIX System Services process	MCS, SMCS or extended MCS consoles or job stream ⁴	SYS
CHNGDUMP (CD)	Override dump options specified in parmlib, on the ABEND, CALLRTM, and SETRP macros, and in the SDUMP parameter list	MCS, SMCS or extended MCS consoles or job stream ⁴	SYS
CMDS	DISPLAY or SHOW information about commands that are executing or waiting for execution ABEND or REMOVE executing commands or commands waiting for execution	MCS, SMCS or extended MCS consoles or job stream ⁴	INFO MASTER
CONFIG (CF)	Place processors online or offline Place central storage elements online or offline Place amounts of central storage online or offline Place ranges of central storage online or offline Place expanded storage elements online or offline Place channel paths online or offline Place Vector Facilities online or offline MCS, SMCS or extended MCS console ⁴	MCS, SMCS or extended MCS consoles	MASTER

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
CONTROL (K)	Change display area specifications Delete certain messages Halt printing of a status display Control area displays Remove information from the screens Activate, deactivate, or display the status of the action message retention facility Change or display the number of allowed message and reply buffers Change or display message deletion or format specifications Change or display the status of WTO user exit IEAVMXIT Define commands for PFKs Purge message queue of a console. Change operating mode of console Select the message levels for a console Increase the RMAX value In a sysplex, change the maximum time MVS waits before aggregating messages from routed commands	MCS and SMCS consoles	INFO INFO INFO INFO INFO MASTER MASTER INFO MASTER INFO or MASTER INFO or MASTER INFO INFO INFO MASTER
DEVSERV (DS)	Display current status of devices and corresponding channel paths	MCS, SMCS or extended MCS consoles or job stream ⁴	INFO

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
DISPLAY (D)	Display APPC/MVS configuration information Display ASCH configuration information Display IOS configuration Display console configuration information Display z/OS UNIX System Services information Display MVS message service and current available languages Display status of external time reference (ETR) ports Display status information for trace Display system requests and status of the AMRF Display CONTROL command functions Display configuration information Display device allocation Display current system status Display system information requests Display local and Greenwich mean time and date Display status or contents of SYS1.DUMP data sets and captured data sets Display dump options in effect Display SMF options in effect or SMF data sets Display information about the cross system coupling facility information (XCF) Display information about operation information (OPDATA) in a sysplex, or display the status of the console ID tracking facility, or display the settings made by the SETCON MONITOR command Display information about the SMS configuration or the status of SMS volumes or storage groups or SMS trace options Display information about all subsystems defined to MVS.	MCS, SMCS Console or extended MCS consoles or job stream ⁴	INFO

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
DISPLAY (D) (continued)	Display page data set information Display current MIH time intervals for individual devices, or for device classes Display SLIP trap information Display commands defined for PFKs Display the messages MPF is processing and color, intensity, and highlighting display options in effect Display entries in the list of APF-authorized program libraries Display dynamic exits Display information about the LNKLST set Display information about modules dynamically added to the LPA. Display state of the systems, a particular system's CTCs, the status of an RNL change, or the contents of RNLs in the global resource serialization complex Display the status of the active workload management service policy for systems or application environments Display information about registered products and the product enablement policy. Display information about system logger and log stream resources.	MCS, SMCS or extended MCS consoles or job stream ⁴	INFO
DUMP	Request a dump of virtual storage to be stored in a SYS1.DUMP data set	MCS, SMCS or extended MCS console ⁴	MASTER
DUMPDS (DD)	Change the system's list of SYS1.DUMP data sets Clear full SYS1.DUMP data sets and make them available for dumps	MCS, SMCS or extended MCS console ⁴	SYS
FORCE	Force termination of: <ul style="list-style-type: none"> • A MOUNT command • A job in execution • An external writer allocation • The writing of a SYSOUT data set by an external writer • A non-cancellable job, time-sharing user, or started task • A running APPC/MVS transaction program 	MCS, SMCS or extended MCS console ⁴	MASTER
HALT (Z)	Record statistics before stopping the system (Must first stop subsystem processing with a subsystem command)	MCS, SMCS or extended MCS console ⁴	SYS
IOACTION (IO)	Stop or resume I/O activity to DASD	MCS, SMCS or extended MCS console ⁴	MASTER

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
LIBRARY (LI)	Eject a volume from a library of removable storage media. Reactivate processing for certain installation exits without stopping or restarting the object access method (OAM). Set or display the media type of scratch volumes that the system places into the cartridge loader of a device within a tape library. Display tape drive status.	MCS and SMCS consoles	SYS
LOG (L)	Enter comments in the system log	MCS, SMCS or extended MCS consoles or job stream ⁴	INFO
LOGOFF	To log off MCS and SMCS consoles	MCS and SMCS consoles	INFO
LOGON	To access the MCS and SMCS consoles	MCS and SMCS console	INFO
MODE	Control recording of or suppress system recovery and degradation machine check interruptions on the logrec data set Control the monitoring of hard machine check interruptions	MCS, SMCS or extended MCS console ⁴	SYS
MODIFY (F)	Change characteristics of a job by modifying the job parameters Specify criteria an external writer uses to select data sets for processing Cause an external writer to pause for operator intervention Build a new LLA directory Display information about the catalog address space or request the catalog address space to perform a specified service. Modify TSO/VTAM time-sharing Rebuild a new LNKLST directory Display the status of the DLF, or change DLF parameters or processing mode	MCS, SMCS or extended MCS consoles or job stream ⁴	SYS
MONITOR (MN)	Continuously display data set status Continuously display job status Monitor time-sharing users logging on and off the system	MCS, SMCS or extended MCS consoles or job stream ⁴	INFO
MOUNT (M)	Mount volumes	MCS, SMCS or extended MCS consoles or job stream ⁴	I/O
PAGEADD (PA)	Add local page data sets Specify data sets as non-VIO page data sets	MCS, SMCS or extended MCS console ⁴	SYS

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
PAGEDEL (PD)	Delete, replace, or drain a local page data set (PLPA, common page data sets, and the last local page data set cannot be deleted, replaced or drained)	MCS, SMCS or extended MCS console ⁴	SYS
QUIESCE	Put system in MANUAL state without affecting step timing	MCS, SMCS or extended MCS console ⁴	MASTER
REPLY (R)	Reply to system information requests Reply to system requests during recovery processing Specify component trace options after issuing TRACE CT Specify system parameters Set the time-of-day clock and specify the installation performance specification Specify SMF options Specify DUMP options	MCS, SMCS or extended MCS consoles or job stream ⁴	INFO
RESET (E)	Change performance group of a job currently in execution Assign work to a new workload management service class. Also, quiesce and resume executing work. Force a hung console device offline.	MCS, SMCS or extended MCS consoles or job stream ⁴	SYS MASTER
ROUTE (RO)	Direct a command to another system, to all systems, or to a subset of systems in the sysplex	MCS, SMCS or extended MCS consoles or job stream ⁴	INFO
SEND (SE)	Communicate with other operators Communicate with specific time-sharing users Communicate with all time-sharing users Save messages in the broadcast data set for issuance at TSO LOGON time or when requested List messages accumulated in the notices section of the broadcast data set Delete a message from the notices section of the broadcast data set	MCS, SMCS or extended MCS consoles or job stream ⁴	INFO

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
SET (T)	<p>Add modules to, or delete modules from, the LPA dynamically. Change:</p> <ul style="list-style-type: none"> • the local time and date • the system resources manager (SRM) parameters • the MPF parameters • the dump analysis and elimination (DAE) parameters • SLIP processing by changing the active IEASLPxx parmlib member • SMS parameters by selecting member IGDSMSxx in , start SMS if not started at IPL, or restart SMS if it cannot be restarted automatically • available PFK tables • MIH time intervals by changing the active IECSIOSxx parmlib member • excessive spin-loop timeout interval recovery actions • RNLs by selecting new GRSRNLxx parmlib members • the APPC/MVS address space information • the APPC/MVS transaction scheduler information • the PPT information • the active console group definitions in the sysplex • the MMS parameters • the command installation exits the system is to use • the product enablement policy the system is to use • the exclusion list that the Console Id Tracking facility will use <p>Restart SMF or change SMF parameters by changing the active SMFPRMxx parmlib member</p> <p>Start or stop the common storage and tracking functions</p> <p>Start, refresh, or stop MMS. Update:</p> <ul style="list-style-type: none"> • the APF list and dynamic exits • the format or contents of the APF list • the LNKLKST set for LNKLST concatenation 	MCS, SMCS or extended MCS consoles or job stream ⁴	SYS
SETCON	Activate console environment functions	MCS, SMCS or extended MCS console ⁴	MASTER
SETETR	Enable external time reference (ETR) ports that have been disabled	MCS, SMCS or extended MCS console ⁴	SYS

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
SETGRS	<ul style="list-style-type: none"> • Migrate a currently active global resource serialization ring complex to a global resource serialization star complex • Modify the current RESMIL or TOLINT values • Set the system values for <ul style="list-style-type: none"> – GRSQ – SYNCRES – ENQMAXA – ENQMAXU • Change the contention notifying system (CNS) in a global resource serialization complex 	MCS, SMCS or extended MCS console ⁴	MASTER
SETIOS	Respecify, add, or delete MIH time intervals, update DCM, enable/disable FICON statistics, and enable/disable the MIDAW facility, all without changing the active IECIOSxx parmlib member	MCS, SMCS or extended MCS console ⁴	SYS
SETLOAD	Switch dynamically from one parmlib concatenation to another without having to initiate an IPL	MCS, SMCS or extended MCS console ⁴	SYS
SETLOGR	Take action on system logger log stream resources.	MVS, SMCS or extended MVS console ⁴	MASTER
SETLOGRC	Change the logrec recording medium	MCS, SMCS or extended MCS console ⁴	MASTER
SETOMVS	Change the options that z/OS UNIX System Services uses.	MCS, SMCS or extended MCS console ⁴	SYS
SETPROG	Update APF list Update dynamic exits Update the LNKLST set Dynamically add modules to, or delete modules from, the LPA.	MCS, SMCS or extended MCS console ⁴	SYS
SETRRS CANCEL	End RRS processing	MCS, SMCS or extended MCS console ⁴	SYS
SETSMF (SS)	Change SMF parameters without changing the active SMFPRMxx parmlib member	MCS, SMCS or extended MCS consoles or job stream ⁴	SYS
SETSMS	Change SMS parameters without changing the active IGDSMSxx parmlib member	MCS, SMCS or extended MCS console ⁴	SYS
SETSSI	Dynamically add, activate or deactivate a subsystem.	MCS, SMCS or extended MCS console ⁴	MASTER
SETUNI	Control the Unicode environment.	MCS, SMCS or extended MCS console. ⁴	SYS
SETXCF	Control the cross-system coupling facility (XCF)	MCS, SMCS or extended MCS console ⁴	MASTER

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
SLIP (SL)	Set SLIP traps Modify SLIP traps Delete SLIP traps	MCS, SMCS or extended MCS consoles or job stream ⁴	SYS
START (S)	Start a job from a console Start the advanced program-to-program communication (APPC/MVS) address space Start the APPC/MVS scheduler (ASCH) address space Start the data facility storage management subsystem (DFSMS/MVS) license compliance facility Start the generalized trace facility (GTF) Start the library lookaside (LLA) address space Start the object access method (OAM) Start resource recovery services (RRS) Start the system object model (SOM) Start TSO/VTAM time-sharing Start the virtual lookaside facility (VLF) or the data lookaside facility (DLF) Start an external writer	MCS, SMCS or extended MCS consoles or job stream ⁴	SYS
STOP (P)	Stop a job in execution Stop an address space Stop an ASCH initiator Stop an initiator Stop the data lookaside facility (DLF) Stop the generalized trace facility (GTF) Stop the library lookaside (LLA) address space Stop the object access method (OAM) Stop the system object model (SOM) Stop TSO/VTAM time-sharing Stop the virtual lookaside facility (VLF) Stop an external writer	MCS, SMCS or extended MCS consoles or job stream ⁴	SYS
STOPMN (PM)	Stop continual display of data set status Stop continual display of job status Stop monitoring the activity of time-sharing users.	MCS, SMCS or extended MCS consoles or job stream ⁴	INFO
SWAP (G)	Move a volume from one device to another	MCS, SMCS or extended MCS consoles ⁴	I/O

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
SWITCH (I)	Manually switch recording of SMF data from one data set to another	MCS, SMCS or extended MCS console ⁴	SYS
TRACE	Start, stop, or modify system trace Start, stop, or modify master trace Start, stop, or modify component trace Display the status of system trace, master trace, or component trace	MCS, SMCS or extended MCS console ⁴	SYS MASTER MASTER SYS
UNLOAD (U)	Remove a volume from system use	MCS, SMCS or extended MCS consoles or job stream ⁴	I/O
VARY (V)	Control the hardcopy message set and the hardcopy medium. Change the status of a console Change the SMS status of a storage group or volume for one or more MVS systems in the SMS complex Place I/O devices online or offline Assign and control consoles Place I/O paths online or offline Remove a system from a sysplex Place I/O paths online after C.U.I.R service Change a system's participation in a global resource serialization complex Change routing codes for a console Activate a workload management service policy for a sysplex Control an application environment	MCS, SMCS, or extended MCS consoles or job stream	MASTER, I/O, or CONS ³
WRITELOG (W)	Schedule printing of system log Change system log output class Close the system log and discontinue the log function Restart system log after closing	MCS, SMCS or extended MCS consoles or job stream ⁴	SYS
Notes:			
<ol style="list-style-type: none"> 1. CONS command group when message routing is specified. 2. For information about VTAM commands, see <i>z/OS Communications Server: SNA Operation</i>. 3. This command is in a different command authority group depending on the parameters specified on the command. See Table 3-5 on page 3-24 for more information. 4. An extended MCS console can be either an interactive TSO/E session or a program that issues the MCSOPER macro. 			

Command Syntax Notation

You must follow certain syntactical rules when you code the MVS commands described in this chapter. Use “How to read syntax conventions” to help you with the syntax.

How to read syntax conventions

This section describes how to read syntax conventions. It defines syntax notations and provides syntax examples that contain these items.

Table 4-2. Syntax conventions

Notation	Meaning	Example	
		Book syntax	Sample entry
Apostrophes	Apostrophes indicate a parameter string and must be entered as shown.	SEND 'message',NOW	SEND 'listings ready',NOW
Comma	Commas must be entered as shown.	DISPLAY C,K	DISPLAY C,K
Ellipsis ...	Ellipsis indicates that the preceding item or group of items can be repeated one or more times. Do not enter the ellipsis.	VARY (devspec[,devspec]...),ONLINE	VARY (282,283,287),ONLINE
Parentheses and special characters	Parentheses and special characters must be entered as shown.	DUMP COMM=(text)	DUMP COMM=(PAYROLL)
Underline	Underline indicates a default option. If you select an underlined alternative, you do not have to specify it when you enter the command.	K M[,AMRF={Y N}] , <u>REF</u>	K M
Lowercase parameter	Lowercase indicates a variable term. Substitute your own value for the item.	MOUNT devnum	MOUNT A30 or mount a30
Uppercase parameter	Uppercase indicates the item must be entered using the characters shown. Enter the item in either upper or lowercase.	DISPLAY SMF	DISPLAY SMF or display smf
Single brackets	Single brackets represent single or group-related items that are optional. Enter one or none of these items.	DISPLAY DMN[=domainum]	DISPLAY DMN=5
Stacked brackets	Stacked brackets represent group-related items that are optional. Enter one or none of these items.	[TERMINAL] [NOTERMINAL]	NOTERMINAL

Table 4-2. Syntax conventions (continued)

Notation	Meaning	Example	
		Book syntax	Sample entry
Single braces	Single braces represent group-related items that are alternatives. You must enter one of the items. You cannot enter more than one.	{COMCHECK COMK}	COMK
Stacked braces	Stacked braces represent group related items that are alternatives. You must enter one of the items. You cannot enter more than one.	MN {DSNAME} {SPACE } {STATUS}	MN SPACE
Or-bar ()	An or-bar indicates a mutually exclusive choice. When used with brackets, enter one or none of the items. When used with braces, you must enter one of the items.	ACTIVATE RECOVER=SOURCE	RECOVER=SOURCE
Stacked items with or-bars () and brackets	Stacked items with or-bars indicates a mutually-exclusive choice. Enter one or none of these items.	CD RESET [,SDUMP] ,SYSABEND ,SYSUDUMP ,SYSMDUMP ,ALL	CD RESET,SYSUDUMP

System Command Formats

Two system command formats are defined.

Typical Format

Most system commands can use the format shown in Figure 4-1.

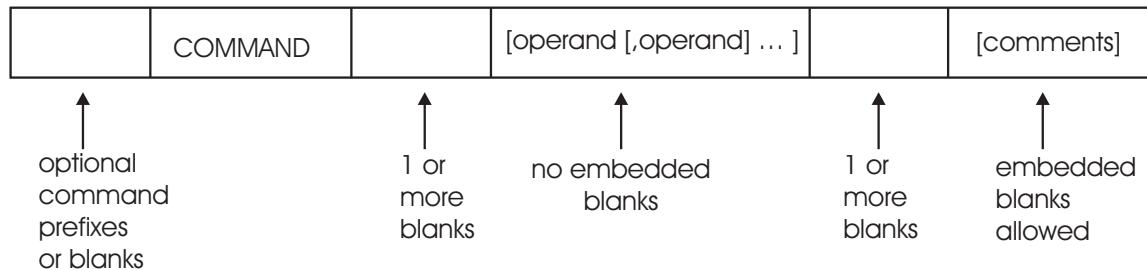


Figure 4-1. One System Command Format

The following restrictions apply to commands using this format:

1. Enter only one command per line. Use a maximum of 126 characters from a console, or 80 characters through a card reader.
2. To include a comment on a command when you have specified no operands, insert the following after the command: a blank, then a comma, then another blank, and then the comment. The comment may contain embedded blanks.

A Second Format

Figure 4-2 shows a format required by some system commands including DISPLAY PROD, DISPLAY PROG, DISPLAY RTLS, and SETPROG.

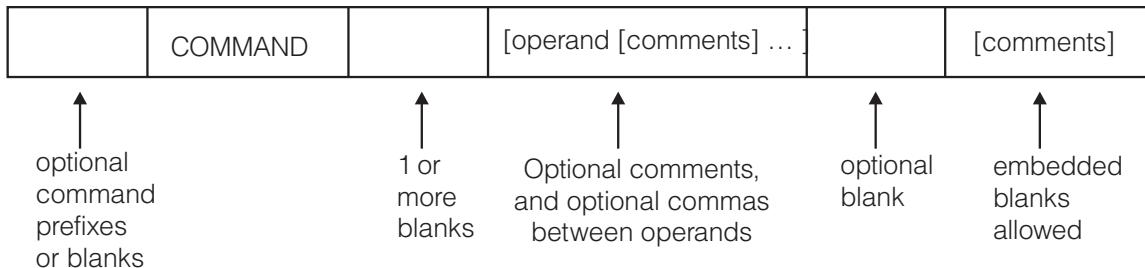


Figure 4-2. A Second System Command Format

This second format provides the opportunity to include a comment after the command and each operand within the command. These restrictions apply:

1. You may, but do not have to use a comma between operands. Examples:

```
D PROG APF  
D PROG,APF
```

2. This format requires that each comment be contained between a slash-asterisk and asterisk-slash pair. Comments may contain embedded blanks. Examples:

```
D PROG APF /* comments */  
D PROG /*comment */ APF /* comment */
```

ACTIVATE Command

Use the ACTIVATE command to activate or test a new I/O configuration definition dynamically.

Restrictions

For a list of restrictions on the ACTIVATE command, see *z/OS HCD Planning*.

Attention: An ACTIVATE command may still be active as a task in IOSAS after the command task has been abended with a CMDS ABEND.

Syntax

The complete syntax for the ACTIVATE command is:

```
ACTIVATE { [,IODF=xx] [,EDT=xx] [,PROC=procname] [,CFID=id] }
          [,RECOVER=SOURCE|TARGET] [,ACTIOCDS=xx]
          {[ [,SOFT[=VALIDATE|=NOVALIDATE] ]
            |,TEST
            |,FORCE
            |,FORCE={DEVICE
                      {CANDIDATE
                        {(DEVICE,CANDIDATE)}
                        {(CANDIDATE,DEVICE)}}
                      }}}
```

Note: Do not specify a comma before the first parameter following ACTIVATE.

Parameters

IODF=xx

Specifies the two-character suffix of the target IODF data set name (IODFx) that contains the configuration definition the system is to activate. When this keyword is omitted, the system defaults to the active IODF data set name.

EDT=xx

Specifies the eligible devices table (EDT) that the system is to construct from the target IODF. If you omit this keyword, the system uses the active EDT identifier.

PROC=*procname*

Indicates the eight-byte name of the processor definition in the target IODF. If you omit this keyword, the system will use the active processor name.

CFID=*id*

Specifies the eight-byte configuration identifier that indicates the operating system definition in the target IODF. If you omit this keyword, the system defaults the configuration identifier as follows:

- When the target IODF has only one configuration identifier, it becomes the default, otherwise, the current configuration identifier is the default.

RECOVER=

Allows the installation to continue a dynamic change that did not complete due to a hardware, software, or PR/SM failure. You can specify:

- SOURCE to retry the original I/O configuration
- TARGET to retry the new I/O configuration
- default:
 - Retry TARGET IODF if ACTIVATE failed during advance

ACTIVATE Command

- Retry SOURCE IODF if ACTIVATE failed while backing out.

ACTIOCDS=xx

Specifies the two-character IOCDS name that the system is to activate. Upon successful completion of the ACTIVATE command, the default IOCDS for the next power-on-reset will be *xx*. It does not make the I/O configuration definition stored in the IOCDS the active one.

For the IOCDS activate process to be successful, the processor token in the target IOCDS must match the current processor token in the Hardware System Area (HSA). This means that the IOCDS that is being activated has an I/O configuration definition that matches the I/O configuration currently active in the channel subsystem.

When you specify ACTIOCDS, you cannot specify TEST.

SOFT

Specifies a dynamic change to the software I/O configuration, which updates the I/O configuration only to the operating system. To change a software and hardware I/O configuration dynamically, omit the SOFT keyword.

When you specify SOFT, you cannot specify FORCE.

When you specify SOFT without any parameters, it is the same as specifying SOFT=VALIDATE.

=VALIDATE or =NOVALIDATE

Allows you to specify whether or not the system is to validate that any specified hardware elements to be deleted are offline and available, and that there is sufficient HSA space available to accommodate the hardware changes.

When a dynamic change is made to the I/O configuration for a processor complex running in LPAR mode, a change to the software I/O configuration is performed for the first N-1 logical partitions, followed by a hardware and software change for the Nth logical partition. By specifying the SOFT keyword (or SOFT=VALIDATE) when changing the I/O configuration on the N-1 logical partitions, you can determine early on whether there will be sufficient HSA space available for the subsequent software and hardware I/O configuration changes on the Nth logical partition.

Specifying SOFT=VALIDATE also ensures that the required processing for changes to coupling facility elements (CF control units or CF channel paths) will be executed. SOFT=VALIDATE is a requirement in all N-1 partitions when you make changes to coupling facility elements.

TEST

Specifies test mode to check, but not to change, the configuration. The system checks include whether:

- The dynamic change will fit into the current HSA
- The target IODF exists
- The target IODF contains the target EDT
- The target IOCDS is a valid data set
- The device support code supports devices being dynamically added or deleted
- The devices to be deleted are offline
- The paths to be deleted are offline

If you are performing a full dynamic activate, the system provides a list showing which channels and devices will be added, deleted, or changed during activation.

Warning If you run the ACTIVATE command with the TEST option and the system detects no errors, there is still no guarantee that ACTIVATE will work without TEST.

When you specify TEST, you cannot specify ACTIOCDS or FORCE.

FORCE

Specify that the system makes it possible to delete hardware resources that might offset other partitions.

You must specify FORCE if your processor complex is running in LPAR mode, and you want to activate a target IODF to delete one or more I/O components. You can also specify FORCE to activate a target IODF to delete a logical partition from a device candidate list. These deletions may be explicit or implicit due to changes in the definitions for some I/O components. When you specify FORCE, you cannot specify SOFT or TEST.

If your processor complex has Enterprise Systems Connection (ESCON) Multiple Image Facility (EMIF) capability, you can specify FORCE to get the results described in Table 4-3.

For information about ESCON Multiple Image Facility (EMIF), see *z/OS HCD Planning*. For information about access lists and candidate lists, see *z/OS HCD User's Guide*.

Table 4-3. Specifying FORCE with EMIF

To do the following:	Specify FORCE as follows:
Delete no I/O components, and do either of the following: <ul style="list-style-type: none">• Delete no logical partitions from the access or candidate list of a channel path.• Delete one or more logical partitions from the access or candidate list of a channel path offline to all of those logical partitions. IBM recommends that you take the channel path offline before issuing the command.	Do not specify FORCE.
Delete no I/O components, and delete one or more logical partitions from the access or candidate list of a channel path online to any of those logical partitions. IBM does not recommend this action.	FORCE=CANDIDATE
Delete one or more I/O components, and do either of the following: <ul style="list-style-type: none">• Delete no logical partitions from the access or candidate list of a channel path.• Delete one or more logical partitions from the access or candidate list of a channel path offline to all of those logical partitions. IBM recommends that you take the channel path offline before issuing the command.	FORCE or FORCE=DEVICE
Delete one or more I/O components, and delete one or more logical partitions from the access or candidate list of a channel path online to any of those logical partitions. IBM does not recommend this action.	FORCE=(DEVICE,CANDIDATE) or FORCE=(CANDIDATE,DEVICE)

ACTIVATE Command

Table 4-3. Specifying FORCE with EMIF (continued)

To do the following:	Specify FORCE as follows:
Delete one or more logical partitions from the device candidate list and delete no other I/O components.	FORCE or FORCE=DEVICE

Note: Before activating the new configuration, you may have to configure offline affected channel paths or vary offline affected devices. See *z/OS HCD Planning* for details about avoiding disruptions to I/O operations during dynamic changes.

Example 1

To ACTIVATE the A0 IOCDS, enter:

```
ACTIVATE ACTIOCDS=A0
```

Example 2

To ACTIVATE the configuration definition COMPUT22, contained in the IODF with suffix 03, enter:

```
ACTIVATE IODF=03,CFID=COMPUT22
```

Example 3

To perform a test ACTIVATE to processor definition PROC1001 contained in the currently active IODF, enter:

```
ACTIVATE PROC=PROC1001,TEST
```

Example 4

To ACTIVATE an IODF with suffix 04, which deletes one or more I/O components from the I/O configuration, enter:

```
ACTIVATE IODF=04,FORCE  
or  
ACTIVATE IODF=04,FORCE=DEVICE
```

CANCEL Command

Use the CANCEL command to end an active job, started task, or time-sharing user immediately. The table that follows summarizes the tasks that the CANCEL command can perform. Following the table are usage notes, the complete command syntax, definition of parameters, and examples of use.

If the program that supports the job or started task was designed to recognize the STOP command, use the STOP command before using the CANCEL command. If the CANCEL command fails several times, consider using the FORCE command.

Table 4-4. CANCEL Command Tasks

Task - Immediately Terminate:	Syntax
<ul style="list-style-type: none"> • A job in execution • A running Advanced Program-to-Program Communication/MVS (APPC/MVS) transaction program • A started task 	CANCEL jobname
<ul style="list-style-type: none"> • A time-sharing user 	CANCEL U=userid
<ul style="list-style-type: none"> • A started task • A MOUNT command • An external writer allocation • The output processing for a job • A z/OS UNIX process 	CANCEL identifier

Notes:

1. If your system was part of a global resource serialization ring (GRS=START, GRS=JOIN or GRS=TRYJOIN was specified at IPL) and the system is either inactive or quiesced (by entering the VARY GRS(system name),QUIESCE command), the CANCEL command might not work for jobs that own any global resources. Use DISPLAY GRS to determine GRS status.
2. If a job is running, you can end it using either the CANCEL system command or the appropriate subsystem command. However, if the job is not running, you must CANCEL the job using the subsystem command.
3. The CANCEL command issues an ABEND with either code 122 or 222 to abnormally end a job step or time-sharing user. The ABEND is asynchronous and might result in additional errors, depending on which programs were active at the time of the request. You might need to issue additional CANCEL commands to completely end the job.
4. Entering the CANCEL command during device allocation terminates the external writer as well as the unit of work. Entering this command when the external writer is processing output for a job terminates the output processing but leaves the external writer to process other data sets.
5. When you cancel a MOUNT command for a tape unit, the MOUNT command can end before the volume has been mounted. If the MOUNT command has ended and the mount request is not satisfied, issue the UNLOAD command to free the tape unit.

CANCEL Command

Syntax

The complete syntax for the CANCEL command is:

```
C {jobname          } [,DUMP] [,A=asid] [,ARMRESTART]
{U=userid         }
{[jobname.]identifier}
```

Parameters

jobname

The name of the batch job, started task, or APPC/MVS transaction program to be canceled.

The job name for a given started task can be assigned based on a variety of inputs. These inputs are examined in the following order, so that if item #1 is not specified, item #2 is used. If neither #1 nor #2 is specified, then #3 is used, and so on.

1. The jobname specified in the JOBNAME= parameter of the START command
or
The identifier specified on the START command.
2. The jobname specified on the JOB JCL statement within the member.
3. The device number specified on the START command, or the device number associated with the device type specified on the START command
or
The device number associated with the device type specified on the START command.
4. The device number associated with the IEFRDER DD statement within the member.
5. The member name.

U=userid

The user ID of the time-sharing user you want to cancel.

If the user is just logging on and does not yet have a unique name, you must find out the address space identifier for the user (see the explanation under A=asid) and use the following version of the command:

CANCEL U=*LOGON*,A=asid

[jobname.]identifier

The identifier for the unit of work that you want to cancel, optionally preceded by the job name.

The following types of identifiers can be used:

- The identifier that was specified on the START command.
- */devnum*, the device number specified when the START or MOUNT command was entered. The device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). You can precede the device number with a slash to prevent ambiguity between the device number and a device type or identifier.
- *devicetype*, the type of device specified when the START or MOUNT command was issued.

If no identifier was specified on the START command, the system assigns temporary identifier "STARTING" to the unit of work, until the system can assign an identifier according to the following order of precedence:

1. If an identifier was not specified on the START command, the identifier is the device type (for example, 3410) or device number (for example, X'0000') specified on the START or MOUNT command.
2. If an identifier, a device type, or a device number was not specified on the START or MOUNT command, the identifier is the device type specified on an IEFRDER DD statement (invoking a cataloged procedure) in the JCL.
3. If none of the above was specified, the identifier defaults to the job name.

When you specify *jobname.identifier*, then *identifier* can be represented by any of the following:

- An asterisk
- One or more characters from the beginning of the identifier, followed by an asterisk
- The entire identifier

When you specify an asterisk, the system responds with message IEE422I.

Attention: When you use the asterisk format, the command affects *all* started tasks that begin with the specified characters. Device numbers are assumed to be four-digit numbers; for example, /13* would match on 1301, 1302, and so on, but would not match on 13C, because 13C is interpreted as 013C.

Specifying both the job name and the entire identifier causes the command to take effect if **one and only one** work unit with that combination of job name and identifier is running. For the case where more than one work units with the same combination of job name and identifier are running, see "A=asid" below.

DUMP

A dump is to be taken. The type of dump (SYSABEND, SYSUDUMP, or SYSDUMP) depends on the JCL for the job. A dump request is only valid when made while the job is running. Dumps are not taken during job allocation or deallocation.

Note: You can use DUMP with any of the other CANCEL parameters.

A=asid

The hexadecimal address space identifier of the work unit you want to cancel.

If more than one work unit is running with the same job name, identifier, combination of job name and identifier, or user ID that you specified on the CANCEL command, the system rejects the command because it does not know which work unit to cancel. To avoid this, you must add the parameter A=asid to your original CANCEL command in order to specify the address space identifier of the work unit.

Note: If the asterisk format is used, you will not be prompted for A=asid. Rather, all work units meeting the specified criteria will be canceled.

You can use the CANCEL operator command to cancel z/OS UNIX address spaces. Each address space is equivalent to a z/OS UNIX process.

To find out the address space identifier for a unit of work, you can use the DISPLAY command as follows:

CANCEL Command

DISPLAY JOBS,ALL

Lists the address space identifiers for all batch jobs and started tasks.

DISPLAY ASCH,ALL

Lists the address space identifiers for all APPC/MVS transaction programs.

DISPLAY TS,ALL

Lists the address space identifiers for all logged-on time-sharing users.

DISPLAY OMVS,ASID=ALL or DISPLAY OMVS,A=ALL

Lists the address space identifiers for all z/OS UNIX processes.

Note: A=*asid* can be used with any of the other CANCEL parameters *except if you specify jobname.identifier with an asterisk (for example, CANCEL aor2.t1*)*.

ARMRESTART

Indicates that the batch job or started task should be automatically restarted after the cancel completes, if it is registered as an element of the automatic restart manager. If the job or task is not registered or if you do not specify this parameter, MVS will not automatically restart the job or task.

Example 1

Cancel the job named EXAMPLE and take a dump.

```
c example,dump
```

Example 2

Cancel the job named EXAMPLE. Whether you get a dump or not depends on the system routine in control when you enter the command.

```
c example
```

Example 3

Of all jobs named EXAMPLE in the system, cancel only the one whose address space identifier is 7F.

```
c example,a=7F
```

Example 4

Log off the system the user just logging on who has an address space identifier of 3D but does not yet have a unique user identifier.

```
c u=*logon*,a=3d
```

Example 5

Log user A237 off the system.

```
c u=a237
```

Example 6

Log user A237 off the system and take a dump.

```
c u=a237,dump
```

Example 7

Cancel the MOUNT command that requests a volume to be mounted on device number 232, enter:

```
c 232
```

Example 8

Cancel the MOUNT command that requests a volume to be mounted on a 3330 device type.

```
c 3330
```

Example 9

End the device allocation for a writer with device number 00E.

```
c 00e
```

Example 10

End the output processing being done for device number 00E and cause another output data set to be processed.

```
c 00e
```

Example 11

End the output processing being done for device number 3480 and cause another output data set to be processed.

```
c /3480
```

Example 12

Of all the transaction programs running with the job name MAIL, end only the one whose address space identifier is 2C, which is the APPC/MVS scheduler (ASCH) initiator ASID.

```
C mail,a=2c
```

Example 13

End the device allocation for a writer on device number F00E.

```
c /f00e
```

Example 14

There are several tasks running with jobname AOR2. End all of those tasks.

```
c aor2.*
```

Example 15

There are several tasks running with jobname AOR2. Some of those tasks have *identifiers* beginning T1. End only those specific tasks.

```
c aor2.t1*
```

Example 16

The following example shows an operator session that cancels a process that is running the shell command **sleep 6000** for the TSO/E user CHAD.

CANCEL Command

```
DISPLAY OMVS,U=CHAD
```

```
BPX0001I 17.12.23 DISPLAY OMVS 700
OMVS      ACTIVE
USER     JOBNAM  ASID      PID      PPID   STATE   START    BPXPRMHF
CHAD     CHAD    001D    262147      1  RI     17.00.10    C
CHAD     CHAD    001B    131076      5  SI     17.00.10    .111
      LATCHWAITPID= 0 CMD=sleep 6000
CHAD     CHAD    0041      5  262147   IW     17.00.10    .596
      LATCHWAITPID= 0 CMD=-sh
CHAD     CHAD3   001B    131076      5  SI     17.00.10    .111
      LATCHWAITPID= 0 CMD=sleep 6000
```

If you want to cancel only the process that is running the shell command **sleep 6000**, enter:

```
CANCEL CHAD3
```

If you want to cancel the TSO/E user CHAD altogether, enter:

```
CANCEL U=CHAD
```

CHNGDUMP Command

Use the CHNGDUMP command to change the mode and system dump options list for any dump type, or to request structures to be dumped when one or more systems connected to a coupling facility fail. The dump types are SDUMP, SYSABEND, SYSUDUMP, and SYSDUMP. If you issue multiple CHNGDUMP commands, the changes to the system dump options are cumulative. Table 4-5 summarizes the information that the CHNGDUMP command provides. Use it to access the pages on which you can find details about a particular use of the CHNGDUMP command.

Table 4-5. Summary of the CHNGDUMP Command

Command	Topic
CHNGDUMP DEL	"Removing Options from or Resetting the System Dump Options Lists" on page 4-26
CHNGDUMP RESET	"Resetting Dump Mode to ADD and the Dump Options to Initial Values" on page 4-32
CHNGDUMP SET	"Setting the Dump Modes and Options" on page 4-34

Dump Options and Modes

The system checks the dump mode and dump options each time the system or a user requests a dump. The dump mode determines whether the system accepts either a dump request or the options a dump request specifies. The starting dump mode for all four dump types is ADD.

The dump options, whether taken from a system dump options list or from a dump request, specify, for each dump type, the data areas to dump. MVS™ sets up system dump options lists each time you initialize the system. These lists specify the dump mode and dump options in effect for each dump type. The system finds the starting dump options lists for the SYSABEND, SYSUDUMP, and SYSDUMP dump types in parmlib members IEAABD00, IEADMP00, and IEADMR00 respectively. Because the SDUMP dump type has no corresponding parmlib member, it starts with an empty dump options list.

Dump Modes

In addition to ADD, other possible dump modes are OVER and NODUMP. The meaning of each mode is:

- ADD — When a dump is requested for a dump type that is in ADD mode, the system merges the options specified on the dump request with the options specified in the system dump options list for that dump type. The merged options determine the data areas to dump. If an option specified on the dump request conflicts with an option in the options list, the system uses the option in the options list.
- OVER — When a dump is requested for a dump type that is in OVER (override) mode, the system ignores the options specified on the dump request and uses only the options specified in the system dump options list for that dump type combined with the override options to determine the data areas to dump.
- NODUMP — When a dump is requested for a dump type that is in NODUMP mode, the system ignores the request and does not take a dump.

CHNGDUMP Command

Changing the Dump Mode and Options

You can change the dump mode or options for a dump type. Before making a change, however, issue DISPLAY DUMP,OPTIONS to see the current mode and options.

Changing the mode of a dump type can also affect its system dump options list as follows:

- If you change the mode for a dump type from OVER to ADD, the system adds the dump type's parmlib options to the dump type's system dump options list. The dump type's system dump options list then contains both the dump type's parmlib options **and** any options set by previous CHNGDUMP commands.
- If you change the mode for a dump type from ADD to OVER, the system removes the dump type's parmlib options from the dump type's system dump options list. The dump type's system dump options list then contains only the options set by previous CHNGDUMP commands.
- If you change the mode for a dump type to NODUMP, the system empties the dump type's system dump options list.

If you make an error entering a CHNGDUMP command, the system rejects the entire command and issues an error message.

Scope in a Sysplex

The CHNGDUMP command has sysplex scope only when all systems are connected to the same coupling facilities, and you specify ,SDUMP,SYSFAIL,STRLIST=. See "Using Commands That Have Sysplex Scope" on page 1-11 for an explanation of sysplex scope.

Syntax

The syntax for each variation of the CHNGDUMP command (CD DEL, CD RESET, and CD SET) is shown immediately preceding its respective parameter list.

CHNGDUMP or CD

Removing Options from or Resetting the System Dump Options Lists

Use the CHNGDUMP DEL command to remove specified options from a dump options list or to reset all dump options lists to values specified at system initialization.

```
CD DEL[,SDUMP[=(option[,option]...)]  
      [,Q={YES|NO}]  
      [,TYPE={XMEM|XMEME}]  
      [,ALL]  
      [,SYSFAIL,STRLIST={ALL|(STRNAME=strname[,STRNAME=strname]...) }]  
      [{,SYSABEND}[,SDATA=(option[,option]...)]]  
  
      [{,SYSUDUMP} |,PDATA=(option[,option]...)  
      |,ALL]  
  
      [,SYSMDUMP[=(option[,option]...)]  
      |,ALL]  
  
      [,ALL]
```

Parameters

DEL

Set the dump mode for each dump type to ADD and reset all system dump options lists to the values established during system initialization. (The system gets the initial dump options for SYSABEND, SYSUDUMP, and SYSDMDUMP from parmlib members IEAABD00, IEADMP00, and IEADMRO0 respectively. The system empties the SDUMP dump options list.)

SDUMP

Set the dump mode for the SDUMP dump type to ADD, and empty the SDUMP system dump options list.

SDUMP,SYSFAIL,STRLIST={ALL or STRNAME=*strname,...}*

To delete STRNAME specifications from the dump options list. You can delete any or all structure name specifications from the dump options list. For a complete description of the keyword specifications “see the SET,SDUMP parameter” on page 4-37.

SDUMP=(*options***)**

Remove the specified options from the SDUMP system dump options list. If the SDUMP dump mode is OVER, and previous CHNGDUMP commands have removed all dump options from the SDUMP dump options list, this command has no effect and leaves the SDUMP dump options list empty. See “Options for SDUMP, SYSABEND, SYSUDUMP, and SYSDMDUMP” on page 4-29 for SDUMP options you can specify.

SDUMP,Q=YES or NO

Specifies whether or not SDUMP is to quiesce the system—set it to nondispatchable (Q=YES) or leave the system dispatchable (Q=NO)—while dumping the contents of the SQA or CSA. For further information, see “Setting the Dump Modes and Options” on page 4-34.

SDUMP,TYPE=XMEM

Turn off “cross memory” in the SDUMP system dump options list.

SDUMP,TYPE=XMEME

Turn off “cross memory at the time of the error” in the SDUMP system dump options list.

SDUMP,ALL

Same as DEL,SDUMP.

SYSABEND

Set the dump mode for the SYSABEND dump type to ADD, and reset the SYSABEND system dump options list to the values established during system initialization. (The system gets the initial dump options for SYSABEND from the IEAABD00 parmlib member.)

SYSABEND,SDATA=(*options***)**

Remove any of the specified SDATA options that previous CHNGDUMP commands put in the SYSABEND system dump options list. Parmlib options do not change. If the SYSABEND dump mode is OVER and previous CHNGDUMP commands have removed all SDATA and PDATA dump options from the SYSABEND dump options list, reset the list to the parmlib options. See “Options for SDUMP, SYSABEND, SYSUDUMP, and SYSDMDUMP” on page 4-29 for SYSABEND options you can specify.

SYSABEND,PDATA=(*options***)**

Remove any of the specified PDATA options that previous CHNGDUMP commands put in the SYSABEND system dump options list. Parmlib

CHNGDUMP Command

options do not change. If the SYSABEND dump mode is OVER and previous CHNGDUMP commands have removed all SDATA and PDATA dump options from the SYSABEND dump options list, reset the list to the parmlib options. See “Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP” on page 4-29 for SYSABEND options you can specify.

SYSABEND,ALL

Same as DEL,SYSABEND.

SYSUDUMP

Set the dump mode for the SYSUDUMP dump type to ADD, and reset the SYSUDUMP system dump options list to the values established during system initialization. (The system gets the initial dump options for SYSUDUMP from the IEADMP00 parmlib member.)

SYSUDUMP,SDATA=(*options*)

Remove any of the specified SDATA options that previous CHNGDUMP commands put in the SYSUDUMP system dump options list. Parmlib options do not change. If the SYSUDUMP dump mode is OVER and previous CHNGDUMP commands have removed all SDATA and PDATA dump options from the SYSUDUMP dump options list, reset the list to the parmlib options. See “Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP” on page 4-29 for SYSUDUMP options you can specify.

SYSUDUMP,PDATA=(*options*)

Remove any of the specified PDATA options that previous CHNGDUMP commands put in the SYSUDUMP system dump options list. Parmlib options do not change. If the SYSUDUMP dump mode is OVER and previous CHNGDUMP commands have removed all SDATA and PDATA dump options from the SYSUDUMP dump options list, reset the list to the parmlib options. See “Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP” on page 4-29 for SYSUDUMP options you can specify.

SYSUDUMP,ALL

Same as DEL,SYSUDUMP.

SYSMDUMP

Set the dump mode for the SYSMDUMP dump type to ADD, and reset the SYSMDUMP system dump options list to the values established during system initialization. (The system gets the initial dump options for SYSMDUMP from the IEADMRO0 parmlib member.)

SYSMDUMP=(*options*)

Remove any of the specified options that previous CHNGDUMP commands put in the SYSMDUMP system dump options list. Parmlib options do not change. If the SYSMDUMP dump mode is OVER and previous CHNGDUMP commands have removed all dump options from the SYSMDUMP dump options list, reset the list to the parmlib options. See “Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP” on page 4-29 for SYSMDUMP options you can specify.

SYSMDUMP,ALL

Same as DEL,SYSMDUMP.

ALL

Same as DEL.

Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP

The options that you can specify on the CHNGDUMP command follow. The default options for the IEAADB00, IEADMP00, and IEADMRO0 parmlib members are CB, DM, ENQ, ERR, IO, JPA, LPA, LSQA, NUC, PSW, REGS, RGN, SA, SPLS, SQA, SUM, SWA, and TRT:

SDUMP options:

ALLNUC	All of the DAT-on nucleus, including page-protected areas, and all of the DAT-off nucleus.
ALLPSA or NOALLPSA or NOALL	Prefix storage area for all processors. NOALLPSA or NOALL specifies that these areas are not to be dumped.
COUPLE	XCF related information in the sysplex.
CSA	Common storage area.
GRSQ	Global resource serialization (ENQ/DEQ/RESERVE) queues.
	Note that when the GRS is running in STAR mode, the output of the GRSDATA subcommand is dependent on the GRSQ option setting of the parmlib member GRSCNFxx. For more information about the GRSCNFxx GRSQ setting, see the <i>z/OS MVS Planning: Global Resource Serialization</i> .
LPA	Link pack area.
LSQA	Local system queue area.
NUC	Non-page-protected areas of the DAT-on nucleus.
PSA	Prefix storage area of the dumping processor.
RGN	Entire private area.
SERVERS	Requests that the registered IEASDUMP.SERVER dynamic exits receive control.
	Be warned, however, this setting also causes all synchronous dumps to be converted into asynchronous dumps, which can result in unexpected behavior if the dump requestor was expecting the dump to complete before control was returned to the program. The use of this option is therefore not recommended for general use, so IBM has defaulted the SERVERS option for all operator initiated and SLIP trap SVC dumps.
SQA or NOSQA	System queue area. NOSQA specifies that this data not be dumped.
SUMDUMP or SUM	Requests the summary dump function. For a description of NOSUM the summary dump function, see <i>z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU</i> .
NOSUM or NOSUMDUMP	Requests that the function not be performed.
SWA	Scheduler work area.
TRT	GTF, system trace, and master trace data.
WLM	Workload management related information in the sysplex.

CHNGDUMP Command

XESDATA	XES-Related information in the sysplex.
SDATA options for SYSABEND dump type: Request application related system storage, including key 0 areas, that the application program or programmer can see.	
ALLSDATA	Sets all of the other SDATA options except NOSYM and ALLVNUC.
ALLVNUC	All of the DAT-on nucleus, including page-protected areas.
<u>CB</u>	Format of task-related control blocks.
<u>DM</u>	Data management control blocks.
<u>ENQ</u>	Global resource serialization control blocks for the current task.
<u>ERR</u>	RTM control blocks.
<u>IO</u>	I/O supervisor control blocks.
<u>LSQA</u>	Local system queue area.
NOSYM	Symptom dump is not to be produced.
NUC	Non-page-protected areas of the DAT-on nucleus.
PCDATA	Program call data for the current task.
SQA	System queue area.
<u>SUM</u>	Summary dump data.
SWA	Scheduler work area.
<u>TRT</u>	GTF and system trace data. (For an authorized user, system trace data is for all address spaces in the system. For an unauthorized user, system trace data is for the user's address space only. The GTF data is for the user's address space only.)
SDATA options for SYSUDUMP dump type: Request application related system storage, including key 0 areas, that the application program or programmer can see.	
ALLSDATA	Sets all of the other SDATA options except NOSYM and ALLVNUC.
ALLVNUC	All of the DAT-on nucleus, including page-protected areas.
CB	Format of task-related control blocks.
DM	Data management control blocks.
ENQ	Global resource serialization control blocks for the current task.
ERR	RTM control blocks.
IO	I/O supervisor control blocks.
LSQA	Local system queue area.
NOSYM	Symptom dump is not to be produced.
NUC	Non-page-protected areas of the DAT-on nucleus.
PCDATA	Program call data for the current task.
SQA	System queue area.
<u>SUM</u>	Summary dump data.
SWA	Scheduler work area.

TRT GTF and system trace data. (For an authorized user, system trace data is for all address spaces in the system. For an unauthorized user, system trace data is for the user's address space only. The GTF data is for the user's address space only.)

| **PDATA options for SYSABEND dump type:** Request areas of application storage that are accessible using problem program keys.

ALLPDATA Sets all of the PDATA options.

ALLPA Sets both the LPA and JPA options.

JPA job pack area.

LPA Link pack area for this job.

PSW Program status word.

REGS General registers.

SA Save area trace (long form).

SAH Save area trace (short form).

SPLS Allocated storage subpools.

SUBTASKS Program data for the main task and all subtasks of this job. (SUBTASKS is always included for abends with a system completion code of X'22'.)

| **PDATA options for SYSUDUMP dump type:** Request areas of application storage that are accessible using problem program keys.

ALLPDATA Sets all of the PDATA options.

ALLPA Sets both the LPA and JPA options.

JPA Job pack area.

LPA Link pack area for this job.

PSW Program status word.

REGS General registers.

SA Save area trace (long form).

SAH Save area trace (short form).

SPLS Allocated storage subpools.

SUBTASKS Program data for the main task and all subtasks of this job. (SUBTASKS is always included for abends with a system completion code of X'22'.)

SYMDUMP options:

ALL Sets all of the other SYMDUMP options except NOSYM and ALLNUC.

ALLNUC All of DAT-on nucleus, including page-protected areas, and all of the DAT-off nucleus.

CSA The portions of the common storage area that are not fetch-protected.

GRSQ Global resource serialization (ENQ/DEQ/RESERVE) queues.

CHNGDUMP Command

Note that when the GRS is running in STAR mode, the output of the GRSDATA subcommand is dependent on the GRSQ option setting of the parmlib member GRSCNFxx. For more information about the GRSCNFxx GRSQ setting, see the *z/OS MVS Planning: Global Resource Serialization*.

LPA	Link pack area for this job.
LSQA	Local system queue area.
NOSYM	Symptom dump is not to be produced.
NUC	Non-page-protected areas of the DAT-on nucleus.
RGN	Entire private area.
SQA	System queue area.
SUM	Requests the summary dump function. For a description of the summary dump function, see <i>z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU</i> .
SWA	Scheduler work area.
TRT	System trace data. (For an authorized user, system trace data is for all address spaces in the system. For an unauthorized user, system trace data is for the user's address space only.)

Resetting Dump Mode to ADD and the Dump Options to Initial Values

Use the CHNGDUMP RESET command to reset the dump mode to ADD and the dump options list to values established during system initialization.

```
CD RESET[,SDUMP      ]
          |,SYSABEND
          |,SYSUDUMP
          |,SYSMDUMP
          |,ALL
```

RESET

Set the dump mode for each dump type to ADD, and reset the system dump options list for each type to the values established during system initialization. (The system gets the initial dump options for SYSABEND, SYSUDUMP, and SYSMDUMP from parmlib members IEAABD00, IEADMP00, and IEADMRO0 respectively. The system empties the SDUMP dump options list.)

RESET,SDUMP or SYSABEND or SYSUDUMP or SYSMDUMP

Set the dump mode for the specified dump type to ADD, and reset the dump type's system dump options list to the values established during system initialization. (The system gets the initial dump options for SYSABEND, SYSUDUMP, and SYSMDUMP from parmlib members IEAABD00, IEADMP00, and IEADMRO0 respectively. The system empties the SDUMP dump options list.)

RESET,ALL

Same as RESET.

Example: How CHNGDUMP Commands Affect Dump Modes and Options

Table 4-6 (using SYSABEND) shows how dump modes and system dump options are set during system initialization and then changed by CHNGDUMP commands or options specified on ABEND macro dump requests. The figure assumes that parmlib member IEAABD00 specifies dump options CB, DM, ENQ, ERR, IO, LSQA, SUM, and TRT. The figure lists system and operator actions and explains each action the system takes.

- The FUNCTION column lists the IPL process, CHNGDUMP commands, and dump requests (from the ABEND macro instruction) as they occur.
- The OPTIONS column identifies the SYSABEND dump options in effect at each point in the example.
- The MODE column identifies the dump mode in effect at each point in the example.

Table 4-6. Example of How CHNGDUMP Commands Affect Dump Modes and Options

FUNCTION (* indicates operator commands/actions)	OPTIONS	MODE
* IPL During IPL, parmlib options are set on, and mode is set to ADD.	CB IO DM LSQA ENQ SUM ERR TRT	ADD
ABEND SDATA=(NUC,SWA) Because mode is ADD, options specified on dump request are added to options set on in options list to determine areas to dump.	CB LSQA DM NUC ENQ SUM ERR SWA IO TRT	ADD
* CD SET,SYSABEND,SDATA=PCDATA An additional option is set on in options list.	CB LSQA DM PCDATA ENQ SUM ERR TRT IO	ADD
ABEND Because dump request specified no additional options, only options set on in options list determine areas to dump.	CB LSQA DM PCDATA ENQ SUM ERR TRT IO	ADD
ABEND SDATA=(NUC,SWA) Because mode is ADD, options specified on dump request are added to options set on in options list to determine areas to dump.	CB NUC DM PCDATA ENQ SUM ERR SWA IO TRT LSQA	ADD
* CD SET,SYSABEND,OVER,SDATA=(CB,IO) Mode is changed to OVER and parmlib options are deleted from options lists. Only options set by previous CD commands remain on in options list; CD commands are cumulative. Options in options list are combined with override options to determine areas to dump.	CB IO PCDATA	OVER
ABEND SDATA=(SWA,TRT) Because mode is OVER, options specified on dump request are ignored. Options set on in options list determine areas to dump.	CB IO PCDATA	OVER

CHNGDUMP Command

Table 4-6. Example of How CHNGDUMP Commands Affect Dump Modes and Options (continued)

FUNCTION (* indicates operator commands/actions)	OPTIONS	MODE
* CD SET,SYSABEND,SDATA=(ENQ, LSQA),OVER Two more options are set on in options list, and mode is unchanged.	CB LSQA ENQ PCDATA IO	OVER
* CD SET,SYSABEND,SDATA=NUC Mode is changed to ADD (the default) for SYSABEND, parmlib options are set on, and CD command options are set on. Previous CD command options remain on.	CB LSQA DM NUC ENQ PCDATA ERR SUM IO TRT	ADD
ABEND SDATA=(CB,SWA,NUC) Because mode is ADD, options specified on dump request are added to options set on in options list to determine areas to dump.	CB NUC DM PCDATA ENQ SUM ERR SWA IO TRT LSQA	ADD
* CD SET,SYSABEND,NODUMP Mode is changed to NODUMP. All options in options list are set off.		NODUMP
ABEND SDATA=(CB,SWA,NUC) Because mode is NODUMP, request is ignored.		NODUMP
* CD SET,SYSABEND,SDATA=SQA Mode is changed to ADD. Parmlib options on this CD command are set.	CB LSQA DM SQA ENQ SUM ERR TRT IO	ADD
* CD DEL,SYSABEND,SDATA=(DM,IO) Because DEL is specified, specified options in options list are set off.	CB SQA ENQ SUM ERR TRT LSQA	ADD
* CD DEL,SYSABEND,SDATA=(SQA, LSQA,TRT, CB,ENQ,ERR,SUM) Specified options are set off in options list.		ADD
ABEND SDATA=(SQA) The option specified in the dump request determines the area to dump. There are no options on in the options list.	SQA	ADD
* CD RESET,SYSABEND Mode and options list are reset to values established at system initialization.	CB IO DM LSQA ENQ SUM ERR TRT	ADD

Setting the Dump Modes and Options

Use the following form of the CHNGDUMP command to set the dump modes and put specified options in the dump options lists.

```

CD SET,{NODUMP} }
{OVER } }
{ADD } }
{SDUMP[=(option[,option]...)] }
  [,Q={YES|NO}]
  [,TYPE={XMEM|XMEME}]
  [,BUFFERS={nnnnK|nnnM}]
  [,MAXSPACE=xxxxxxxxM]
  [,MSGTIME=yyyyy]
  [,SYSFAIL,STRLIST=(s-option[,s-option]...)]
  [,NODUMP]
    |,OVER
    |,ADD
{{SYSABEND}[,{SDATA=(option[,option]...)}][,{NODUMP}] }
{{SYSUDUMP}|,{PDATA=(option[,option]...)}}|,OVER } }
|,ADD
{SYSMDUMP[=(option[,option]...)]}[,{NODUMP}] }
  |,OVER
  |,ADD
{ABDUMP,TIMENQ=yyyy}

```

Where **s-option** represents:

```

STRNAME=strname
  [,CONNNAME=conname] }

[,ACCESSTIME={ENFORCE|NOLIMIT|NOLIM}] }

[,LOCKENTRIES] }

[,USERCNTLS] }

[,EVENTQS] }

[,,(EMCONTROLS={ALL|(list)})] }

[,({COCLASS|STGCLASS|LISTNUM}={ALL|(list)})
  [,ADJUNCT={CAPTURE|DIRECTIO}][,ENTRYDATA={UNSERIALIZE|SERIALIZE}]] }
  [,SUMMARY] }

```

SET

Set the dump mode and put specified options in the system dump options list.

NODUMP

Set the SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP dump modes to NODUMP, and remove all options from the system dump options lists for these dump types.

ADD

Set the SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP dump modes to ADD. If any of these dump types were previously in OVER mode, add its parmlib dump options to its system dump options list.

OVER

Set the SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP dump modes

CHNGDUMP Command

to OVER. If any of these dump types were previously in ADD mode, remove its parmlib options from its system dump options list.

SDUMP

Set the SDUMP dump mode to ADD.

SDUMP,NODUMP

Set the SDUMP dump mode to NODUMP.

You cannot specify other parameters when specifying NODUMP. For example, you can specify CD SDUMP,NODUMP, but not CD SDUMP,Q=YES,NODUMP.

SDUMP,OVER or ADD

Set the SDUMP dump mode to the specified mode.

SDUMP,SYSFAIL,STRLIST=(structure names and options)

Set structures to be dumped when a single system fails or when all the systems connected to a coupling facility fail. SYSFAIL may not be specified with other dump options, and must be specified with STRLIST. The syntax for the STRLIST specification is identical to the DUMP command. SYSFAIL and STRLIST may be specified in ADD or OVER mode. In either mode, the STRLIST parameter list is saved and when a system fails or when all the systems fail, a dump is requested that includes the specified structures.

SDUMP=(options)

Put the specified options in the SDUMP system dump options list. See "Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP" on page 4-29 for SDUMP options you can specify.

SDUMP,TYPE=XMEM

Set the SDUMP system dump options list to "cross memory." Specifying TYPE=XMEM causes SVC dump to dump the cross memory address spaces that the caller has when SVC dump gets control.

SDUMP,TYPE=XMEME

Set the SDUMP system dump options list to "cross memory at the time of the error." Specifying TYPE=XMEME causes SVC dump to dump cross memory address spaces that the caller has when the error causing the dump occurs.

SDUMP,Q=YES or NO

Specifies whether or not SDUMP is to quiesce the system—set it to nondispatchable (Q=YES) or leave the system dispatchable (Q=NO)—while dumping the contents of the SQA or CSA.

Quiescing the system increases the ability to capture SQA and/or CSA data when a failure occurs. However, the system can appear hung when collecting large amounts of data. When the system is not quiesced, users are more likely to receive uninterrupted system service.

If the operator issuing the CHNGDUMP command specifies Q=YES, SDUMP will always quiesce the system, whether or not it dumps the SQA or CSA. That is, the command will override the specification in the program.

If the operator specifies Q=NO, SDUMP will not quiesce the system. An exception to this is that SDUMP *will* quiesce the system if the program specified to dump the SQA or CSA *and* it included the QUIESCE parameter on the SDUMP macro.

If the operator issuing the CHNGDUMP command specifies neither Q=YES nor Q=NO, the system will use the value the program specified for the QUIESCE parameter on the SDUMP macro.

SDUMP,BUFFERS=nnnnK or nnnM

Reserves the specified space in real storage frames for exclusive use by SVC dump. The amount of storage that is set aside is expressed in kilobytes (0K to 9999K) or in megabytes (0M to 999M). The default is 0K.

You can change the BUFFERS option at any time. Consider, however, specifying a large BUFFERS value can seriously degrade system performance because the reserved frames will be unavailable to any other systems or user applications.

SDUMP,MAXSPACE=xxxxxxxxM

Specifies the maximum amount of virtual storage that SVC dump can use to capture volatile virtual storage data, summary dump data, and component-specific data before writing the dump to DASD. The default value is 500 megabytes. The value that can be specified may range from 1 to 99999999 (with, or without, an M suffix). The new value takes effect immediately. If the value specified is lower than the space used, SVC dump will not continue to capture data.

The CHNGDUMP command only provides two ways to change the MAXSPACE value: by using the SET,SDUMP invocation as described here; or by using the RESET,SDUMP invocation that resets all initial SDUMP parameters, including setting the MAXSPACE value to 500M.

SDUMP,MSGTIME=yyyyy

Specifies how long message IEA793A appears on the console, where yyyyy is a number of minutes from 0 — 99999. The default is 99999. When the system deletes the message, it also deletes the captured dump.

You cannot delete this option with CHNGDUMP DEL. To change the value of MSGTIME, issue the CHNGDUMP command in the SET mode with a new value. If you change the MSGTIME value after the message IEA793A appears, the new value will be in affect immediately. If you set the MSGTIME value to 0, the system will not issue the message and it deletes the captured dump.

STRLIST= or STL=(STRNAME=strname...)

Used to include in the dump a list of coupling facility structures. Following are the structure-related keywords:

STRNAME= or STRNM=strname

Designates a particular coupling facility list or cache structure. *strname* is the name of the coupling facility structure to be included in the dump. Any dump options for this structure are replaced when you issue this command. If *strname* does not begin with a letter or is longer than 16 characters the system issues syntax error message IEE866I. If a structure does not exist, or the update fails for any reason, the system issues message IEE816I. You may include more than one STRNAME=strname within the parentheses, separated by commas.

CONNNAME= or CONNM=connname

When specified for a coupling facility cache structure, requests the user registry information for this user be included in the dump. *connname* is the name of a connected user. If the connected user represented by the connname does not exist, the dump will not contain user registry information.

CHNGDUMP Command

ACCESSTIME= or ACC={ENFORCE or ENF or NOLIMIT or NOLIM}

Indicates whether the dump time limit specified on the ACESSTIME parameter of the IXLCNN macro is in effect. When ACESSTIME=ENFORCE is specified, the system holds structure dump serialization no longer than the time interval specified on the IXLCNN macro. This is the default. If ACESSTIME=0 is specified on the IXLCNN macro and ACESSTIME=ENFORCE is specified on the dump request, the structure will not be included in the dump.

When ACESSTIME=NOLIMIT is specified, the dump time limit is not in effect and the system will hold structure dump serialization until processing is completed.

LOCKENTRIES or Locke

When specified for a coupling facility list structure, the system includes in the dump the lock table entries for the requested structure. Since lock table entries do not exist for coupling facility cache structures, this keyword is ignored when specified for a coupling facility cache structure.

USERCNTLS or UC

Requests that the user attach controls be included in the dump.

(list)

Represents a list of values, ranges of values, or values and ranges of values.

(start1-end1,value2,start3-end3, ...)

COCLASS= or COC=ALL or (list)

Specifies which cast-out classes are included in the dump. For each cast-out class, the cast-out class controls are dumped and the directory information for each of the entries within the requested cast-out classes are dumped (if SUMMARY is not specified).

COCLASS is valid only for a coupling facility cache structure. If specified for a coupling facility list structure, the structure is not included in the dump.

When COCLASS=ALL is specified, the cast-out class controls for all cast-out classes are dumped along with the directory information for all entries within the classes (if SUMMARY is not specified).

When COCLASS=(list) is specified, the cast-out class controls for (list) are dumped along with the directory information for the entries in the requested cast-out classes (if SUMMARY is not specified). The values specified in a range are the decimal cast-out class values in the range 0–65535. When a requested class does not exist, it is not dumped.

STGCLASS= or SC=ALL or (list)

Specifies which storage classes are included in the dump. For each storage class, the storage class controls are dumped and the directory information for each of the entries within the requested storage classes are dumped (if SUMMARY was not specified).

STGCLASS is valid only for a coupling facility cache structure. If specified for a coupling facility list structure, the structure will not be included in the dump.

When STGCLASS=ALL is specified, the storage class controls for all storage classes are dumped along with the directory information for all entries within the classes (if SUMMARY is not specified).

When STGCLASS=(*list*) is specified, the storage class controls for (*list*) are dumped along with the directory information for the entries in the requested storage classes (if SUMMARY is not specified). The values specified are the decimal storage class values, 0–255. When a requested class does not exist, it is not dumped.

LISTNUM= or LNUM=ALL or (*list*)

Specifies which lists are included in the dump. The list controls are dumped along with the entry controls for the entries on each requested list (if SUMMARY is not specified).

LISTNUM is only valid for a coupling facility list structure. If specified for a coupling facility cache structure, the structure is not included in the dump.

When LISTNUM=ALL is specified, the list controls for all lists in the coupling facility list structure are dumped along with the entry controls (if SUMMARY is not specified).

When LISTNUM=(*list*) is specified, the list controls for (*list*) are included in the dump along with the entry controls for those lists. The values specified are the decimal list values, 0–4294967295. The system ignores a zero in the case of LISTNUM. No error results. When a requested list does not exist, it is not dumped.

You may use the following keyword to further modify the STGCLASS, COCLASS and LISTNUM keywords:

ADJUNCT= or ADJ={CAPTURE or CAP or DIRECTIO or DIO}

Indicates that the adjunct data for each entry specified by the range is included in the dump. When you do not specify this keyword, or when adjunct data does not exist for this structure, the dump does not include the adjunct data.

ADJUNCT may not be specified with SUMMARY. If they are both specified, a syntax error is issued.

When ADJUNCT=CAPTURE is specified, the adjunct data is captured in the facility dump space along with the directory information while dumping serialization is held.

When ADJUNCT=DIRECTIO is specified, the adjunct data is written directly to the dump data set after the directory information is captured. The adjunct data is not captured in the structure dump table. The adjunct data may be changing as dumping proceeds.

ENTRYDATA= or EDATA={UNSERIALIZE or UNSER or SERIALIZE or SER}

Indicates that the entry data for each entry within the requested range is included in the dump. When this keyword is not specified or when entry data does not exist for the structure, entry data is not included in the dump.

ENTRYDATA may not be specified with SUMMARY. If they are both specified, a syntax error is issued.

When ENTRYDATA=UNSERIALIZE is specified, the entry data is dumped after structure dump serialization is released. The entry data may be changing relative to the entry controls that were captured while structure dump serialization was held.

CHNGDUMP Command

When ENTRYDATA=SERIALIZE is specified, the entry data is dumped while serialization is held. If ACESSTIME=ENFORCE is specified and the dump time limit expires before the entry data is written to the dump data set, the system continues to write the entry data to the dump data set even though serialization is not held.

SUMMARY or SUM

Indicates that a summary of the range of classes or lists is dumped. The directory information for the entries is excluded from the dump.

SUMMARY may not be specified with ADJUNCT or ENTRYDATA. If it is specified with either of these keywords, a syntax error is issued.

Notes:

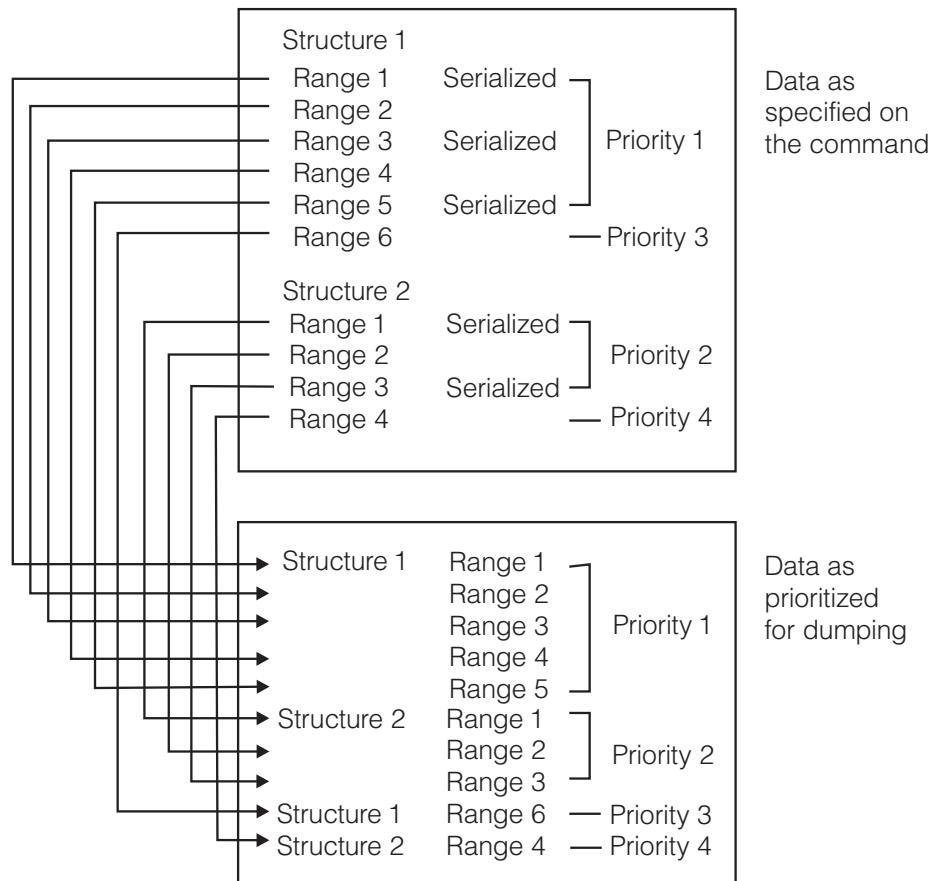
1. A syntax error is issued if STRNAME is not the first keyword.
2. If CONNAME and ACESSTIME are specified more than one time for a structure, the first CONNAME and the last ACESSTIME are used.
3. When a list number, a storage class, a cast-out class, or an entry is specified in the STRLIST more than once, it will be dumped more than once. An example of this is when STGCLASS=ALL is specified with COCLASS=ALL. All entries in the coupling facility cache structure are dumped twice. Once grouped by storage class and again grouped by cast-out class.
4. When neither LISTNUM, STGCLASS, or COCLASS is specified, no list or class controls are dumped and no entries are dumped.
5. If you request a large amount of dump data, the system may not be able completely to dump all the data. You can expect to successfully dump up to a maximum of 47 structures if you specify no more than a total of 6 ranges. If you must specify more than 6 ranges, you must specify fewer structures. For each structure less than 47 that you specify, you can specify another 10 ranges, as follows:

Number of Structures	Number of Ranges
47	6
46	16
45	26
44	36
:	:

If the system cannot dump all the data you requested, it prioritizes the data according to your specifications on the command in the following manner:

- a. The system will attempt to dump the first requested structure first.
 - 1) Within that structure, the system processes the LOCKENTRIES, USERCNTLS, COCLASS, STGCLASS, and LISTNUM parameters in the order that they are specified. COCLASS, STGCLASS, and LISTNUM may be specified more than once for a single structure.
 - 2) The system dumps requested serialized data before requested unserialized data starting with the first requested data in the structure and proceeding through the last data that was requested as *serialized*.

- b. The system then dumps the next-requested structure data starting with the first requested data in the structure and proceeding through the last data that was requested as *serialized*.
- c. The system continues in this manner until all *serialized* data in all requested structures has been prioritized for dumping.
- d. The system then dumps any remaining data that was requested as *unserialized* that may not have been dumped beginning with the first-requested structure.



- 6. The CONT parameter allows the operator to provide input to the CHNGDUMP command that spans more than one line of input. You can specify the CONT parameter after any comma within the STRLIST parameter list. If a line of input ends with a comma and any closing parentheses are missing, the system assumes the CONT parameter.

SYSABEND

Set the SYSABEND dump mode to ADD.

SYSABEND,NODUMP

Set the SYSABEND dump mode to NODUMP.

You cannot specify other parameters when specifying NODUMP. For example, you can specify CD SYSABEND,NODUMP, but not CD SYSABEND,PDATA=option,NODUMP.

SYSABEND,OVER or ADD

Set the SYSABEND dump mode to the specified mode.

CHNGDUMP Command

SYSABEND,SDATA=(option[,option]...)

Put the specified SDATA options in the SYSABEND system dump options list. See “Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP” on page 4-29 for SYSABEND options you can specify.

SYSABEND,PDATA=(option[,option]...)

Put the specified PDATA options in the SYSABEND system dump options list. See “Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP” on page 4-29 for SYSABEND options you can specify.

SYSUDUMP

Set the SYSUDUMP dump mode to ADD.

SYSUDUMP,NODUMP

Set the SYSUDUMP dump mode to NODUMP.

You cannot specify other parameters when specifying NODUMP. For example, you can specify CD SYSUDUMP,NODUMP, but not CD SYSUDUMP,PDATA=option,NODUMP.

SYSUDUMP,OVER or ADD

Set the SYSUDUMP dump mode to the specified mode.

SYSUDUMP,SDATA=(option[,option]...)

Put the specified SDATA options in the SYSUDUMP system dump options list. See “Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP” on page 4-29 for SYSUDUMP options you can specify.

SYSUDUMP,PDATA=(option[,option]...)

Put the specified PDATA options in the SYSUDUMP system dump options list. See “Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP” on page 4-29 for SYSUDUMP options you can specify.

SYSMDUMP

Set the SYSMDUMP dump mode to ADD.

SYSMDUMP,NODUMP

Set the SYSMDUMP dump mode to NODUMP.

You cannot specify other parameters when specifying NODUMP. For example, you can specify CD SYSMDUMP,NODUMP, but not CD SYSMDUMP=option,NODUMP.

SYSMDUMP,OVER or ADD

Set the SYSMDUMP dump mode to the specified mode.

SYSMDUMP=(option[,option]...)

Put the specified options in the SYSMDUMP system dump options list. See “Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP” on page 4-29 for SYSMDUMP options you can specify.

ABDUMP,TIMEENQ=yyyy

Sets the approximate number of seconds that ABDUMP processing waits for obtaining required resources. The value may be changed up to the maximum of 9999. If this command is used to change the value from the IBM default interval of 240 seconds, place the command into a COMMNDxx member of PARMLIB. That establishes the setting as the installation default when the system is IPLed. See *z/OS MVS System Messages, Vol 6 (GOS-IEA)* for the discussion in message IEA848I, DEADLOCK AVOIDED to find more information.

CHNGDUMP Command

Whenever the TIMEENQ value is changed, the new value takes effect immediately. A value of 0 does not force an SVC dump to occur, because the resources required might be immediately available. You can not use CHNGDUMP DEL to delete this option.

CMDS Command

Use the CMDS command to display executing and waiting MVS commands, to delete commands that are waiting for execution, or to cancel commands that are executing.

Syntax

The complete syntax for the CMDS command is:

```
CMDS {ABEND,CMD=cccccccc, ID=nnnn [,CLASS=classname] [,JOB=jobname] }  
      {DISPLAY|D[,CLASS=classname] [,CMD=cccccccc] [,ID=nnnn] [,JOB=jobname] }  
      {REMOVE|R[,CLASS=classname] [,CMD=cccccccc] [,ID=nnnn] [,JOB=jobname] }  
      {SHOW|S[,CLASS=classname] [,CMD=cccccccc] [,ID=nnnn] [,JOB=jobname] }
```

Notes:

1. The ABEND parameter requires that you specify the CMD= and ID= subparameters.
2. The REMOVE parameter requires that you specify at least one subparameter.
3. You may specify the optional subparameters of the default or specified parameter in any order.

Parameters

CMDS

- **ABEND** — abnormally end a command that is currently executing.

This parameter requires subparameters CMD= and ID=.

The system terminates, with ABEND code 422, reason code 00010301, the command that CMD=cccccccc and ID=nnnn identifies.

Use the ABEND option with extreme caution, being careful to avoid leaving the system in an inconsistent state. Use this parameter only as a last resort, such as when a command is hanging in execution.

The system issues message IEE064I in response to this command. It does not send any response message to the console that issued the abended command.

Attention: An ACTIVATE command may still be active as a task in IOSAS after the command task has been abended with a CMDS ABEND.

- **DISPLAY | D** — display the numbers and brief information about the commands that are currently executing and those that are waiting for execution.

The system issues message IEE062I in response to this command.

- **REMOVE | R** — remove commands that are waiting for execution, as specified by the subparameters. You cannot use this option to cancel any commands that are executing.

The REMOVE parameter requires you specify at least one of the keyword subparameters CLASS=, CMD=, ID=, or JOB=.

The system issues message IEE064I in response to this command, and sends message IEE065I to the console that issued the removed command.

- **SHOW | S** — display full information about the specific command(s) specified by the subparameters.

The system issues message IEE063I in response to this command.

The CMDS command uses the following keyword subparameters to limit the number of commands to show or remove:

- **CLASS= classname**

The command class in which the commands belong.

If you do not specify this parameter, commands in all classes will be displayed or removed.

The classes are:

1. Class M1 commands may be essential to clearing a backlog of Class M2 commands.
2. Class M2 commands are ordinary attached commands that run in the MASTER address space.
3. Class M3 is only for SEND commands executed in the MASTER address space.
4. Class C1 commands might be needed to clear a backlog of Class C2 commands.
5. Class C2 commands are ordinary attached commands that run in the CONSOLE address space.
6. Class C3 is only for the ROUTE command executed in the CONSOLE address space.

For a list of the commands by class, see “Command Flooding” on page 1-35.

- **CMD= command verb**

The name of the command, as displayed by the SHOW option. For example,

```
CMDS REMOVE,CMD=CONFIG
```

will delete all CONFIG commands that are waiting for execution.

You can use command abbreviations instead of full command names.

- **ID= number**

– The command’s sequence number, which appears in the output from a CMDS DISPLAY or CMDS SHOW command (messages IEE062I or IEE063I).

- **JOB= jobname of the command issuer, as displayed by the SHOW option. For example,**

```
CMDS REMOVE,JOB=JOB1111
```

will remove all commands issued by JOB1111.

If you specify more than one of the keyword parameters, the command must meet all specified criteria. For example,

```
CMDS REMOVE,CMD=VARY,JOB=JOB1111
```

will remove waiting VARY commands that job JOB1111 issued.

Note: Some commands remain active indefinitely, so the system will display them whenever CMDS is issued. For example, if any SLIP commands have been issued and SLIP traps are in effect, one SLIP command will be “executing” until all traps are deleted. This is also true for many SET commands, such as SET SLIP and SET MPF.

CONFIG Command

CONFIG Command

CONFIG ESTOR and CONFIG VF commands no longer supported:

The CONFIG ESTOR command is not supported on system at z/OS V1R7 and higher.

The CONFIG VF command is not supported on systems at z/OS V1R6 and higher.

The CONFIG CPUAD...,ONLINE,VFON | VFOFF command is not supported on systems at z/OS V1R6 and higher.

Use the CONFIG command to change or check the configuration of the system.

You can use the CONFIG command to change the online or offline status of available processors, storage amounts, storage ranges, central storage elements, and channel paths:

1. Directly
2. In response to a configuration display
3. With the options in a CONFIGxx parmlib member that you specify

Table 4-7 summarizes the information that the CONFIG command provides. Use it to access the pages on which you can find details about a particular use of the CONFIG command.

Table 4-7. Summary of the CONFIG Command

Command:	Topic:
CONFIG CHP CONFIG CPUAD CONFIG STORAGE	"Reconfiguring the System Directly" on page 4-47
CONFIG MEMBER	"Reconfiguring the System with a CONFIGxx Parmlib Member" on page 4-53
CONFIG OFFLINE CONFIG ONLINE	"Reconfiguring the System in Response to a Configuration Display" on page 4-54

The CONFIG command reconfigures (both logically and physically) available processors, central storage ranges, amounts, and elements, and channel paths.

Note: To configure an Integrated Cryptographic Facility (ICRF) online or offline, you have to configure online or offline the processor to which the ICRF is attached and, when configuring an ICRF online, Integrated Cryptographic Service Facility/MVS (ICSF/MVS) must be active. You can enter the CONFIG command only from a console with master authority.

Syntax

The syntax for each variation of the CONFIG command is shown immediately preceding its respective parameter list.

CONFIG or CF

Reconfiguring the System Directly

Use the CONFIG command to change the online or offline status of any of the following, directly, that is, without invoking a configuration display or a CONFIGxx parmlib member: available processors, and Integrated Cryptographic Facilities (ICRFs), attached to online processors, storage amounts, storage ranges, central storage elements, logical partitions, or channel paths.

See Chapter 2, “System Configuration,” on page 2-1 for more information about how to reconfigure the resources associated with a processor or a processor complex. See *PR/SM Planning Guide* for more information about logical partitions.

```
CF {{CPUAD|CPU}(x[,x]...) [,,{ONLINE|ON}|,{OFFLINE|OFF}] }

{{STORAGE|STOR}{(dddddddm) } [,,{ONLINE|ON}|,{OFFLINE|OFF}] }
  {(dddddmm-dddddm) }
  {(E=id) }

{CHP{(xx) } [,,{ONLINE|ON} [NOVARY] |,{OFFLINE|OFF} [,UNCOND] ] }
  {(aa-bb) }
  {(list) }
  {(ALL,id) }
```

The parameters are:

CPUAD or CPU

The system is to reconfigure one or more processors. The system is also to reconfigure one or more ICRFs attached to specified online processors.

(*x*[,*x*]...)

One or more processors identified by *x* in hexadecimal format.

ONLINE or ON

The system is to bring the specified processor(s) online. If necessary, the system synchronizes the processor’s TOD clocks. If Integrated Cryptographic Service Facility/MVS (ICSF/MVS) is active, the system brings online any ICRF attached to each processor.

OFFLINE or OFF

The system is to take offline the specified processor(s) and any ICRF attached to the processor(s).

STORAGE or STOR

The system is to reconfigure central storage, both logically and physically. Note that storage reconfiguration is not supported on all processors and that central storage reconfiguration in a PR/SM environment without enhanced dynamic storage reconfiguration must be specified by storage element ID. The starting and ending addresses of the central storage for which you want the status display.

dddddddmX

The amount of central storage to be reconfigured. Specify up to eight decimal digits followed by a multiplier (M-megabytes, G-gigabytes, T-terabytes, P-petabytes) for this amount. Check the configuration of your processor to see which size storage increments are supported. The value for *dddddddm* must be a multiple of the storage increment size (usually 2, 4, or 8), and cannot exceed 16383P.

CONFIG Command

Instead of specifying a decimal amount, you may specify a hexadecimal amount, with or without a multiplier, in the format X'xxxxxx'. For example:

- X'123456789A00000'
- X'123'M

You may use underscores in any hexadecimal specification for better clarity. Underscores in the specification are ignored during processing.

dddddddX-dddddddX

The starting and ending addresses of the central storage section to be reconfigured. Specify up to eight decimal digits followed by a multiplier (M-megabytes, G-gigabytes, T-terabytes, P-petabytes) for each address. The value for each *ddddddd* must be a multiple of the storage increment size (usually 2, 4, or 8), and cannot exceed 16383P. The starting and ending addresses must not be the same.

Instead of specifying the range using decimal numbers, you may specify it in hexadecimal, with or without a multiplier, in the format

X'xxxxxx'-X'xxxxxx'. For example:

- X'123456789A00000'-X'123456789B00000'
- X'123'M-X'124'M

You may use underscores in any hexadecimal specification for better clarity. Underscores in the specification are ignored during processing.

E=id

The storage element to be reconfigured, identified by the storage element *id*. Use this parameter only under the direction of a system programmer to take a storage element offline or online.

ONLINE or ON

The system is to bring the specified storage range or storage element online. The system rejects the command if you specify:

- An address higher than the storage limit set at system initialization
- An address or an element id for storage that is not available to the system

OFFLINE or OFF

The system is to take the specified storage range or storage element offline.

Notes:

1. There can be a delay between the time you enter CONFIG STOR ... OFFLINE and the time the system issues a message indicating the storage is offline. This delay occurs when there is activity in the specified storage; all activity in the storage must stop before the command can take effect. If the storage does not go offline within a short time, a message appears that lets you cancel the command.
2. When you issue CONFIG STOR ... OFFLINE without E=id, the system rejects the command if you specify storage that is either part of the hardware system area (HSA) or assigned permanently to the system. Generally, you can take non-preferred (reconfigurable) storage offline, but, you cannot take preferred (non-reconfigurable) storage offline.
3. When you issue CONFIG STOR ... OFFLINE with E=id, the system moves any storage associated with the HSA or permanently assigned to the system to another storage element. The system saves the addresses of the former storage and displays their address ranges.

4. In order to configure a range of storage online through the CONFIG STOR(xx-xx),ONLINE command, the subject storage must be in a storage element that is online. If a storage element is offline, the only way to bring online any storage within that element is to configure the entire element online, through CONFIG STOR(E=id),ONLINE.

CHP

The system is to reconfigure one or more channel paths.

Note that if you have systems running on a processor at the z990 level or higher, you can automatically reconfigure channel paths on and offline using the hardware management console (HMC) instead of issuing the CONFIG command. On a z990, you may have multiple logical channel subsystems, which means that if an ESCON card fails, you need to reconfigure 15 channels being used across 30 different partitions in each logical channel subsystem. Automatic CHPID reconfiguration lets you issue a reconfiguration request centrally from HMC, which in turn triggers the z/OS systems in each partition to issue the CONFIG command. Then only those partitions that cannot process the request or are not running on a z990 level processor or higher need to be individually reconfigured with the CONFIG command.

(xx)

A single channel path identified by *xx*. The channel path identifier may have a value from 0 to FF.

(aa-bb)

A range of channel paths. The starting and ending channel path identifiers may have values from 0 to FF.

(list)

One or more single channel paths, ranges of channel paths, or a combination of single channel paths and ranges of channel paths. ALL,id cannot be included in the list.

(ALL,id)

All of the channel paths associated with one side of a partitioned processor complex are to be placed online or offline, where *id* is the identifier (0 or 1) of the side. Use ALL,id only when your processor complex is one that can be partitioned (such as a 3090 Model 400 Processor Complex). Message IEE172I indicates that all channel paths on a side have been brought online or taken offline.

Note: If you configure a partitionable processor from single image to partitioned mode, and a tape mount is pending, the tape drive(s) might not start after you mount them. You can avoid the problem by mounting the tape before you issue the CONFIG CHP(ALL,id),OFFLINE command to perform the partitioning or, after partitioning, you can issue the VARY device,ONLINE command to start the tape drive(s).

ONLINE or ON

The system is to bring the specified channel path(s) online.

ONLINE,NOVARY

The system is to bring the specified channel paths online without bringing online the paths to the associated devices. Use this command when you want to configure online a channel path that does not currently have a device connected. Example 10 shows the operator commands and system responses.

CONFIG Command

OFFLINE or OFF

The system is to take the specified channel path(s) offline. The system rejects this command if it would remove the last path to a device that is:

- In use by a system function
- Online
- Allocated
- A TP device
- The only active console in the system
- A coupling facility.

To remove the last path to all other devices, use the CONFIG command without the UNCOND or FORCE parameters.

OFFLINE,UNCOND

The system is to take the specified channel path(s) offline, even if it is the last path to a device. The system rejects this command if it would remove the last path to a device that is:

- In use by a system function
- Allocated
- A TP device
- The only active console in the system
- A coupling facility in use by an active XES connection on the system from which the CONFIG command is issued. (Structures in the coupling facility can be in use, persistent, or have failed-persistent connectors.)

Use OFFLINE,UNCOND to remove the last path to an unallocated online device. You cannot do this by specifying OFFLINE alone.

OFFLINE,FORCE

CAUTION: FORCE is a very powerful option. Never specify FORCE unless you understand all its consequences for your system.

The system is to take the specified channel path(s) offline, even if it is the last path to a device. The system rejects this command if it would remove the last path to a device that is:

- The only active console in the system

The last path to all other devices listed in the OFFLINE,UNCOND option can be removed by the OFFLINE,FORCE option.

Responding to the FORCE Option

Message IEE100E lists any devices that are affected by the OFFLINE,FORCE options. The following message then asks you to confirm the FORCE option:

IEE131D REPLY 'CANCEL' OR 'CONTINUE'

Reply CANCEL to leave the channel path and devices online. Reply CONTINUE if you want to remove the channel path. After you reply CONTINUE, the following message appears:

IEE507D SHOULD ACTIVE DEVICES HAVE I/O TERMINATED?
REPLY NO OR YES

Reply NO to leave the affected devices online and allocated.

If you specify NO to message IEE507D, the channel path will NOT be configured offline.

Reply YES to have the system stop all I/O in progress on the affected devices, permanently reject all future I/O requests to the devices, and mark the affected devices pending-offline.

If you specify YES to message IEE507D, further system action depends on whether or not there are reserved devices on the channel path you want to take offline:

- If there are no reserved devices on the channel path, the system takes the channel path offline when it stops I/O on the devices.
- If there are reserved devices on the channel path for which there are no alternate paths, you get the following messages after you reply YES to message IEE507D:

```
IEE508E NO ALTERNATE PATHS TO RESERVED  
DEVICES ddd [,ddd]...  
IEE131D REPLY 'CANCEL' OR 'CONTINUE'
```

If you don't want to lose I/O on reserved devices, reply CANCEL to terminate the CONFIG command and leave the channel path and devices online. Reply CONTINUE to have the system continue to remove the channel path. After you reply CONTINUE, message IOS062E is issued. All the processors for this image enter a restartable disabled wait (WAIT062) state.

The IOS062E message asks you to stop all systems sharing the reserved devices so the system can reserve the devices again, if possible, through alternate paths. If the system cannot find alternate paths, it stops I/O in progress on the devices, rejects any future I/O requests to the devices as permanent I/O errors, and marks the devices pending-offline.

Once the system has been restarted from the wait state, recovery for the channel path will be started and completed. Then, message IOS201E will inform you to start the processors stopped when message IOS062E was issued.

Generally, when CONFIG CHP OFFLINE,FORCE causes the system to take a device offline, you can bring the device back online by bringing online a channel path that provides a path to the device. Once it is back online, the device is again available for allocation. However, if the device was reserved when the system took it offline with the channel path, to bring the device back online and make it again available for allocation, you must provide a path to the device with a CONFIG CHP command and issue a VARY device ONLINE command.

Responding to the FORCE Option for a Coupling Facility

Message IXL126I identifies the coupling facility that is affected by the OFFLINE,FORCE option. The following message then asks you to confirm the FORCE option:

```
IXL127A REPLY CANCEL OR  
CONTINUE
```

Reply CANCEL to leave the coupling facility online. Reply CONTINUE if you want to remove the coupling facility.

Example 1

To take processor 2 offline, enter:

CONFIG Command

```
cf cpu(2),offline
```

Example 2

To bring online a storage range from real addresses four to eight megabytes, enter:

```
cf stor(4m-8m),on
```

Example 3

To take storage element 0 offline, enter:

```
cf stor(e=0),offline
```

Example 4

To bring channel paths 4-9 and 12 online, enter:

```
cf chp(4-9,12),online
```

Example 5

To take channel paths 0-6 offline, even though one might be the last path to an unallocated online device, enter:

```
cf chp(00-06),offline,uncond
```

Example 6

To bring all channel paths associated with side 1 online, enter:

```
cf chp(all,1),online
```

Example 7

CHP(01) is associated with devices 223 and 224. To correct an error condition, CHP(01) was configured offline to the system. Problem analysis determined that device 224 has a hardware problem that cannot be immediately corrected. This example shows how to configure CHP(01) online without bringing the path to device 224 online.

To display status for devices 223 and 224 before configuring CHP(01) online, issue the following commands:

```
d m=dev(223)
```

```
d m=dev(224)
```

The output, which shows that the paths to the devices are not online and not operational, appears as follows:

```
IEE174I 09.05.00 DISPLAY M 197
DEVICE 0223  STATUS=OFFLINE
CHP          01
PATH ONLINE      N
CHP PHYSICALLY ONLINE N
PATH OPERATIONAL  N

IEE174I 09.05.30 DISPLAY M 200
DEVICE 0224  STATUS=OFFLINE
CHP          01
PATH ONLINE      N
CHP PHYSICALLY ONLINE N
PATH OPERATIONAL  N
```

To configure channel path 01 online without bringing online the paths to devices 223 and 224, issue the following command:

```
cf chp(1),online,novary
```

The system issues the following messages to indicate that not all paths were brought online:

```
IEE754I NOT ALL PATHS BROUGHT ONLINE WITH CHP(01)
IEE502I CHP(1),ONLINE
IEE712I CONFIG PROCESSING COMPLETE
```

To display the status of each device after configuring the channel path online, issue the following commands:

```
d m=dev(223)
d m=dev(224)
```

The output, which shows that the paths to the devices are not online but are operational, appears as follows:

```
IEE174I 09.05.40 DISPLAY M 200
DEVICE 0223 STATUS=OFFLINE
CHP 01
PATH ONLINE N
CHP PHYSICALLY ONLINE Y
PATH OPERATIONAL Y

IEE174I 09.05.50 DISPLAY M 200
DEVICE 0224 STATUS=OFFLINE
CHP 01
PATH ONLINE N
CHP PHYSICALLY ONLINE Y
PATH OPERATIONAL Y
```

To vary the path online for device 223, issue the following command:

```
vary path(223,01),online
```

To display the status of devices 223 and 224 after varying the channel path online, issue the following commands:

```
d m=dev(223)
d m=dev(224)
```

The output, which shows that only the path to device 223 is online and operational, appears as follows:

```
IEE174I 09.05.55 DISPLAY M 200
DEVICE 0223 STATUS=ONLINE
CHP 01
PATH ONLINE Y
CHP PHYSICALLY ONLINE Y
PATH OPERATIONAL Y

IEE174I 09.05.59 DISPLAY M 200
DEVICE 0224 STATUS=OFFLINE
CHP 01
PATH ONLINE N
CHP PHYSICALLY ONLINE Y
PATH OPERATIONAL Y
```

Reconfiguring the System with a CONFIGxx Parmlib Member

Use the CONFIG MEMBER command when you want the system to use options in a CONFIGxx parmlib member to change the online or offline status of available processors, storage sections, and channel paths. See *z/OS MVS Initialization and*

CONFIG Command

Tuning Reference for detailed information on the syntax and contents of the CONFIGxx parmlib member.

```
CF MEMBER[(member-id)]
```

The parameters are:

MEMBER

The system is to use a CONFIGxx parmlib member to reconfigure available processors, storage sections, and channel paths.

member-id

The identifier (xx) of the CONFIGxx member you want the system to use to reconfigure the available processors, storage sections, and channel paths. This identifier may be any two alphanumeric characters. If you do not specify *member-id*, the default member is CONFIG00.

Example

To reconfigure available processors, central storage, and channel paths in the system to match the options in the CONFIGT3 parmlib member, enter:

```
cf member(t3)
```

Reconfiguring the System in Response to a Configuration Display

Use the following form of the CONFIG command before making any changes to the system configuration, or to display the processors, Integrated Cryptographic Features (ICRFs) attached to the processors, total amount of storage, channel paths available to the system, and the online or offline status of each channel path. (To obtain more information on storage, use the DISPLAY M command.)

In response to this command, the system issues messages IEE521I and IEE522D. Message IEE521I displays the reconfigurable resources available to the system, including processors, ICRFs attached to the processors, total amount of central storage, central storage elements, and channel paths. If your processor complex is partitioned, message IEE521I contains this information for one side. Respond to message IEE522D with the processors, total amount of central storage, as well as channel paths you want to bring online or take offline.

If the system is unable to display the resources available to the system, message IEE521I indicates that the command was unsuccessful.

```
CF {ONLINE|ON } [,L={a|name|name-a}]{OFFLINE|OFF}
```

ONLINE or ON

The system is to display the system configuration so that you can decide which processors, ICRFs attached to the processors, central storage elements, and channel paths you want to bring online. The system brings online the processors, ICRFs attached to the processors, storage elements, and channel paths you specify in response to message IEE522D. If an ICRF is attached to a processor that is being brought online and the Integrated Cryptographic Service Facility/MVS (ICSF/MVS) is active, the system brings the ICRF online.

OFFLINE or OFF

The system is to display the system configuration so that you can decide which processors, ICRFs attached to the processors, central storage elements, and channel paths you want to take offline. The system takes offline the processors, ICRFs attached to the processors, storage elements, and channel paths you specify in response to message IEE522D. If an ICRF is attached to a processor that is being taken offline and the ICSF/MVS is active, the system takes the ICRF offline.

L=a, name or name-a

The display area where the system is to display the system configuration. You can specify the display area (a), console name (name), or both (name-a). If you do not specify this option, the system displays the system configuration in the first available display or message area of the console on which you entered the command.

Example 1

To check the channel paths available to the system before bringing any online, enter:

cf online

When message IEE522D appears after the configuration display message, IEE521I, reply with the channel paths you want to bring online.

Example 2

To check the available processors, central storage elements, and channel paths in the system and the online or offline status of each, enter:

cf offline or cf online

When message IEE522D appears after the configuration display message, IEE521I, reply with NONE.

CONTROL Command

Use the CONTROL command to control the screen display of MCS and SMCS consoles. Table 4-8 summarizes the information that the CONTROL command provides. Use it to access the pages on which you can find details about a particular use of the CONTROL command.

Table 4-8. Summary of the CONTROL Command

Command:	Topic:
CONTROL A	"Changing Out Of Line Display Area Specifications" on page 4-57
CONTROL C	"Deleting Retained Action Messages" on page 4-58
CONTROL C,D	"Halting the Printing or the Display of a Status Display" on page 4-60
CONTROL D	"Controlling Displays in Areas" on page 4-60
CONTROL E	"Removing Information From the Screen" on page 4-62
CONTROL M,AMRF	"Activating, Deactivating, or Displaying the Status of the Action Message Retention Facility" on page 4-63
CONTROL M,APPLID	"Setting the APPLID of the System" on page 4-68
CONTROL M,GENERIC	"Setting or Turning Off the VTAM Generic Resource Name for SMCS" on page 4-69
CONTROL M,LOGLIM	"Changing or Displaying the Number of Allowed WTL SYSLOG Buffers" on page 4-64
CONTROL M,MLIM	"Changing or Displaying the Number of Allowed WTO and WTOR Message Buffers" on page 4-64
CONTROL M,ROUTTIME	"Changing the Time the System Waits for ROUTE Command Responses" on page 4-65
CONTROL M,RMAX	"Increasing the Maximum Number of Reply IDs" on page 4-66
CONTROL M,UEXIT	"Changing or Displaying the Status of WTO Installation Exit IEAVMXIT" on page 4-67
CONTROL N,PFK	"Changing a PFK Definition" on page 4-70
CONTROL Q	"Deleting Message Queues" on page 4-72
CONTROL S	"Changing or Displaying Message Deletion and Format Specifications" on page 4-72
CONTROL T	"Changing or Displaying Time Intervals for Dynamic Displays" on page 4-76
CONTROL V,USE	"Changing the Operating Mode of a Console" on page 4-77
CONTROL V,LEVEL	"Selecting the Message Levels for a Console" on page 4-78

The following CONTROL commands have no effect on extended MCS consoles or on system consoles, and are not valid for managing these consoles:

- K A
- K C,D
- K D
- K E
- K N,PFK
- K Q
- K S
 - CON=

- SEG=
- DEL=
- RNUM=
- RTME=
- K T
- K V,USE

The system does not support the following variants of the CONTROL command at z/OS V1R7 and above:

- K T
- K D,H
- K D,U

Many of the functions of the CONTROL command are controlled at IPL by parameters in the CONSOLxx parmlib member. Accompanying the descriptions of some operands on the CONTROL command are the corresponding parameters in CONSOLxx. If you need more information about the parameters in CONSOLxx, see *z/OS MVS Planning: Operations* and *z/OS MVS Initialization and Tuning Reference*. See “Defining and Changing Console Characteristics” on page 3-20 for more information about using the CONTROL command.

Scope in a Sysplex

The following table describes the conditions under which the CONTROL command has sysplex scope. See “Using Commands That Have Sysplex Scope” on page 1-11 for an explanation of sysplex scope. If a command has **All** under “Conditions”, then the command has sysplex scope under all circumstances and for all variations.

Table 4-9. Sysplex Scope for CONTROL Command

Command	Conditions
CONTROL C,A	All
CONTROL C,D	Has sysplex scope only when you specify L=.
CONTROL M	Has sysplex scope only when you do not specify MLIM, UEXIT, LOGLIM, or APPLID.
CONTROL other	Other parameters of CONTROL have sysplex scope only when you specify L=.

Syntax

The syntax for each of the many variations of the CONTROL command is shown immediately preceding its respective parameter list.

CONTROL or K

Changing Out Of Line Display Area Specifications

Use the CONTROL A command to change, remove, or examine out of line display area specifications for any MCS or SMCS console in the system.

The AREA parameter on the CONSOLE statement in the CONSOLxx parmlib member controls the display area specifications at IPL. *z/OS MVS Planning: Operations* shows the maximum message area sizes for all devices that MVS supports as consoles and the defaults for the AREA parameter.

CONTROL Command

```
K A[,nn[,nn]...][,L={name }]  
|,NONE  
|,REF
```

- A The display area specifications are to be altered or referenced.

nn[,nn]...

The number of message lines in each display area. The first number (*nn*) defines the bottom area of the screen; additional numbers (,*nn*) define areas working toward the top of the screen. The minimum number of lines in one display area is 4, the maximum is 99. The total number of all specified lines cannot exceed the number of lines in the message area of the screen. The maximum number of display areas that you can define is 11.

NONE

All out-of-line display area specifications will be removed for the specified console.

REF

Displays, in the command input area, the size of out-of-line display areas for the specified console. For example, if you enter K A,REF in the command input area, and the size of the display area is 14, the system displays K A,14 in the command input area.

L= name

The MCS or SMCS console whose display area is to be changed or referenced. If this operand is omitted, the console on which K A is entered is assumed.

Example

To define two display areas of 4 and 6 lines respectively, enter:

K A,4,6

Note: The CONTROL A command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

Deleting Retained Action Messages

Use the CONTROL C command to delete the outstanding action messages that the action message retention facility (AMRF) has retained.

```
K C,{A|I|E|CE},{id}id-id[,id|id-id]...
```

The parameters are:

- C The system is to delete one or more action messages that AMRF has retained.

A The system is to delete one or more outstanding retained action messages in any of the following categories:

- immediate action (descriptor code 1 or 2)
- eventual action (descriptor code 3)
- critical eventual action (descriptor code 11).

These messages are identified by number in response to the DISPLAY R,LIST command.

- I** The system is to delete one or more outstanding immediate action messages (descriptor code 1 or 2). These messages are identified by number in response to the DISPLAY R,I command.
- E** The system is to delete one or more outstanding eventual action messages (descriptor code 3). These messages are identified by number in response to the DISPLAY R,E command.

CE

The system is to delete one or more outstanding critical eventual action messages (descriptor code 11). These messages are identified by number in response to the DISPLAY R,CE command.

- id** The one-to-ten-digit decimal message identification number of the message to be deleted. This number is listed in response to the DISPLAY R command (message IEE112I).

id-id

The one-to-ten-digit decimal message identification numbers of the beginning and end of a range of messages to be deleted. The ending number must be greater than or equal to the beginning number.

When a range of numbers is specified, all retained immediate action, eventual action and/or critical eventual action messages with identification numbers in the specified range are deleted.

Example 1

To delete a range of immediate action, eventual action, and critical eventual action messages that have been retained with identification numbers from 0 to 110, enter:
CONTROL C,A,0-110

Example 2

Assume you have completed the requested action for three eventual action messages, but the messages remain marked as outstanding. Use the DISPLAY R,E command to get the identification numbers of the three messages (id1, id2, and id3) and then use K C,E to delete the three messages.

DISPLAY R,E K C,E,id1,id2,id3

Example 3

Assume you have performed the requested immediate action, but the message is still marked as outstanding. Use the DISPLAY R,I command to obtain the message identification number and then issue CONTROL C,I to delete the message.

DISPLAY R,I
CONTROL C,I,id

Notes:

1. Specifying a large range of message identification numbers (more than 1000) can result in system resources being held and performance being impaired.
2. When you delete retained messages, the requests relating to them may still be outstanding.
3. When you specify a range, certain conditions are not flagged as errors that would be errors if individual identification numbers were specified. For example,

CONTROL Command

nonexistent message identification numbers that fall within the range do not cause an error message. If you specify a nonexistent identification number by itself, you receive an error message.

Halting the Printing or the Display of a Status Display

Use the CONTROL C,D command to halt the printing or display of the status:

- On a printer console that is not the hardcopy medium
- On a display console that does not have display areas

The CONTROL C,D command must be entered while the system is displaying or printing the display.

```
K C,D,id[,L={a|name|name-a}]
```

The parameters are:

C,D

The inline display, indicated by the id operand, is to be stopped. This command is only valid on an MCS or SMCS console.

id The three-digit identification number of the status display you want to stop.
The identification number appears in the first line of the display.

L=a, name, or name-a

The name of the active MCS or SMCS (printer or display) console where the status display is to be stopped. If you do not specify the L= operand, then the K C,D,id command applies to the console from which the command is issued.

Example 1

To stop the status display, identification number 121, which is in progress in the general message area of console CON21, enter:

```
k c,d,121,L=CON21
```

Note: The CONTROL C,D command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

Controlling Displays in Areas

CONTROL D,H and CONTROL D,U OPTIONS

The CONTROL D,H and CONTROL D,U options of this command are not supported on systems at z/OS V1R7 and higher.

Use the CONTROL D command to control displays, message numbering, and the PFK display line.

```
K D[,N[,HOLD]
      |{,F|,H|,U}|[,L={a|name-a}]
      |,PFK]
```

The parameters are:

- D** Control what or how information is to be displayed on the screen. This command is only valid for MCS and SMCS display consoles.
- N** Messages on the screen are to be consecutively numbered. The operator uses these numbers as references to delete messages from the screen using the K E,nn command. (For very large screen sizes the K D,N command will only number the first 99 rows on the screen. All subsequent lines will contain AA in the number field.) The numbers are removed from the screen when the operator deletes a message or performs a cancel action. (The K E,nn command cannot delete messages after line 99 on the screen.) This option is not valid when:
 - The console is in wrap mode; the system issues message IEE290I.
 - The console is in roll or roll-deletable mode; the system issues message IEE158I.

HOLD

In conjunction with the N operand, HOLD specifies that the system display consecutive numbers for each message on the screen and renumber messages after each message deletion that the operator performs.

- F** The next frame of a status display is to be displayed. When you do not specify an area, the oldest area on the screen is scrolled. This option is not valid when there is no status display on the console; the system issues message IEE158I.
- H** The updating of a dynamic status display is to be suppressed. This option is not valid when there is no dynamic status display on the console; the system issues message IEE158I. **The H option is not supported on systems at z/OS V1R7 and higher.**
- U** The updating of a dynamic status display is to be resumed. This option is not valid when there is no dynamic status display on the console; the system issues message IEE158I. **The U option is not supported on systems at z/OS V1R7 and higher.**

L=a or name-a

Specifies the display area (a), console name (name), or both (name-a) where the display will appear.

The L= operand only applies to the F, H, and U parameters.

PFK

Specifies that the numbers of the program function keys (PFKs) designated for command entry are to be displayed in the PFK display line. This operand applies only to display consoles that have the PFK function and support the PFK display line (for example, 3277-2).

Example

To display the next frame of a status display, enter:

k d,f

Note: The CONTROL D command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

CONTROL Command

Removing Information From the Screen

Use the CONTROL E command to remove various types of messages from the screen or to cause message numbers to be deleted.

```
K E[,nn]          ]
      ,nn,nn
      ,SEG
      ,F
      ,N
      ,PFK
      ,D[,L={a|name|name-a}]
```

The parameters are:

E Messages are to be removed from the screen.

nn The system is to remove a single (*nn*) message or a range (*nn,nn*) of deletable messages from the screen. The value of *nn* must be a decimal number from 01 to 99. The K E,nn command cannot delete messages after line 99 on very large screen sizes. A deletable message is any message that is either a WTOR or a WTO message issued without descriptor code 1, 2, 3, or 11.

Note: Do not use this command to try to remove a range of non-deletable messages; you can remove only one non-deletable message at a time.

SEG

Deletable messages in the predefined message segment are to be removed from the screen. A deletable message is any message that is either a WTOR or a WTO message issued with descriptor code 1,2,3, or 11.

F All flagged messages (messages marked with a vertical or horizontal bar in position 3) are to be removed from the screen.

N Removes the numbers preceding the messages displayed on the screen. The messages on the screen were consecutively numbered by the K D command.

D Specifies that a status display is to be deleted.

L=a, name, or name-a

For the D operand, specifies where the specified action is to take place. You can specify the display area by area (*a*), console name (*name*), or both (*name-a*). When this keyword is not specified, the oldest area on the screen is deleted.

PFK

Specifies that the numbers are to be erased from the program function key (PFK) display line. This operand applies only to display consoles that have the PFK function and support the PFK display line (for example, 3277-2); the system issues message IEE158I for all other consoles.

Example 1

To delete the message at line 10, which appears on the screen as follows:

```
10 IEE334I HALT EOD SUCCESSFUL
```

enter:

K E,10

Example 2

To delete the non-action messages from a segment of messages, enter:

K E,SEG or K E

Example 3

To delete the non-action messages on lines 4-10, enter:

K E,4,10

Example 4

To delete all flagged messages (messages marked with a vertical or horizontal bar in position 3), enter:

K E,F

Note: The CONTROL E command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

Activating, Deactivating, or Displaying the Status of the Action Message Retention Facility

Use the CONTROL M,AMRF command to change or display the status of the action message retention facility. You can enter the command from a console with master authority.

The AMRF parameter on the INIT statement in the CONSOLOxx parmlib member controls whether or not the system starts the action message retention facility at IPL. If the AMRF parameter is not coded, the action message retention facility is active. Use the CONTROL M command to stop or restart the facility once the system is active.

The syntax of the command is as follows:

```
K M[,AMRF={Y|N}]
  ,REF
```

The parameters are:

M Indicates message processing by the message retention facility.

AMRF=

The status of the action message retention facility is to be modified for all systems in the sysplex.

Y The action message retention facility is to become active.

N The action message retention facility is to be deactivated.

REF

Displays the current values of all the operands on the K M command.

Example

CONTROL Command

To activate the action message retention facility, enter:
K M,AMRF=Y

Changing or Displaying the Number of Allowed WTL SYSLOG Buffers

Use the CONTROL M,LOGLIM command to change or display the number of allowed WTL (write-to-log) SYSLOG buffers. You can enter the command from consoles with master authority.

The LOGLIM parameter on the INIT statement in the CONSOLxx parmlib member controls the number of WTL buffers. If the LOGLIM parameter is not coded, the number of WTL buffers is 1000.

The syntax for the command is:

```
K M[,LOGLIM={nnnnnn|0}]\n      |,REF
```

The parameters are:

M Indicates message processing by the message retention facility.

LOGLIM=nnnnnn

The maximum number of outstanding WTL requests that the system can hold in buffers on the system, where *nnnnnn* can be a decimal number from 1000 to 999999. In a sysplex, the WTL SYSLOG buffers hold messages that the system has not yet presented to the hard-copy medium in the sysplex.

Note: When you set LOGLIM=999999 you allocate over 100 megabytes of CSA storage for WTL SYSLOG buffer storage. Be careful that WTL SYSLOG buffer storage does not hamper your system's performance.

LOGLIM=0

All outstanding WTL buffer storage is freed by the system log task.

Note: Use this command value only at the direction of the system programmer. It results in the potential loss of messages sent to hard-copy.

REF

Displays the current values of all the operands on the K M command. Unless you specifically change it, the LOGLIM value that the system displays is the value that was specified on the INIT statement in CONSOLxx parmlib member.

Example

To allow 4000 WTL message buffers, enter:
K M,LOGLIM=4000

Changing or Displaying the Number of Allowed WTO and WTOR Message Buffers

Use the CONTROL M,MLIM command to change or display the number of allowed WTO (write-to-operator) or WTOR (write-to-operator-with-reply) message buffers. You can enter the command from consoles with master authority.

The MLIM and RLIM parameters on the INIT statement in the CONSOLxx parmlib member control the number of WTO and WTOR buffers. If the MLIM or RLIM parameters are not coded, the number of WTO buffers is 1500, and the number of WTOR buffers is 10.

The syntax for the command is:

```
K M[,REF|[,MLIM=nnnn][,RLIM=mmmm]]
```

The parameters are:

M Indicates message processing by the message retention facility.

MLIM=nnnn

The maximum number of WTO message buffers to be allowed in the system, where *nnnn* can be a decimal number from 20 to 9999. The WTO buffers hold the WTO messages that the system has not yet displayed at the eligible MCS and SMCS consoles in the sysplex.

RLIM=mmmm

The maximum number of outstanding WTOR messages that the system or sysplex can hold in buffers, where *mmmm* can be a decimal number from 5 to 9999. Each WTOR buffer holds a WTOR message that the system or sysplex has displayed and has not received a response to. The maximum upper limit is set by the RMAX keyword on the DEFAULT statement in CONSOLExx.

REF

Displays the current values of all the operands on the K M command.

Example

Assuming RMAX is set to 9999: To allow the maximum number of WTO message buffers and WTOR message buffers, enter:

```
K M,MLIM=9999,RLIM=9999
```

Changing the Time the System Waits for ROUTE Command Responses

Use the CONTROL M,ROUTTIME command to display or change the maximum amount of time the ROUTE *ALL, ROUTE *systemgroupname*, or ROUTE *OTHER command waits for a response from each system in the sysplex before aggregating the responses. ROUTTIME applies to any ROUTE command with the *ALL or *systemgroupname* operand when issued from any system in a sysplex.

The syntax of the command is as follows:

```
K M[,ROUTTIME=nnn ]
|,REF
```

The parameters are:

ROUTTIME=nnn

Dynamically changes the maximum number of seconds the ROUTE *ALL or ROUTE *systemgroupname* command waits for command responses from each

CONTROL Command

system before aggregating the responses. (If not specified in CONSOLxx, the IBM-supplied default value is 30 seconds.) The *nnn* value is a decimal number from 0-999.

If *nnn* is zero, command responses are not aggregated. The change applies across the sysplex, and affects ROUTE commands issued after the CONTROL command is processed.

If the timeout value (T= operand) is specified on a ROUTE *ALL or ROUTE *systemgroupname* command, the value of the T= operand overrides the value then in effect on the system.

REF

Displays the current values of all the operands on the K M command.

Example 1

To change the maximum amount of time ROUTE *ALL or ROUTE *systemgroupname* waits for command responses to 45 seconds, enter:

K M,ROUTTIME=45

Example 2

To display the maximum amount of time ROUTE *ALL or ROUTE *systemgroupname* waits for command responses, enter:

K M

Increasing the Maximum Number of Reply IDs

Use the CONTROL M,RMAX command to display or dynamically increase the maximum number of reply IDs.

The syntax of the command is as follows:

```
K M[,RMAX=nnnn ]  
|,REF
```

The parameter is:

RMAX=nnnn

Dynamically increase the maximum number of reply IDs, where *nnnn* is a decimal number from 99 to 9999.

Note: The value for RMAX also determines the size of the reply ID displayed in the message text. For example, specifying an RMAX of 999 means that all WTOR messages have a 3-character reply ID.

You can increase the value of RMAX only in a system running in local mode or in a sysplex whose couple data set supports more than eight systems.

The new value of RMAX must be greater than the previous value of RMAX.

REF

Displays the current values of all the operands on the K M command.

Example

To increase the maximum number of reply IDs to 200, enter:

K M,RMAX=200

Changing or Displaying the Status of WTO Installation Exit IEAVMXIT

Use the CONTROL M,UEXIT command to change or display the status of the WTO installation exit IEAVMXIT. This exit receives control when the system issues a WTO message unless your installation names another WTO installation exit routine for the message. To learn what messages are currently processed by IEAVMXIT, issue the DISPLAY MPF command.

The UEXIT parameter on the INIT statement in the CONSOLxx parmlib member controls whether IEAVMXIT is active at IPL. If the UEXIT parameter is not coded, IEAVMXIT will be activated, if it is installed. If IEAVMXIT is not installed, the system will IPL with UEXIT=N.

The syntax of the command is:

```
K M[,UEXIT={Y|N}]
  |,REF
```

The parameters are:

UEXIT=

The status of the general WTO installation exit IEAVMXIT is to be changed.

Y The general user exit routine IEAVMXIT is to become active. If IEAVMXIT is already active and you want a new copy, deactivate IEAVMXIT, refresh the library lookaside (LLA), and then reactivate IEAVMXIT.

N The general user exit routine IEAVMXIT is to become inactive.

REF

Displays the current values of all the operands on the K M command.

Example 1

To activate the general WTO installation exit routine IEAVMXIT, enter:

K M,UEXIT=Y

Example 2

To load a new copy of the general WTO installation-exit routine IEAVMXIT, first enter:

K M,UEXIT=N

to deactivate the current copy. Then enter:

MODIFY LLA,REFRESH

After you receive notification that the library lookaside (LLA) is refreshed, enter:

K M,UEXIT=Y

to activate the new copy.

CONTROL Command

Displaying the SMCS APPLID of the current system and VTAM generic resource name for SMCS

Use the CONTROL M,REF command to display the SMCS APPLID of the current system and VTAM generic resource name for SMCS.

If either of these values has been changed by a prior CONTROL M command, but SMCS has not yet been recycled using the VARY NET,INACT and VARY NET,ACT commands to deactivate and restart the SMCS application, CONTROL M will show the new APPLID and GENERIC, even though SMCS will be using the old APPLID and GENERIC. The DISPLAY CONSOLES,SMCS command can be used to display the APPLID and GENERIC in use on each system in the sysplex, as well as the APPLID and GENERIC set by the CONTROL M command.

If the system is in XCFLOCAL or MONOPLEX mode, the GENERIC keyword will not be displayed. If the system does not have an APPLID in effect, the APPLID keyword will not be displayed.

The syntax of the command is:

```
K M[,REF]
```

The parameters are:

REF

Displays the current values of all the operands on the K M command.

Setting the APPLID of the System

Use the CONTROL M,APPLID command to set the APPLID of the system where it is issued. If there is not an APPLID in effect on the system, either because an APPLID was not specified in CONSOLxx, or because the APPLID was invalid or in use by another system during this system's IPL, this command is rejected.

For the new APPLID to take effect, after issuing the CONTROL M command to change it, the VARY NET,INACT,ID=oldapplid[,I or ,F] command must be issued to deactivate SMCS, followed by the VARY NET,ACT,ID=newapplid command to activate SMCS using the new APPLID. This is sometimes referred to as "recycling the APPLID" or "recycling SMCS". Until SMCS is recycled, the old APPLID value is still in use. Message IEE821E is issued to reflect the need to recycle SMCS.

The syntax of the command is:

```
K M,APPLID=aaaaaaaa
```

The parameters are:

APPLID=aaaaaaaa

Sets the APPLID of the system where it is issued.

Value Range: Is from 2 to 8 characters. The first character must begin with the letters A through Z or with a #, \$, or @; the remaining characters can be A through Z, 0 through 9, or #, \$, or @.

Example:

Systems SY1 and SY2 are in a sysplex. System SY1 is using APPLID SMCS1, and SY2 is using APPLID SMCS2. To change SY1's APPLID to SMCSA, enter:

```
K M,APPLID=SMCSA
```

to change the APPLID. However, SY1 will continue to use SMCS1 as its APPLID until SMCS is recycled with the following commands:

```
VARY NET,INACT,ID=SMCS1,I  
VARY NET,ACT,ID=SMCSA
```

Setting or Turning Off the VTAM Generic Resource Name for SMCS

Use the CONTROL M,GENERIC command to set or turn off the VTAM generic resource name for SMCS in the sysplex.

If the system is in XCFLOCAL or MONOPLEX mode, this command is rejected.

For the updated GENERIC value to take effect, after issuing the CONTROL M command to change it, the VARY NET,INACT,ID=applid[,I or ,F] command must be issued to deactivate SMCS, followed by the VARY NET,ACT,ID=applid command to reactivate SMCS using the new GENERIC value. This is sometimes referred to as "recycling the APPLID" or "recycling SMCS". Each SMCS application in the sysplex will continue to use the old GENERIC value until it is recycled. It is not necessary to recycle all of the SMCS applications at the same time, however, this may result in some systems using the old value of GENERIC and others using the new value of GENERIC until all SMCS applications in the sysplex are recycled.

The syntax of the command is:

```
K M[,GENERIC={aaaaaaaa}]  
| {*NONE*}
```

The parameters are:

GENERIC=

aaaaaaaa

Sets the VTAM generic resource name for SMCS in the sysplex.

NONE

Turns off the VTAM generic resource name for SMCS in the sysplex.

Example:

Systems SY1 and SY2 are in a sysplex. System SY1 is using APPLID SMCS1, and SY2 is using APPLID SMCS2, and the current GENERIC is SMCSX. To change the GENERIC to ANYSMCS, on either SY1 or SY2, enter:

```
K M,GENERIC=ANYSMCS
```

SY1 and SY2 will continue to use SMCSX as the GENERIC until SMCS is recycled on each system. To recycle SMCS on SY1, issue the following commands:

```
VARY NET,INACT,ID=SMCS1,I  
VARY NET,ACT,ID=SMCS1
```

CONTROL Command

To recycle SMCS on SY2, issue the following commands:

```
VARY NET,INACT,ID=SMCS2,I  
VARY NET,ACT,ID=SMCS2
```

Changing a PFK Definition

Use the CONTROL N,PFK command to change the definition of a PFK on a particular console or to assign a PFK table to a particular console. The set of commands associated with the PFKs on your console resides in a PFK table in a PFKTABxx parmlib member.

A PFK command that you assign to a PFK by using the CONTROL N,PFK command is not associated with the PFK when you bring the console online again. To have a command associated with a PFK when you bring a console online, it must be defined in the appropriate PFK table in PFKTABxx. See “Defining PFKs Using the CONTROL Command” on page 3-36.

Use the following form of the CONTROL command to define commands for program function keys (PFKs) or assign a PFK table.

```
K N,PFK={(nn1{,CMD='text[;text]...'})[,CON={Y|N}] }  
    { ,KEY=nn2[,nn2]... }  
    {  
        {nnnnnnnn[,L=name]}  
    }
```

The parameters are:

N,PFK

A PFK command definition is to be altered.

nn1

The number of the PFK being defined. The *nn1* value must be the number of a PFK designated for command entry at system installation

CMD

The text of one or more commands is to be associated with PFK *nn1*.

‘text[;text]...’

The text of the operator's commands to be associated with PFK *nn1*. Up to 110 characters can be included within the quotes. If more than one command is to be associated with a PFK, the commands must be separated by a semicolon. Do not put a semicolon after the last command. Text characters can be entered in upper or lower case; the system converts all characters to uppercase. A command that must be entered lowercase, such as a reply to a WTOR, cannot be entered using the PFK command entry function.

Note: Text characters should not contain sensitive or secure data (such as passwords).

KEY

The commands associated with other PFKs are to be associated with *nn1*.

nn2[,nn2]...

The number(s) of the PFK whose commands are to be associated with PFK *nn1*. Up to 54 key numbers (numbers can be repeated) can be included in the list. Separate key numbers with a comma.

Note: You cannot nest the lists of keys. That is, a PFK defined as a list of PFKs cannot be included in a list of keys assigned to another PFK. For example, if PFK 5 is associated with a list of keys (such as KEY=3,4), and you attempt to associate PFK 6 with a list of keys that includes PFK 5 (such as KEY=1,2,5), the system rejects the request.

CON

Specifies whether conversational mode of command entry is in effect.

Y Conversational mode of command entry is to be in effect.

N Conversational mode of command entry is not to be in effect (non-conversational mode of command entry is to be in effect). If CON is not specified, CON=N is assumed.

nnnnnnnn

The name of the PFK table that contains the commands that define the PFKs for a console.

L=name

The console whose PFKs are to be defined by the PFK table you specify. The issuing console is the default.

Example 1

To associate a START GTF command with PFK 5, enter:

```
K N,PKF=(5,CMD='S GTF,285'),CON=N
```

Example 2

To associate a START READER and a START WRITER command with PFK 5, enter:

```
K N,PKF=(5,CMD='S RDR,001;S XWTR,292'),CON=N
```

Example 3

If PFK 3 is associated with commands S RDR,001 and S XWTR,292, and PFK 4 is associated with the command S GTF,MODE=INT,BUF=387,TIME=YES,DEBUG=YES, you can associate all three of these commands with PFK 5 by entering:

```
K N,PKF=(5,KEY=3,4),CON=Y
```

The commands associated with PFK 5 are now S RDR,001; S XWTR,292, and S GTF,MODE=INT,BUF=387,TIME=YES,DEBUG=YES, *in that order*.

Example 4

To remove a definition previously set for PFK 5, leaving PFK 5 undefined, enter:

```
K N,PKF=(5,CMD='')
```

Example 5

To assign the commands in the table PFK22 to cons8, enter:

```
K N,PKF=PKF22,L=CONS8
```

Note: The CONTROL N,PKF command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

CONTROL Command

Deleting Message Queues

Use this command to delete messages that are queued to an MCS or SMCS console. It affects only messages currently on the console's queue. Subsequent messages are queued as usual. This command is generally, though not exclusively, for use in error situations.

The command gives you more control over MCS and SMCS console message queues. If there is a WTO buffer shortage, you can delete the messages to speed up console processing or alleviate storage problems. You might need to issue the command several times to clear the console of messages.

Note: This CONTROL command is not valid for extended MCS consoles or extended MCS console message queues.

The kinds of messages that are deleted are:

- Action messages.
- All in-line messages queued for a particular console.
- WTOR messages and unconditional messages.

The kinds of messages that are not deleted are:

- Out-of-line messages. You can delete these with K E,D.
- Messages queued to SYSLOG.

The syntax is:

```
K Q[,L=name]
```

Q The CONTROL command is to delete a console's message queue.

L=name

The name of the console whose message queue is to be deleted. If this operand is omitted, the message queue of the console from which the K Q command is entered is deleted.

Example 1

To delete any messages on the issuing full-capability console's queue, enter:

```
K Q
```

Example 2

To delete messages queued on console CON2, enter:

```
K Q,L=CON2
```

Changing or Displaying Message Deletion and Format Specifications

Use the CONTROL S command to change console specifications or to display the console specifications currently in effect. Any changes you make with the CONTROL S command do not exist when you IPL the system the next time.

The following parameters on the CONSOLE statement in the CONSOLxx parmlib member control, at IPL, the same console specifications as the CONTROL S command. The system defaults are:

CON(N)	Conversational or nonconversational mode
DEL(RD)	Message deletion mode
MFORM(M)	Format of messages
RNUM(5)	Maximum number of messages included in one message roll
RTME(2)	Number of seconds between message rolls
SEG	Number of lines in the message area that the CONTROL E,SEG command deletes.

The complete syntax for the CONTROL S command is:

```
K S[,REF
      [,CON={Y|N}] [,SEG=nn] [,DEL={Y|N|R|RD|W}]
      [,RNUM=nn] [,RTME=nnn] [,MFORM=(option[,option]...)]
[,L=name]]
```

S The current console specifications are to be temporarily altered or referenced.

CON=

Conversational message deletion is requested or canceled.

Y Requests conversational message deletion.

N Cancels conversational message deletion. (non-conversational message deletion is to go into effect).

Note: The CONTROL S,CON= command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

REF

The current console specification values are to be displayed in the entry area in CONTROL command form.

SEG=nn

The size of the message segment is to be altered, where *nn* specifies the number of lines to include in the segment of messages deleted when a CONTROL E,SEG command is entered. The CONTROL S,SEG= command can accept a maximum value of 99 or the number of lines on the screen, whichever is smaller.

Note: The CONTROL S,SEG= command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

DEL=

The message deletion mode is to be changed.

Y Automatic mode of message deletion is to go into effect. That is, all flagged messages are removed from the screen whenever the screen becomes full.

N Automatic mode of message deletion is canceled. Messages must be removed manually.

CONTROL Command

R Roll mode is to go into effect. That is, a specified number of messages (determined by *RNUM*) roll off the screen each specified interval (determined by *RTME*).

RD

Roll-deletable mode of message deletion is to go into effect. That is, messages roll off as with roll mode, except that the action messages accumulate at the top of the screen.

W Wrap mode is to go into effect. When the screen is full, the next message overlays the message at the top of the screen and subsequent messages continue overlaying older messages down the screen. The separator line, with the same highlighting attribute as the warning line, moves with the new messages and includes the count of the undisplayed messages. WTOs and action messages are also overlaid.

Note: The CONTROL S,DEL= command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

RNUM=nn

The number of lines in the message roll. The CONTROL S,RNUM=nn command can accept a value of *nn* from 1 to 99 (decimal) or the number of lines on the screen, whichever is smaller, as the number of lines in the message area.

Note: The CONTROL S,RNUM=nn command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

RTME=nnn

The time interval in seconds between message rolls. The *nnn* value can be any decimal number from 1 to 999, 1/2, or 1/4. This time interval sets the MCS and SMCS screen refresh rate. Messages will be displayed each *nnn* seconds in R, RD, and W modes.

Notes:

1. The value for 3290 consoles should be 1 or higher.
2. The CONTROL S,RTME=nnn command has no effect on, and is not valid for managing, extended MCS consoles or system consoles.

MFORM=(option[,option]...)

The format of messages sent to a console is to be changed. You can control whether the text of each message (including those from JES2 and JES3) is accompanied by:

- a time stamp
- the name of the system that issues the message
- the jobname or job id of the issuer of the message

The format of a message that includes all MFORM options is:

Time stamp System name Jobname/id Message text

You can enter more than one of the options. If you do, place parentheses around the list of options and separate them with commas. The system displays the information that accompanies the message text in the order described, regardless of the order of the options you specify on the MFORM operand. *option* can be any of the following:

T Requests that each message appear with a time stamp.

S Requests that each message appear with the name of the system that sent the message.

J Requests that each message appear with the job name or job ID associated with this message.

Note: This value is initially the job name or ID of the issuer of the message, but either the issuer or subsystem code can change the value. For example, messages that JES issues often change the initial value from the JES name/id to that of the job the message is describing.

M Requests that the text of each message appear without a time stamp, the job name/job ID of its issuer, or the name of the system that sent the message. The text of the message is displayed whether or not you use this operand. At IPL, if the MFORM operand in the CONSOLxx parmlib member is not coded, the system displays the message text without time stamp, system name, or job name/ job id.

Note: M is the default MFORM option for extended MCS consoles. To change the default value for the extended MCS consoles use the RACF command, ALTUSER userid OPERPARM(MFORM(T,S,J,M,X)). See *z/OS Security Server RACF Command Language Reference* for more information. The default for MCS and SMCS consoles can be changed with the CONTROL command.

X Requests not to prefix messages flagged as exempt from sysname and jobname formatting with a sysname and jobname field when the S and/or J operands are specified. X does not affect the T operand.

L= *name*

The console this command is to affect. Before using the L operand, realize:

- You can specify this operand to change the specifications of another console only from a console with at least CONS command group authority.
- For *name*, you can specify the name of a full-capability console only from a console with master authority.
- For *name*, You cannot specify the name of a status display console.
- If you specify the name of a message stream console, you cannot specify DEL=Y or DEL=N, CON=N; if the name is for a non-display console, you cannot specify any operand other than MFORM.

Example 1

To set SEG equal to 10 lines, enter:

```
K S,SEG=10
```

Example 2

To cancel roll mode on console CON4, enter:

```
CONTROL S,DEL=N,L=CON4
```

In this case, you must delete messages manually.

Example 3

To determine the current value of SEG, enter:

CONTROL Command

K S,REF or K S

Example 4

To place a console in wrap mode, enter:

K S,DEL=W

Example 5

To display all messages on the full-capability console named CON5 with time stamps and the job names/job IDs of their issuers, enter the following command from the console with master authority:

K S,MFORM=(M,T,J),L=CON5

Notes:

1. The system displays the time stamps and the job names/job IDs in the order described earlier.
2. Whether or not you specify the option **M**, the system displays the text of the message.

Changing or Displaying Time Intervals for Dynamic Displays

CONTROL T NO LONGER SUPPORTED

The CONTROL T command is not supported on systems at z/OS V1R7 and higher.

Use the CONTROL T command to change the time interval or to see the current time interval used for updating dynamic displays on a console.

The UTME parameter on the CONSOLE statement in the CONSOLxx parmlib member controls the time interval value at IPL. If the UTME parameter is not coded, the time interval is 30 seconds. The CONTROL T command is not valid for extended MCS consoles.

The syntax for the CONTROL T command is:

K T[,REF][,L=name]
|,UTME=nnn

The parameters are:

T The time interval for updating status displays is to be changed or displayed.

REF

Displays the current value of the CONTROL K T,UTME operand.

UTME=nnn

The time interval for updating status displays in seconds, where *nnn* can be any decimal number from 10 to 999.

L= name

The console where the time interval updating or change is to occur.

Notes:

1. If the issuing console is not a display console, the system responds to the CONTROL T,REF command with the following:
IEE922I K T,UTME=nnn
The current time interval is indicated by *nnn*. You can change the time interval by entering another CONTROL T,UTME command.
2. In general, if there is a dynamic display in progress when you change the time interval, the new interval does not take effect until the interval in progress elapses. If a K D,U command is directed to the display area, however, the new interval takes effect immediately.
3. The CONTROL T command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

Changing the Operating Mode of a Console

Use the CONTROL V,USE command to change the operating mode of a console.

The USE parameter on the CONSOLE statement in the CONSOLxx parmlib member controls the operating mode of a console at IPL. If the USE parameter is not coded and the console is a display console, the console's operating mode is full-capability. Use the K V,USE command to change the operating mode for MCS consoles. You cannot change the operating mode for SMCS or extended MCS consoles.

The syntax for the CONTROL V command is:

K V[,REF	[,L=name]
,USE={FC SD MS} [,CMDSYS={sysname *}]	

The parameters are:

USE=

The operating mode of a console is to be changed.

FC

The console is to be changed to full-capability mode (input/output capability).

SD

The console is to be changed to output-only for presentation of status displays.

MS

The console is to be changed to output-only for presentation of messages other than status displays.

Note:

1. If you issue the VARY command to take the console offline and then bring the console online, the console will resume the operating mode in effect when the console was taken offline.
2. K V USE=SD and K V USE=MS operating modes are not valid for SMCS consoles.

CONTROL Command

REF

Displays the current value of the CONTROL V,USE operand.

CMDSYS

Indicates the system where all commands will be sent for processing.

sysname

The system where all commands are to be sent. If this keyword is not specified or is incorrect, the commands are processed on the system where you issue the command.

- * The system where you issue the command.

L= name

The name of the console where the specified action is to take place.

Note: The CONTROL V,USE command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

Example

To direct all commands issued from this console to processor SY2, enter:

```
K V,CMDSYS=SY2
```

Selecting the Message Levels for a Console

Use the CONTROL V,LEVEL command to specify the message levels for messages that are to be displayed at a console.

The LEVEL parameter on the CONSOLE statement in the CONSOLxx parmlib member controls the message levels for the console at IPL. If the LEVEL parameter is not coded, the system sends all messages, including broadcast messages, to the console.

The syntax for the command is:

```
K V[,REF] [,L=name ]  
      |  
      ,LEVEL=(type[,type]...)
```

The parameters are:

LEVEL=(type[,type]...)

The message levels for a console are to be changed. The following operands specify which messages are to be displayed at the console. You can enter more than one of the following operands. If you do, place parentheses around the list of operands and separate them with commas. *type* can be any of the following:

ALL The system is to display all messages routed to the console, including broadcast messages.

ALL,NB The system is to display all messages routed to the console, except for broadcast messages.

CE Critical eventual action messages (descriptor code 11) are to be displayed

E Eventual action messages (descriptor code 3) are to be displayed

I	Immediate action messages (descriptor codes 1 and 2) are to be displayed
IN	Informational messages are to be displayed
NB	Broadcast messages are not to be displayed.
R	Write-to-operator with reply (WTOR) messages are to be displayed.
UNCOND	The system is to execute this command even though it means broadcast and informational messages with certain routing codes will not be assigned to any console. Use this operand if you want broadcast and certain informational messages to be sent only to the hardcopy medium.

Notes:

1. If you don't specify NB, your console receives broadcast messages.
2. If a WTOR or action message is not directed to any console, the message is logged. It can be retrieved and displayed at a console using the DISPLAY REQUESTS command.
3. If a message is directed to a specific console, it appears there regardless of the message level of the console.
4. If you specify a second K V,LEVEL command, the K V,LEVEL command in effect is canceled.
5. If you specify only one message type on the LEVEL operand, you can omit the parentheses.

REF

Displays the current value of the CONTROL V,LEVEL operand.

L= name

The name of the console where the specified action is to take place. The issuing console is the default.

Example 1

To route only informational and broadcast messages to console CON20, enter:

K V,LEVEL=IN,L=CON20

Example 2

To route WTOR, immediate action, and broadcast messages to the issuing console, enter:

K V,LEVEL=(R,I)

Example 3

To route all messages except broadcast messages to the issuing console, enter:

K V,LEVEL=(ALL,NB)

To suppress all broadcast and informational messages destined for the issuing console, enter:

K V,LEVEL=(NB,UNCOND)

DEVSERV Command

Use the DEVSERV command to request a display of the status of DASD and tape devices. The response is a display of basic status information about a device, a group of devices, or storage control units, and optionally can include a broad range of additional information. You can display:

- Device number
- Device type
- Logical mode of the device
- Number of data sets allocated on the volume
- Volume serial label
- Channel path ID
- Status of the path
- Status of an SMS-managed device
 - Volume status
 - Storage group name
 - Storage group status
- Control unit type and model
- Control unit serial number
- Device capacity, in cylinders
- Device extended function status
- Unit control block (UCB) device type information
- Help text, when you request it
- The following, if the device belongs to the DASD storage subsystem:
 - Real device type (if what is shown is an emulated device type)
 - Control unit type and model (or emulated control unit type and model if the real and emulated control units are not the same)
 - Subsystem ID for this storage subsystem
 - Cache fast write state
 - Track caching state
 - DASD fast write state
 - State of pinned data
 - State of dual copy, PPRC, or SPARing -- if there is any
 - Address of the other device in a dual copy pair
 - Channel subsystem device address
 - Subsystem internal logical device address
 - An indication if the device extended function status information is inconsistent between MVS control blocks and the storage subsystem
 - An indication if the defined (UCB) device type is inconsistent with the real device type
 - Optionally, the total number of cylinders for each unique track format (3380, 3390, and 9345) for all of the devices within the scope of the request
- The following, if the device belongs to a tape library:
 - Device type equivalent to DTYPEn from the DS P command
 - Device status indicating online / offline and ready / not ready
 - Device type and model
 - Device serial number
 - Library identification number
 - An indication if the defined (UCB) device type is inconsistent with the real device type

You can also use several options of the DEVSERV command:

- You can compare the DEVSERV PATHS command with the DISPLAY U and DISPLAY M commands by referring to “Displaying the Status of Devices and Availability of Paths” on page 1-10.
- “Using the DEVSERV QDASD option”
- “Using the DEVSERV QTape option”
- “Using the DEVSERV QLIB option”

Using the DEVSERV QDASD option

Use the QDASD option of the DEVSERV command to validate MVS storage resident control blocks for extended function status with data acquired directly from the storage subsystem. Optionally, you can obtain a hexadecimal display of:

- the following device-related MVS system control blocks:
 - Unit control block (UCB), UCB prefix, and UCB common extension
 - Device class extension (DCE)
 - Storage subsystem control block (SSSCB)
 - Device performance characteristics table (DPCT)
- and the following data buffers acquired directly from the device:
 - Read device characteristics (RDC) data
 - Read configuration data (RCD) data
 - Sense subsystem status (SNSS) data

Using the DEVSERV QTape option

Use the QTape option of the DEVSERV command to request a hexadecimal display of:

- the following device-related MVS system control blocks:
 - Unit control block (UCB), UCB prefix, and UCB common extension
 - Device class extension (DCE)
- and the following data buffers acquired directly from the device:
 - Read device characteristics (RDC) data
 - Read configuration data (RCD) data

Using the DEVSERV QLIB option

Use the QLIB option of the DEVSERV command to:

- Request a list of tape library subsystems that are defined to the host. Libraries are listed by serial number (library-id)
- Request a list of devices within a library. Devices are listed by device number and displays the library port for each device
- Validate the connection status of devices in a library, for example, devices that are connected to the host.
- Delete an improperly defined library control block in preparation for an IODF activate.
- Issue a diagnostic state save order to a library when requested by the IBM Support Center.

DEVSERV Command

Syntax

The syntax for the DEVSERV command is:

```
DS {PATHS|P},[/]devnum[,nn][,ONLINE|ON][,NOSYM|NOS][,DUMP]
    {SMS|S }           |,OFFLINE|,OFF
    [,L=a,name,name-a]

{QDASD | QD}{[,? |

[ [ [,ccuu [,1] | ,VOL=volser ] [ [,UCB] [,DCE][,SSSCB] [,DPCT]
  [ [,NOIO] | [,RDC] [,RCD] [,SNSS] ] ] ] | ,ccuu,nnn |
  [,ccuu,nnn] ,VOL=volser |
  ,MACH=[mmp-p-sssss | XXXX-sssss] |
  ,SSID=[ssid | ALL] |
  ,TYPE=[type | ALL] ]
  [,ONLINE] [,OFFLINE] [,DEFINED] [,CHKFAIL] [,VALIDATE] [,TOTALCYL] [,RDFEATS]

| {QTAPE | QT}{[,? |

[ [ [,ccuu [,1] ] [ [,UCB] [,DCE][,NOIO] | [,RDC]
  [,RCD] ] ] | ,ccuu,nnn |
  [,ccuu,nnn] ,LIB=libid | ALL |
  ,MACH=[mmp-p-sssss | XXXX-sssss] |
  ,TYPE=[type | ALL] ]
  [,ONLINE] [,OFFLINE] [,DEFINED]

{QPAVS | QP},{dddd}
  {dddd,nn}
  {SSID=ssid}
  {ddd,{VOLUME | UCB | DCE | UNBOX}}}

[QLIB | QL],
  [LIST] {ACTIVE|INACTIVE|QUEUE}
  [LISTALL] {ACTIVE|INACTIVE}
  [LIBID] {ACTIVE|INACTIVE|VALIDATE|QUEUE|DELETE}
  [DDDD] {SS}
  [?]
```

Parameters

The basic status parameters are:

PATHS or P

Displays (in message IEE459I) the status of specified devices. The display includes any device(s) the Storage Management Subsystem (SMS) manages, but does not show any SMS information such as the status of any volume or storage group associated with the device(s).

SMS or S

Displays (in message IGD001I) the volume and storage group status for *nn* devices that SMS manages, starting with the specified device number.

[/]devnum

The device number for which the system is to display information. The number consists of three or four hexadecimal digits, optionally preceded by a slash (/).

You can specify any device that the operating system supports, except that with the SMS operand, the system displays the status of the volume and the storage group only for devices that SMS manages.

nn The number, from 1 to 32, of devices for which the system is to display the information, in ascending order beginning with the device you specify. If you do not code *nn*, the default is 1, and the system displays information only about the device you specify.

ONLINE or ON

Directs the system to display information about only those specified devices that are online to this MVS host. If you do not specify **ONLINE** or **OFFLINE**, the system displays information about both online and offline devices.

OFFLINE or OFF

Directs the system to display information about only those specified devices that are offline to this MVS host. If you do not specify **ONLINE** or **OFFLINE**, the system displays information about both online and offline devices.

NOSYM or NOS

Directs the system *not* to display (with message IEE459I) the definitions of symbols. If you omit NOSYM, the system displays the definition of all the symbols. You may use this parameter with PATHS, but not with SMS.

DUMP

Requests an SVC dump after execution of the **DEVSERV PATHS** command. If you specify both **DUMP** and a value for *nn*, the system ignores the value for *nn*. The SVC dump will cause an '0C1'X abend. You may use this parameter with **PATHS**, but not with **SMS**.

L=*a* , *name*, or *name-a*

The display area (*a*), console name (*name*), or both (*name-a*) where the display will appear. If you omit this operand, the display appears in the first available display area or in the message area of the console at which you entered the command.

QDASD or QD

Displays (in message IEE459I) diagnostic information about the status of direct access storage devices and storage control units. You use two classes of **QDASD** parameters to control the scope of the display: *unit selection parameters* and *dump selection parameters*.

- Use *unit selection parameters* to identify the units whose information you want to see. These parameters include **ccuu** and **nnn**, **VOL**, **MACH**, **SSID**, **TYPE**, **ONLINE**, **OFFLINE**, **DEFINED**, and **CHKFAIL**.

- Use the *dump selection parameters*, to define the contents of the display. Beyond the basic status information, you can specify which of the following MVS system control blocks, and/or what information acquired directly from the following device information buffers, to display in hexadecimal format.

- The system control blocks are **UCB**, **DCE**, **SSSCB**, and **DPCT**.

If you specify the parameter **NOIO**, the display will show only the storage resident information in the MVS control blocks; the command will not issue any I/O to the selected devices.

- The device information buffers are **RDC**, **RCD**, and **SNSS**.

Any hexadecimal information you request appears following the basic status information.

In addition, there is one action parameter, **VALIDATE**, and one display request parameter, **TOTALCYL**.

QDASD-specific Unit Selection Parameters

- ? Enter the command **DEVSERV QDASD,?** to view online help text.

DEVSERV Command

ccuu

The number of the DASD device you are querying.

nnn

The number of DASD devices or units to query. Valid values are from 1 to 256.

The default is 1.

VOL=volser

The serial number of the volume whose information **DEVSERV** will display. The volume must be online to the system where you issue the **DEVSERV** command.

MACH=mmpp-sssss | XXXX-sssss

A ten-character serial number specifying either the storage control unit or DASD device about which **DEVSERV** will display information. If you specify **XXXX-sssss**, the search will be done only on the **sssss** portion of the number. **MACH=** will cause an I/O operation for each DASD in the system. To limit the number of I/O operations, specify *ccuu* and *nnn*.

SSID=ssid | ALL

Specifies the identification number of the subsystem whose information **DEVSERV** will display. Valid *ssid* numbers are from 1 to FFFF.

SSID=ssid will cause an I/O operation for each DASD in the system. To limit the number of I/O operations, specify *ccuu* and *nnn*.

SSID=ALL requests a display of information for all DASD devices that support the **RCD** (Read Configuration Data) command.

TYPE=type | ALL

Specifies the type of DASD or storage control unit about which **DEVSERV** will display information. Valid *type* values are 3380, 3390, and 9345. **TYPE=ALL** causes the system to display information for all DASD devices that meet all other selection criteria. **TYPE=** will cause an I/O operation for each DASD in the system. To limit the number of I/O operations, specify *ccuu* and *nnn*.

ONLINE or ON and OFFLINE or OFF

See the basic status parameters, above.

DEFINED

Displays information about all DASD units defined in the current I/O configuration that meet all other selection criteria. The display contains information based on the existence of unit addresses (UCBs) for DASD type devices, and not on the existence of physical devices. Therefore, the display may contain information even for unit addresses that have no accessible physical devices, or for which an accessible physical device type is inconsistent with the defined device type.

CHKFAIL

Directs the system to display information about a device with a status that is inconsistent between the MVS control blocks and the device. This parameter requires a unit address with a physical device attached to it. **CHKFAIL** will cause an I/O operation for each DASD in the system. To limit the number of I/O operations, specify *ccuu* and *nnn*.

VALIDATE

Uses status information acquired directly from a device to correct inconsistent extended function status information maintained in host processor storage.

VALIDATE has no effect if the unit address has no physical device attached.

TOTALCYL

Accumulates device capacities during the scan. Valid track formats are 3380,

3390, and 9345. For each valid track format, the total capacity for all accessible devices will appear at the end of the **DEVSERV QDASD** display.

RDFEATS

Displays (via message IEE459I) the data in the MVS software control block. If there is an inconsistency between the software and hardware, RDFEATS? appears in the EF-CHK result line of the message IEE459I. The DS QD,devnum,VALIDATE command can correct the inconsistency.

QDASD Dump Selection Parameters

Parameters that are only valid when (a) specifying *ccuu*, and *nnn* has a value of 1, or (b) when specifying the **VOL=volser** parameter. Dump selection parameter information appears in hexadecimal format. The parameters are:

UCB	unit control block
DCE	device class extension block (of the UCB)
SSSCB	storage subsystem control block
DPCT	device performance characteristics table
RDC	read device characteristics
RCD	read configuration data
SNSS	sense subsystem status
NOIO	no input/output requests

NOIO prevents I/O requests and allows a display only of storage resident information. All other dump selection parameters cause I/O operations.

QTAPE or QT

Displays identification, status, and diagnostic information about tape devices in MVS/390 configurations. You can request information about a specific tape device or multiple tape devices. The **DEVSERV QTAPe** command can obtain information from any tape device that is responsive to the **SENSEID** command. You use two classes of **QTAPe** parameters to control the scope of the display: *unit selection parameters* and *diagnostic information selection parameters*.

- Use *unit selection parameters* to identify the units whose information you want to see. These parameters include *ccuu* and *nnn*, **LIB**, **MACH**, **TYPE**, **ONLINE**, **OFFLINE**, and **DEFINED**.
- Use *diagnostic information selection parameters* to define the contents of the display. Beyond the basic status information, you can select which of the following MVS system control blocks, and/or what information acquired directly from the following device information buffers, to display in hexadecimal format.

- The system control blocks are **UCB** and **DCE**.

If you specify the parameter **NOIO**, the display will show only the storage resident information in the requested MVS control blocks; the system will not issue an I/O to the selected device.

- The device information buffers are **RDC** and **RCD**.

Any hexadecimal information you request appears in the display following the basic status information.

QTAPe-specific Unit Selection Parameters

? Enter the command **DEVSERV QTAPe,?** to view online help text.

ccuu

The number of the starting, or only, tape device you are querying.

nnn

A decimal value indicating the number of sequential device numbers, starting with *ccuu*, for which to display information.

DEVSERV Command

Valid values for *nnn* are from 1 to 256. The default is 1. The value must be defaulted (unspecified), or specified with a value of 1, if you are specifying any diagnostic information selection parameters. *nnn* is valid only when you also specify *ccuu*.

nnn has a different meaning for **DEVSERV QTape** than for **DEVSERV PATHS** or **DISPLAY UNITS**. For those commands, *nnn* indicates the number of device numbers to display, ignoring gaps in the device number sequence. For the **DEVSERV QTape** command, if gaps exist in the sequence of tape device numbers defined to the operating system, and **DEFINED** is not specified, the missing tape device numbers are listed in the form '...nnnn(01)...' where nnnn is the device number and 01 is the reason code indicating that no unit control block was found for that device number. If **DEFINED** is specified, the display contains no information for missing device numbers.

LIB=libid | ALL

Requests information about the devices having the specified **libid**. If you specify **LIB=ALL**, the display will show information for all library tape devices. **LIB=** is mutually exclusive with **MACH=** and **TYPE=**.

MACH=mmpp-sssss

A ten-character serial number of either a tape control unit or a tape device. The display will show information for the specific device, or for all devices on the tape control unit having the serial number **mmpp-sssss**. If you specify the **mmpp** portion as **XXXX**, the command processor will ignore the "manufacturer" and "plant of manufacture" fields of the serial number, and will search only on **sssss**, the sequence number portion. **MACH=** is mutually exclusive with **LIB=** and **TYPE=**.

TYPE=type | ALL

Specifies the type of tape device or control unit about which **DEVSERV** will display information. Valid values for *type* include any valid four character tape device or tape control unit number.

TYPE=ALL causes the system to display information for all tape devices that meet all other selection criteria, such as **ONLINE**, **OFFLINE**, and **DEFINED**.

TYPE= is mutually exclusive with **LIB=** and **MACH=**.

ONLINE or ON and OFFLINE or OFF

See the basic status parameters, above.

DEFINED

Displays information about all tape units defined in the current I/O configuration that meet all other selection criteria.

The display contains information based on the existence of unit addresses (UCBs) for tape type devices and not on the existence of physical devices. Therefore, the display may contain information even for unit addresses that have no accessible physical devices, or for which an accessible physical device type is inconsistent with the defined device type.

The system ignores **DEFINED** if you also specify **LIB=** or **MACH=**, as these options require the existence of a physical device.

If you specify both **DEFINED** and **TYPE=ALL**, the display will include information for all tape units defined in the configuration.

If you specify both **DEFINED** and **TYPE=type**, where *type* is other than **ALL**, the display will include information only for units of the type *type*. Valid *type* values are 3400, 3480, 3490, and 3590.

For the 3400 device type, **QTAPE** supports only the devices that are responsive to the **SENSEID** command. For other tape devices, **QTAPE** annotates the display with reason code 9: QTAPE is not supported.

QTAPE Diagnostic Information Selection Parameters

Parameters that are only valid when you specify *ccuu* and *nnn*, with *nnn* having a value of 1. The data appears in hexadecimal format. The parameters are:

UCB	unit control block
DCE	device class extension block
RDC	read device characteristics block
RCD	read configuration data block
NOIO	no input/output requests

NOIO calls for a display of processor storage resident information only. The command processor issues no I/O requests to the device(s). The display will contain only unit numbers and device types defined in MVS. **NOIO** is valid only when specified in combination with **UCB** and **DCE**. **NOIO** is mutually exclusive with **RDC** and **RCD**, because those parameters can be acquired only via an I/O operation with the device.

QPAVS or QP

Displays the logical subsystem configuration as defined to the host software, and highlights any inconsistencies between the host configuration definition and the subsystem configuration for parallel access volumes (PAVs).

dddd

Specifies a three- or four-digit device number.

nn

Specifies the number of devices, a decimal number from 1 to 256.

ssid

Specifies the subsystem identification number (SSID) of the subsystem whose information **DEVSERV** will display.

VOLUME

Displays the parallel access volume (PAV) relationship information for the logical volume, including the PAV base device number and all PAV alias device numbers bound to that base.

UCB

Displays the unit control block (UCB) information associated with the device.

DCE

Displays the device class extension block (DCE) of the BASE UCB.

UNBOX

Causes QPAVS to unbox the unbound alias device if it is in a BOX state.

QLIB or QL

Displays (in message IEE459I) the requested **DEVSERV** library information. You use two classes of **QLIB** parameters to control the scope of the display: *parameters* and *sub-parameters*.

QLIB Parameters

LIST

Indicates that QLIB should display a list of the ACTIVE library-ids (the default). You can optionally generate a list of INACTIVE library-ids or QUEUE'd library orders. LIST uses the sub-parameters ACTIVE, INACTIVE, and QUEUE.

LISTALL

Produces a detailed list of all libraries, including the devices and port-ids within each library. LISTALL uses the sub-parameters ACTIVE and INACTIVE.

DEVSERV Command

LIBID

Indicates that the request is for a specific library. LIBID uses the sub-parameters ACTIVE, INACTIVE, VALIDATE, QUEUE, and DELETE.

DDDD

Indicates that the request is either for the library that contains device dddd, or is for the device dddd itself. A sub-parameter is required when DDDD is specified. DDDD uses the sub-parameter SS.

- ? Causes QLIB to display the command syntax.

QLIB Sub-parameters

ACTIVE

Displays information about the library configuration that is currently in use by the system.

INACTIVE

Displays information about the library configuration that becomes active following the next IODF activate. The INACTIVE configuration is similar to ACTIVE, but may contain additional devices or libraries.

VALIDATE

Displays the same information as the INACTIVE configuration. However, before the configuration is displayed, I/O is issued to each device in the configuration to validate connectivity to the host.

DELETE

Indicates that QLIB should delete the INACTIVE control blocks for library LIBID and not affect the existing ACTIVE library definition. The DELETE command is used to remove incorrectly defined library control blocks so that they can be rebuilt. DEVSEVR DELETE provides an alternative to the method described in information APAR II09065, which requires two IODF activates.

The DEVSEVR QLIB method is as follows:

1. Use QLIB DELETE to delete all of the devices from the incorrect control blocks.
2. Choose one action depending on whether you have LIBID and LIBPORT coded in the IODF:
 - If LIBID and LIBPORT are coded, use QLIB LIST to display that the INACTIVE control blocks have been deleted.
 - If LIBID and LIBPORT are not coded, use the VARY command to vary online the devices in the library. This will create control blocks. The system issues message IEA437I, for example:
IEA437I TAPE LIBRARY DEVICE(dev), ACTIVATE IODF=XX, IS REQUIRED
3. Use ACTIVATE IODF to redefine the devices.
4. Use QLIB LIST to display that the ACTIVE control blocks are properly defined.

QUEUE

Lists the library orders that are waiting to be completed. Such orders include:

- MOUNT
- DEMOUNT
- EJECT
- AUDIT

When an order completes, the library notifies the host and the order is removed from the queue. This QLIB display can list orders for all libraries, or be limited for a single library.

SS

Indicates that QLIB should issue a diagnostic state save to the library containing device DDDD. This command is intended to be used at the request of IBM Support Center. For example, SS can be used to diagnose a hardware error that results in a mount failure message. Automated Operator code can extract the failing device number from the failure message, then insert the device in a QLIB SS command.

QPAVS Output Formats

When UNBOX *is not* specified (see Example 13) the format is:

IEE459I			DEVSERV QPAVS		
HOST			SUBSYSTEM		
CONFIGURATION			CONFIGURATION		
-----			-----		
UNIT	UNIT	UA	UNIT	UA	TYPE
NUM.	STATUS	SSID	ADDR.		
-----	-----	-----	-----	-----	-----
ddd d aa	BASE	INV-ALIAS	sss s	uu	BASE
	ALIAS-bbbb	NOT-BASE			ALIAS-bb
	NON-PAV	NOT-ALIAS			NC
		NON-NPAV			
UCB AT V.....					
U.....					
UCB PREFIX AT V.....					
U.....					
UCB COMMON EXTENSION AT V.....					
U.....					
**** UNLISTED DEVICES AND REASON CODES					
ddd d(rc) ddd d(rc) ...					
**** n DEVICE(S) MET THE SELECTION CRITERIA					

where:

- Host Configuration
 - dddd = the device number
 - aa = the unit address from the host configuration
 - BASE = dddd is a BASE device
 - ALIAS-bbbb = the dddd is an ALIAS device, the BASE is at address bbbb
 - NON-PAV = the unit is neither a BASE nor an ALIAS
- Status
 - INV-ALIAS = on the host side, the unit is defined as an ALIAS whose BASE is different from the one on the subsystem side
 - NOT-BASE = on the host side, the unit is a BASE, while on the subsystem side it is not
 - NOT-ALIAS = on the host side, the unit is an ALIAS, while on the subsystem side it is not
 - NON-NPAV = on the host side, the unit is not a BASE nor an ALIAS, while on the subsystem side it is an ALIAS
- Subsystem Configuration
 - sss s = the SSID of the subsystem where device dddd belongs
 - uu = the unit address from the subsystem configuration
 - BASE = the unit is a BASE

DEVSERV Command

- ALIAS-bb = the unit is an ALIAS device and the BASE is at unit address bb
- NC = the unit is neither a BASE nor an ALIAS

If optional keyword UCB is specified:

V..... = virtual address

U..... = UCB content in HEX digits

For unlisted devices the reason codes are:

- 01 = Device not configured, UCB not found
- 02 = UCB not connected
- 03 = Device unavailable, SCP routine in control
- 04 = Subchannel error
- 05 = Device boxed
- 06 = UCB not a DASD
- 07 = Device I/O error
- 08 = Device is not a DASD
- 09 = DSE-1 CCW build failed
- 0A = Device is an unbound PAV-ALIAS
- 0B = Device is a secondary of a PPRC pair

When UNBOX is specified (see Example 14), the format is:

IEE459I DEVSERV QPAVS
 e.....

where:

- e..... = The DEVSERV QPAV UNBOX command has been executed successfully.
- e..... = The DEVSERV QPAV UNBOX command has been executed with a return code of yy and a reason code of zz.
- e..... = dddd is not an unbound PAV-ALIAS device. The DEVSERV QPAV UNBOX command is not executed.
- e..... = dddd is not in BOX state. The DEVSERV QPAV UNBOX command is not executed.

Examples

Example 1

To display the status of a DASD with device number 380, enter:

DS P,380

Example 2

To display the status of a DASD with device number 3480, enter:

DS P,/3480

Example 3

To display the status of all online devices with device numbers 380 through 38F, enter:

DS P,380,16,ON

Example 4

To display the status of SMS-controlled device 430 and the seven devices whose addresses follow it, enter:

```
ds s,430,8
```

This command would produce the following display:

```
11.49.20      IGD001I 11:49:20 DEVSERV SMS 455
```

UNIT,TYPE	,M,VOLUME,VOLSTAT	STORGRP,	SGSTAT
430 ,3380	,0,XP0101,ENABLED	SXP01	,ENABLED
431 ,3380	,A,XP0201,ENABLED	SXP02	,ENABLED
432 ,3380	,A,XP0202,ENABLED	SXP02	,ENABLED
433 ,3380	,0,XP0301,ENABLED	SXP03	,ENABLED
434 ,3380	,0,XP0302,ENABLED	SXP03	,ENABLED
435 ,3380	,0,XP0303,ENABLED	SXP03	,ENABLED
436 ,3380	,0,338001,STRG/RSDNT,	VOLUME NOT MANAGED BY SMS	
437 ,3380	,A,SMSPCK,STRG/RSDNT,	VOLUME NOT MANAGED BY SMS	

Note: Indications in the M column are: O=online, A=allocated, F=offline.

Example 5

The following two sample displays compare the response to DEVSERV P with the response to DEVSERV S. Note that DEVSERV P provides volser and CHPID information, while DEVSERV S provides SMS volume and storage group status.

Issuing **DEVSERV P,430** produces this display:

```
00- 16.24.41      devserv p,430
```

UNIT DTYPE	M	CNT VOLSER	CHPID=	PATH STATUS
430, 3380D,	0,	000, XP0101,	25+=	2E=+

***** SYMBOL DEFINITIONS *****
O = ONLINE + = PATH AVAILABLE

Issuing **DEVSERV S,430** produces this display:

```
16.24.26      devserv s,430
```

UNIT, TYPE	,M, VOLUME, VOLSTAT	STORGRP,	SGSTAT
430, 3380	,0, XP0101, ENABLED	SXP01,	QUIESCED

***** SYMBOL DEFINITIONS *****
O = ONLINE + = PATH AVAILABLE

Example 6

The extra header line and data line appear in the response only when there are 3990 Model 3 Storage Controls in the system. If record caching has not been installed, the RC column in the third header line is left blank.

Issuing **DEVSERV P,430,2** produces this display:

```
00- 16.24.41      devserv p,430,2
```

UNIT DTYPE	M	CNT VOLSER	CHPID=	PATH STATUS	RC	TC	DFW	PIN	DC-STATE	CCA	DDC	ALT	CU-TYPE
430, 3380D,	0,	000, XP0101,	25+=	2E=+	IE	YY	NN	N	SIMPLEX	C0	01		3990-3

DEVSERV Command

```
431, 3380E, 0, 000, XP0101, 25+= 2E=+
IE YY NN N SIMPLEX C1 01 3990-3

***** SYMBOL DEFINITIONS *****
O = ONLINE + = PATH AVAILABLE
```

Example 7

This example shows the dual copy status.

Issuing **DS P,D2A,2** produces this display:

```
IEE459I 10.06.45 DEVSERV PATHS 297
UNIT DTYPE M CNT VOLSER CHPID=PATH STATUS
    RTYPE SSID CFW TC DFW PIN DC-STATE CCA DDC ALT CU-TYPE
0D2A,3380D ,0,000,DSFXA0,1B=+ 9B=+ 1C=+ 9C=+
    00AB Y NY. NY. N PRIMARY 20 20 0D2B 3990-3
0D2B,3380D ,F,000, ,1B=+ 9B=+ 1C=+ 9C=+
    00AB Y NY. NY. N SECONDARY 21 21 0D2A 3990-3

***** SYMBOL DEFINITIONS *****
F = OFFLINE O = ONLINE
+ = PATH AVAILABLE
```

Example 8

This example shows the sparing status.

Issuing **DS P,F7E** produces this display:

```
IEE459I 16.02.11 DEVSERV PATHS 389
UNIT DTYPE M CNT VOLSER CHPID=PATH STATUS
    RTYPE SSID CFW TC DFW PIN DC-STATE CCA DDC ALT CU-TYPE
0F7E,33903 ,F,000, ,B5=X B6=X B9=X BA=X
    9392-2 00FD Y NY. NN. N SPARE 2E 00 3990-3

***** SYMBOL DEFINITIONS *****
F = OFFLINE X = INDETERMINATE FAILING UNIT
```

Example 9

This example shows the PPRC status.

Issuing **DS P,F7C** produces this display:

```
IEE459I 15.14.12 DEVSERV PATHS 113
UNIT DTYPE M CNT VOLSER CHPID=PATH STATUS
    RTYPE SSID CFW TC DFW PIN DC-STATE CCA DDC ALT CU-TYPE
0F7C,33903 ,F,000, ,B1=X B7=X C1=X C7=X
    053F Y YY. YY. N PSEONDRY 0C 0C 3990-6

***** SYMBOL DEFINITIONS *****
F = OFFLINE X = INDETERMINATE FAILING UNIT
```

Example 10

This example uses the DEVSERV QTape command to diagnose an error, namely an inconsistent device definition.

- VARY ONLINE fails.

V 931,ONLINE

```
IEE103I UNIT 0931 NOT BROUGHT ONLINE
IEE763I NAME- IECDINIT CODE= 0000000800000000
```

```
IEA435I PHYSICAL DEVICE INCONSISTENT WITH LOGICAL DEFINITION
IEE764I END OF IEE103I RELATED MESSAGES
```

- b. DEVSERV QTAPE shows inconsistent device definition.

```
DS QT,931,1
```

```
IEE459I 15.28.22 DEVSERV QTAPE
UNIT DTYPE DSTATUS CUTYPE DEVTYPE CU-SERIAL DEV-SERIAL ACL LIBID
0931 3480X OFFLINE 3490A20 3490B40? 0113-97231 0113-97231 I
**** 1 DEVICE(S) MET THE SELECTION CRITERIA
```

Example 11

This example illustrates the help text provided when you issue the command:

```
DS QT,?
```

```
IEE459I 15.27.49 DEVSERV QTAPE
DEVSERV QTAPE COMMAND SYNTAX:
DS QT,ccuu,n,filter1,filter2,diagnostic info
ccuu -- device number, n -- number of devices (1-256)
filter1 -- LIB=ALL or library id, or
          MACH=ALL or cu or device serial, or
          TYPE=ALL or device type or defined device type
filter2 -- DEFINED, ONLINE, OFFLINE
          DEFINED is valid only if TYPE= is selected
diagnostic info -- UCB, DCE, RDC, RCD, NOIO
          valid only if n=1 is specified
DS QT,LIB=libid,filter
          libid -- ALL or library id
          filter -- ONLINE, OFFLINE
DS QT,MACH=serialnmbr,filter
          serialnmbr -- control unit or device serial
          filter -- ONLINE, OFFLINE
DS QT,TYPE=type,filter
          type -- ALL or defined device type
          filter -- DEFINED, ONLINE, OFFLINE
```

Example 12

This example illustrates the basic DEVSERV QTAPE display without hexadecimal data.

```
DS QT,TYPE=ALL
```

```
IEE459I 12.57.36 DEVSERV QTAPE
UNIT DTYPE DSTATUS CUTYPE DEVTYPE CU-SERIAL DEV-SERIAL ACL LIBID
0930 3480SX ON-NRD 3490A02 3490B04 0112-47671 0112-47671 I
093F 3480SX OFFLINE 3490A02 3490B04 0112-47671 0112-47671 I
0990 3490 ON-RDY 3490A20 3490B40 0113-97231 0113-97231 I-A
09A0 3490 OFFLINE 3490C2A 3490C2A 0113-55565 0113-55565 I 10382
**** UNLISTED DEVICE(S) AND REASON CODES :
093E(05) 093F(05)
**** 4 DEVICE(S) MET THE SELECTION CRITERIA
**** (05) - DEVICE BOXED
```

Example 13

These three variations illustrate the DEVSERV QPAVS command when the UNBOX parameter is not specified.

```
DS QP,D2FF,VOLUME
```

```
IEE459I 08.20.32 DEVSERV QPAVS 591
HOST                               SUBSYSTEM
```

DEVSERV Command

CONFIGURATION			CONFIGURATION		
UNIT	UNIT	UA	UNIT	ADDR.	TYPE
NUM.	UA	TYPE	STATUS	SSID	
D222	22	BASE		0102	22
D2FE	FE	ALIAS-D222		0102	FE
D2FF	FF	ALIAS-D222		0102	FF
**** 3 DEVICE(S) MET THE SELECTION CRITERIA					
DS QP,E27B,UCB					
IEE459I 08.03.55 DEVSERV QPAVS 920					
HOST CONFIGURATION			SUBSYSTEM CONFIGURATION		
UNIT	UNIT	UA	UNIT	ADDR.	TYPE
NUM.	UA	TYPE	STATUS	SSID	
E27B	7B	ALIAS-E200		3205	7B
UCB AT V02336568					
0088FF04E27B0000 0000000008E4C3C2 3010200F00336541 0000000000000000					
000000000100000 00F451F0023362A8 0080000000000000					
UCB PREFIX AT V023BF6E0					
0004004000000000 000000000013920 289C0DB2B00080F0 0A0B1213FFFFFF					
010800000000001					
UCB COMMON EXTENSION AT V02336540					
00000900202A0000 023BF6E000000000 000000000FC31C 00F4539000000000					
**** 1 DEVICE(S) MET THE SELECTION CRITERIA					
DS QP,E279,2					
IEE459I 08.23.08 DEVSERV QPAVS 952					
HOST CONFIGURATION			SUBSYSTEM CONFIGURATION		
UNIT	UNIT	UA	UNIT	ADDR.	TYPE
NUM.	UA	TYPE	STATUS	SSID	
E279	79	ALIAS-E200		3205	79
**** UNLISTED DEVICE(S) AND REASON CODES :					
E27A(0A)					
**** (0A) - DEVICE IS AN UNBOUND PAV-ALIAS					
**** 1 DEVICE(S) MET THE SELECTION CRITERIA					

Example 14

This command illustrates the DEVSERV QPAVS command when the UNBOX parameter IS specified.

DS QP,E200,UNBOX

IEE459I 08.12.53 DEVSERV QPAVS 935
E200 IS NOT AN UNBOUND PAV-ALIAS DEVICE.
THE DEVSERV QPAV UNBOX COMMAND IS NOT EXECUTED.

Example 15

This example illustrates the use of DEVSERV QLIB to delete the INACTIVE control blocks for library 10382.

DS QL,10382,DELETE

*04 Reply 'YES' to delete the INACTIVE configuration for library 10382,
any other reply to quit.
R 4, YES

```
IEE459I 10.27.54 DEVSERV QLIB 432
Inactive configuration for library 10382 successfully deleted
```

Example 16

This example illustrates the use of DEVSERV QLIB to list all of the queued requests.

```
DS QL,LIST,QUEUE
```

```
IEE459I 10.19.40 DEVSERV QLIB 421
LIBID DEVICE REQUEST VOLSER CATEGORY TIME ON QUEUE
10382 F018 MOUNT VOL001 000F 3 MINUTE(S)
10382 F020 MOUNT VOL002 000F 1 MINUTE(S)
10382 F031 EJECT VOL003 000F 8 MINUTE(S)
10382 F021 DEMOUNT VOL005 0001 3 MINUTE(S)
```

Example 17

This example illustrates the use of DEVSERV QLIB to list all of the libraries that are defined to the system and then list all of the devices for library 10382. An * indicates that at least one device in library 10382 has been initialized. It also indicates that library 15393 is logically defined to the system, but has never gone through device initialization. Initialization occurs during IPL or IODF activate for any library devices that are connected to the system. For devices that are connected after IPL or IODF activate, initialization occurs when the device is varied online. For library 10382, devices on ports 03, 02, and 01 have never been initialized.

```
DS QL,LIST
```

```
IEE459I 09.48.44 DEVSERV QLIB 391
The following libids are defined in the ACTIVE configuration:
*10382 15393
DS QL,10382
IEE459I 09.49.03 DEVSERV QLIB 394
LIBID PORTID DEVICES
10382 04      0940* 0941* 0942* 0943* 0944* 0945* 0946* 0947*
                  0948* 0949* 094A* 094B* 094C* 094D* 094E* 094F*
          03      09A0 09A1 09A2 09A3 09A4 09A5 09A6 09A7
                  09A8 09A9 09AA 09AB 09AC 09AD 09AE 09AF
                  0DB0 0DB1 0DB2 0DB3 0DB4 0DB5 0DB6 0DB7
                  0DB8 0DB9 0DBA 0DBB 0DBC 0DBD 0DBE 0DBF
          02      09D0 09D1 09D2 09D3 09D4 09D5 09D6 09D7
                  09D8 09D9 09DA 09DB 09DC 09DD 09DE 09DF
          01      F990 F991 F992 F993 F994 F995 F996 F997
```

Example 18

This example illustrates the use of DEVSERV QLIB to verify the connectivity of devices in library 15393.

This display indicates that devices DC8 and DC9 were connected to the system at some point and were initialized.

```
11.55.47 SYSTEM1      ds ql,15393,INACTIVE
11.55.48 SYSTEM1      IEE459I 11.55.47 DEVSERV QLIB 471
The following are defined in the INACTIVE configuration:
LIBID PORTID DEVICES
15393 03      0DC8* 0DC9* 0DC0 0DC1 0DC2 0DC3 0DC4 0DC5
                  0DC6 0DC7 0DCA 0DCB 0DCC 0DCD 0DCE 0DCF
```

This display uses VALIDATE to determine the current state of device connectivity:

- DC8 is no longer connected to the system

DEVSERV Command

- DC9 is still connected
- DC5 has now become connected

```
11.56.15 SYSTEM1      ds ql,15393,validate
11.56.16 SYSTEM1      IEE459I 11.56.15 DEVSERV QLIB 476
The following are defined in the INACTIVE configuration:
LIBID  PORTID      DEVICES
15393  03          0DC8  0DC9* 0DC0  0DC1  0DC2  0DC3  0DC4  0DC5*
                  0DC6  0DC7  0DCA  0DCB  0DCC  0DCD  0DCE  0DCF
```

DISPLAY Command

Use the DISPLAY system command to display information about the operating system, the jobs and application programs that are running, the processor, devices that are online and offline, central storage, workload management service policy status, and the time of day. Use the following table to access the pages on which you can find details about a particular use of the DISPLAY command.

Table 4-10. Summary of the DISPLAY Command

Topic	Command
"Displaying System Activity" on page 4-149	DISPLAY A
"Displaying APPC/MVS Information" on page 4-100	DISPLAY APPC
"Displaying ASCH Configuration Information" on page 4-106	DISPLAY ASCH
"Displaying Page Data Set Information" on page 4-108	DISPLAY ASM
	DISPLAY ASM
"Displaying the current system level Language Environment run-time options" on page 4-110	DISPLAY CEE
"Displaying CONTROL Command Functions" on page 4-110	DISPLAY C,K
"Displaying Attached Coupling Facility Information" on page 4-111	DISPLAY CF
"Displaying Console Group Definitions" on page 4-111	DISPLAY CNGRP
"Displaying Console Status Information" on page 4-112	DISPLAY CONSOLES
"Displaying DIAG Parmlib Information" on page 4-116	DISPLAY DIAG
"Displaying Data Lookaside Facility Information" on page 4-117	DISPLAY DLF
"Displaying Dump Options or Dump Data Set Status" on page 4-119	DISPLAY DUMP
"Displaying Extended MCS Information" on page 4-122	DISPLAY EMCS
"Displaying the Timer Synchronization Mode and ETR Ports" on page 4-127	DISPLAY ETR
"Displaying Global Resource Serialization Information" on page 4-128	DISPLAY GRS
"Displaying TSO/E Parmlib Information" on page 4-139	DISPLAY IKJTSO
"Displaying I/O Configuration Information" on page 4-140	DISPLAY IOS,CONFIG
"Displaying Captured UCB Information" on page 4-141	DISPLAY IOS,CAPTUCB
"Displaying Dynamic Channel Path Management Information" on page 4-141	DISPLAY IOS,DCM
"Displaying Encryption Key Manager (EKM) Status" on page 4-142	DISPLAY IOS,EKM
"Displaying FICON Switch Data Information" on page 4-143	DISPLAY IOS,FICON
"Displaying IOS Group Information" on page 4-143	DISPLAY IOS,GROUP
"Displaying IOS HYPERPAV Information" on page 4-143	DISPLAY IOS,HYPERPAV
"Displaying MIDAW Facility Status" on page 4-144	DISPLAY IOS,MIDAW
"Displaying MIH and I/O Timing Limits" on page 4-144	DISPLAY IOS,MIH
"Displaying IOS Storage Residency Information" on page 4-148	DISPLAY IOS,STORAGE
"Displaying the Devices Stopped by the IOACTION Command" on page 4-148	DISPLAY IOS,STOP
"Displaying IPL Information" on page 4-148	DISPLAY IPLINFO

DISPLAY Command

Table 4-10. Summary of the DISPLAY Command (continued)

Topic	Command
“Displaying System Activity” on page 4-149	DISPLAY JOBS or DISPLAY J or DISPLAY A or DISPLAY TS
“Displaying Library Lookaside Information” on page 4-160	DISPLAY LLA
“Displaying the System Logger and its Log Streams” on page 4-161	DISPLAY LOGGER
“Displaying the Logrec Recording Medium” on page 4-165	DISPLAY LOGREC
“Displaying System Configuration Information” on page 4-166	DISPLAY M
“Displaying MVS Message Service Status and Languages” on page 4-171	DISPLAY MMS
“Displaying Message Suppression, Retention, Color, Intensity, and Highlighting Options” on page 4-172	DISPLAY MPF
“Displaying z/OS UNIX System Services Status” on page 4-174	DISPLAY OMVS
“Displaying Operator Information (OPDATA)” on page 4-191	DISPLAY OPDATA
“Displaying PARMLIB Information” on page 4-192	DISPLAY PARMLIB
“Displaying Commands Defined for PFKs” on page 4-194	DISPLAY PFK
“Displaying Registered Products” on page 4-195	DISPLAY PROD
“Displaying Entries in the List of APF-Authorized Libraries” on page 4-197	DISPLAY PROG,APP
“Displaying Dynamic Exits” on page 4-198	DISPLAY PROG,EXIT
“Displaying LNKLST Information” on page 4-200	DISPLAY PROG,LNKLST
“Displaying LPA Information” on page 4-202	DISPLAY PROG,LPA
“Displaying System Requests” on page 4-202	DISPLAY R
“Displaying Resource Recovery Services (RRS) Information” on page 4-208	DISPLAY RRS
“Displaying RTLS Information” on page 4-211	DISPLAY RTLS
“Displaying SLIP Trap Information” on page 4-214	DISPLAY SLIP
“Displaying SMF Data” on page 4-215	DISPLAY SMF
“Displaying Storage Management Subsystem Information” on page 4-215	DISPLAY SMS
“Displaying Information about All Subsystems” on page 4-232	DISPLAY SSI
“Displaying Static System Symbols” on page 4-235	DISPLAY SYMBOLS
“Displaying the Local and Coordinated Universal Time and Date” on page 4-235	DISPLAY T
“Displaying Component or Transaction Trace Status” on page 4-236	DISPLAY TRACE
“Displaying System Activity” on page 4-149	DISPLAY TS
“Displaying Device Status and Allocation” on page 4-239	DISPLAY U
“Displaying Unicode Services” on page 4-242	DISPLAY UNI
“Displaying Virtual Storage Information” on page 4-245	DISPLAY VIRTSTOR,HVSHARE
“Displaying Workload Manager Information” on page 4-245	DISPLAY WLM

Table 4-10. Summary of the DISPLAY Command (continued)

Topic	Command
"Displaying Cross System Coupling Facility (XCF) Information" on page 4-251	DISPLAY XCF

Some uses of the DISPLAY command are described in other books. They are:

- TCPIP activity and functions. See *z/OS Communications Server: IP System Administrator's Commands*.
- VTAM network activity and functions. See *z/OS Communications Server: SNA Operation*.

Scope in a Sysplex

The following table describes the conditions under which the DISPLAY command has sysplex scope. See "Using Commands That Have Sysplex Scope" on page 1-11 for an explanation of sysplex scope. If a command has **All** under "Conditions," then the command has sysplex scope under all circumstances and for all variations.

Table 4-11. Sysplex Scope for DISPLAY Command

Command	Conditions
DISPLAY CF	Has sysplex scope only when displaying information about the coupling facility and only for those systems connected to the coupling facility. Does not have sysplex scope when displaying an individual system's coupling facility configuration information (coupling facility channels and paths).
DISPLAY CNGRP	All
DISPLAY CONSOLES	Has sysplex scope unless you specify DISPLAY C,B or DISPLAY C,U=.
DISPLAY DUMP	Has sysplex scope only when you issue the OPTIONS parameter to display the results of a CHNGDUMP,SDUMP,SYSFAIL,STRLIST= command.
DISPLAY EMCS	Has sysplex scope, except when you specify STATUS=B or STATUS=ERR. When you specify STATUS=FULL, consoles from all systems will be displayed (for consoles that are not active on the system where this command is processed, some information will not be displayed).
DISPLAY GRS	Has sysplex scope unless you specify SUSPEND. Also, note the following about DISPLAY GRS,C and DISPLAY GRS,RES: the output generated by these commands includes both system-specific information (S=SYSTEM) and sysplex information (S=SYSTEMS). The S=SYSTEM information is valid only for the system on which you issue the command. The S=SYSTEMS information is identical regardless of the system on which you issue the command.
DISPLAY OPDATA	Has sysplex scope except for the TRACKING operand.
DISPLAY PFK	Has sysplex scope only when you specify CN=.

DISPLAY Command

Table 4-11. Sysplex Scope for DISPLAY Command (continued)

Command	Conditions
DISPLAY R	Has sysplex scope, but the output might be different on different consoles, because the output of DISPLAY R is dependent on the routing criteria for the console specified by CN=. If you do not specify CN=, the routing criteria of the console issuing the command is used. If you issue the command in a program (by using the MGCRC macro) the console you specify in the macro is used. If you specify a console ID of 0, all retained messages are included in the command response.
DISPLAY WLM	All
DISPLAY XCF,ARMSTATUS	Has sysplex scope provided all systems are using the same ARM couple data set.
DISPLAY XCF,CF	Has sysplex scope provided all systems in the sysplex are connected to the same coupling facilities.
DISPLAY XCF,COUPLE	Has sysplex scope as long as all systems are using the same types of couple data sets, as specified on the TYPE parameter (SYSPLEX, ARM, CFRM, SFM, LOGR, and WLM.) If you do not specify the TYPE parameter, only system-specific data is displayed.
DISPLAY XCF,GROUP	All
DISPLAY XCF,POLICY	Has sysplex scope as long as all systems are using the same types of couple data sets, as specified on the TYPE parameter (ARM, CFRM, SFM, and LOGR.)
DISPLAY XCF,STRUCTURE	Has sysplex scope provided all systems in the sysplex are connected to the same coupling facilities.
DISPLAY XCF,SYSPLEX	All

Syntax

The syntax for each of the many variations of the DISPLAY command is shown immediately preceding its respective parameter list.

DISPLAY or D

Notes:

1. You must supply all commas between DISPLAY U or DISPLAY R and a specified positional operand. For example, DISPLAY U,,ONLINE.
2. You must enclose any comments on the commands DISPLAY PROD, DISPLAY PROG, and DISPLAY RTLS in slash-asterisk — asterisk-slash pairs. See "System Command Formats" on page 4-13 for further information.

Displaying APPC/MVS Information

Use the DISPLAY APPC command to display information about the APPC/MVS configuration.

```

D APPC,{TP[,SUMMARY|SUM|S][,{ASID|A}=asid]
      |,LIST|,L          [,ASNAME=asname]
      |,ALL|,A           [,DIR=IN|OUT]
                           [,IT=sssss[.ttt]]
                           [,LLUN=lluname]
                           [,LTPN=ltpname]
                           [,PNET=pnetid]
                           [,PLUN=pluname]
                           [,PTPN=ptpname]
                           [,SCHED={schedname}]
                           {*NONE*}
                           [,STPN=stpname]
                           [,USERID=userid]

      {UR[,SUMMARY|SUM|S][,URID=urid]
       |,LIST|,L          [,LUWID=luwid]
       |,ALL|,A           [,PNET=pnetid]
                           [,PLUN=pluname]
                           [,LLUN=lluname]

      {SERVER[,SUMMARY|SUM|S][,{ASID|A}=asid]
       |,LIST|,L          [,ASNAME=asname]
       |,ALL|,A           [,LLUN=lluname]
                           [,STPN=stpname]

      {LU[,SUMMARY|SUM|S][,LLUN=lluname]
       |,LIST|,L          [,PNET=pnetid]
                           [,PLUN=pluname]
       |,ALL|,A           [,SCHED={schedname}]
                           {*NONE*}

[,L={a|name|name-a}]

```

The parameters are:

TP

Indicates that the system is to display information (message ATB102I) about local transaction programs (TPs) and their conversations.

SUMMARY or SUM or S

Indicates that the system is to use the SUMMARY form of output. The resulting message contains the number of local transaction programs and the number of inbound and outbound conversations.

LIST or L

Indicates that the system is to use the LIST form of output. This output is the same as the SUMMARY display, followed by a list of the transaction programs that are running or that were selected through optional keyword filter parameters. Each entry in the list contains the name of a local transaction program, along with related information.

ALL or A

Indicates that the system is to use the ALL form of output. This output is the same as the LIST output, except that the system inserts a sublist after each entry in the list of transaction programs. The sublist contains information about each conversation associated with the particular local transaction program.

DISPLAY APPC Command

Note: For the output of the DISPLAY APPC,TP command, see the description of message ATB102I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

SERVER

Indicates that the system is to display information (message ATB103I) about APPC/MVS servers and the allocate queues they are serving.

SUMMARY or SUM or S

Indicates that the system is to use the SUMMARY form of output. The resulting display contains the number of servers, the number of allocate queues, and the total number of queued allocate requests in the system.

LIST or L

Indicates that the system is to use the LIST form of output. This output is the same as the SUMMARY display, followed by a list of allocate queues. Each entry in the list contains the name of the served transaction program associated with the allocate queue, along with related information.

ALL or A

Indicates that the system is to use the ALL form of output. This output is the same as the LIST output, except that the system inserts a sublist after each entry in the list. The sublist contains information about each server for a particular allocate queue.

Note: For the output of the DISPLAY APPC,SERVER command, see the description of message ATB103I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

LU

Indicates that the system is to display information (message ATB101I) about logical units (LUs).

SUMMARY or SUM or S

Indicates that the system is to use the SUMMARY format of output. The resulting display contains the number of active, outbound, pending, and terminating logical units.

LIST or L

Indicates that the system is to use the LIST form of output. This output is the same as the SUMMARY display, followed by a list of logical units. Each entry in the list contains the name of a local logical unit, and related information.

ALL or A

Indicates that the system is to use the ALL form of output. This output is the same as the LIST output, except that the system inserts a sublist after each entry in the list. The sublist contains the names of the partner logical units that have sessions established with the local logical unit.

Note: For the output of the DISPLAY APPC,LU command, see the description of message ATB101I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

The following is a list of keywords that filter the displays. When you specify a filter keyword, the system displays only the data that meet the keyword’s criteria.

Notes:

1. The same keyword cannot be used twice with a single command.
2. A command line cannot exceed 126 characters in length.

ASID or A=asid

The address space identifier of the transaction program (with DISPLAY APPC,TP) or server (with DISPLAY APPC,SERVER). Specify a one- to four-digit hexadecimal value.

ASNAME=asname

The address space name of the transaction program (with DISPLAY APPC,TP) or server (with DISPLAY APPC,SERVER). The address space name is one to eight alphanumeric (a-z, 0-9) or special (@, #, \$) characters, but the first character **cannot** be numeric (0-9).

DIR=IN or OUT

The direction of the conversation. DIR can have the values: IN (for INBOUND), or OUT (for OUTBOUND). These values specify INBOUND conversations, which the partner transaction program allocated, and OUTBOUND conversations, which the local transaction program allocated.

IT=sssss[.ttt]

The idle time for a conversation. Idle time is the amount of time that the local transaction program waits for data or for a confirmation from the partner transaction program. sssss specifies the number of seconds, from 0 - 99999; ttt specifies the number of thousandths of a second, from .0 - .999. When you specify this keyword, the system displays only conversations with an idle time greater than or equal to the value you specify.

LLUN=/luname

The local logical unit name. This name is one to eight alphanumeric (a-z, 0-9) or special (@, #, \$) characters, but the first character **cannot** be numeric (0-9).

LTPN=ltpname

The local transaction program name. This name is one to 64 alphanumeric (a-z, 0-9) or special (@, #, \$) characters. The name may also contain the characters in character set 00640, except for the following:

- comma(,) - used as a parameter delimiter and means that the preceding character is interpreted as the end of the transaction program name
- blank() - used as a parameter delimiter and means that the preceding character is interpreted as the end of the transaction program name
- asterisk(*) - used to filter the name of the transaction program. It can only be used as the last character of the name. An asterisk causes the LTPN keyword filter to match every transaction program name that begins with the characters preceding the asterisk.

You abbreviate the name of the local transaction program by entering the first part of the name, followed by an asterisk. For example, PROCESS* matches every local transaction program name that begins with the letters PROCESS.

To list all the local, non-served TPs on the system (and filter out served TPs), enter LTPN=*.

You can also specify the asterisk as the last character of the displayable format of a SNA service transaction program name.

The displayable format of the SNA service transaction program name is in the form:

-X'hh'ccc

DISPLAY APPC Command

Where *hh* is the hexadecimal value for the first non-displayable character and *ccc* is a character string (one to three characters) from character set Type A. You can abbreviate the name of the SNA transaction program by entering the first part of the name, followed by an asterisk.

Character sets 00640 and Type A are listed in *z/OS MVS Planning: APPC/MVS Management*.

PNET=*pnetid*

The network ID where the partner LU resides. This ID is one to eight alphanumeric (a-z, 0-9) or special (@, #, \$) characters, and is equivalent to the network-ID portion of a network-qualified LU name. Together with the PLUN parameter, PNET filters the information to be displayed.

PLUN=*pluname*

The partner logical unit name. This name is one to eight alphanumeric (a-z, 0-9) or special (@, #, \$) characters, and is equivalent to the network-LU-name portion of a network-qualified LU name. The first character **cannot** be numeric (0-9).

Together with the PNET parameter, PLUN filters the partner LU information to be displayed, as follows:

- PNET=*pnetid*, without a value for PLUN, results in a display of all partner LUs in only the specified network.
- PLUN=*pluname*, without a value for PNET, results in a display of all the partner LUs that share the same specified network LU name in all the networks in the installation.
- PNET=*pnetid* with PLUN=*pluname* results in a display of only the partner LU that has a network-qualified name that matches the specified network ID and network LU name.
- A DISPLAY command without specified values for PNET and PLUN results in a display of information for all partner LUs in all networks.

PTPN=*ptpname*

The partner transaction program name. This name is one to 64 alphanumeric (a-z, 0-9) or special (@, #, \$) characters. The name may also contain the characters in character set 00640, except for the following:

- comma(,) - used as a keyword delimiter and means that the preceding character is interpreted as the end of the transaction program name
- blank() - interpreted as the end of the command and means that the preceding character is interpreted as the end of the transaction program name
- asterisk(*) - used to filter the partner transaction program name, it can only be used as the last character of the name. It causes the PTPN keyword filter to match every transaction program name which begins with the characters preceding the asterisk.

You can abbreviate the name of the partner transaction program by entering the first part of the name, followed by an asterisk. For example, PROCESS* matches every partner transaction program name that begins with the letters PROCESS.

You can also specify the asterisk as the last character of the displayable format of a SNA service transaction program name.

The displayable format of the SNA service transaction program name is in the form:

X'hh'ccc

Where *hh* is the hexadecimal value for the first non-displayable character and *ccc* is a character string (one to three characters) from character set Type A. You can abbreviate the name of the SNA transaction program by entering the first part of the name, followed by an asterisk.

Character sets 00640 and Type A are listed in *z/OS MVS Planning: APPC/MVS Management*.

SCHED=schedname or *NONE*

The transaction scheduler name or *NONE*. The scheduler name is one to eight alphanumeric (a-z, 0-9) characters. Special characters (@, #, \$) are not permitted.

For the DISPLAY APPC,TP command, the system displays only the transaction programs scheduled for the transaction scheduler you specify. If you specify *NONE* instead of the name of a transaction scheduler, the system displays only those transaction programs that are not associated with a transaction scheduler (such as transaction programs engaged in an outbound conversation, or transaction programs that are served by an APPC/MVS server).

For the DISPLAY APPC,LU command, the system displays only the logical units controlled by the transaction scheduler you specify. If you specify *NONE* instead of the name of a transaction scheduler, the system displays only those logical units that are not associated with a transaction scheduler. These logical units are known as NOSCHED logical units.

Note: The installation defines the names of the transaction schedulers on the SCHED keyword in the APPCPMxx parmlib member. Although lower-case alphabetic characters are not permitted for scheduler names specified in parmlib, you can use lower-case on the SCHED keyword name. The system translates lower-case characters to their upper-case equivalent before it processes the DISPLAY APPC command.

STPN=stpname

The name of the served transaction program (TP). For DISPLAY APPC,TP, this is the name of a TP that was served by an APPC/MVS server on this system. For DISPLAY APPC,SERVER, this is the TP name for which the server registered.

The name of the served transaction program is one to 64 alphanumeric (a-z, 0-9) or special (@, #, \$) characters. This name may also contain the characters in character set 00640, except for the following:

- comma(,) - used as a keyword delimiter and means that the preceding character is interpreted as the end of the transaction program name
- blank() - interpreted as the end of the command and means that the preceding character is interpreted as the end of the transaction program name
- asterisk(*) - used to filter the served transaction program name, it can only be used as the last character of the name. It causes the STPN keyword filter to match every transaction program name that begins with the characters preceding the asterisk.

You abbreviate the name of the served transaction program by entering the first part of the name, followed by an asterisk. For example, SERV* matches every served transaction program name that begins with the letters SERV.

To list all the served TPs on the system (and filter out non-served TPs), enter STPN=*.

DISPLAY APPC Command

You can also specify the asterisk as the last character of the displayable format of a SNA service transaction program name.

The displayable format of the SNA service transaction program name is in the form:

`-X'hh'ccc`

Where `hh` is the hexadecimal value for the first non-displayable character and `ccc` is a character string (one to three characters) from character set Type A. You can abbreviate the name of the SNA transaction program by entering the first part of the name, followed by an asterisk.

Character sets 00640 and Type A are listed in *z/OS MVS Planning: APPC/MVS Management*.

USERID=userid

The userid of the transaction program that is running because of an allocate request. For an inbound conversation, this is the userid of the local MVS transaction program. For an outbound conversation, this is the userid of the partner transaction program. If you specify this keyword with the TP parameter, the system only displays conversations in which the userid of the allocated transaction program matches the userid you specify. The userid is one to ten alphanumeric (a-z,0-9) or special (@, #, \$) characters.

L=a, name, or name-a

Specifies the display area (`a`), console name (`name`), or both (`name-a`) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Displaying ASCH Configuration Information

Use the DISPLAY ASCH command to display information about the APPC/MVS scheduler configuration (message ASB101I).

```
D ASCH{ [,SUMMARY|,SUM|,S ] [,,{ASID|A}=asid]
      { [,LIST|,L           [,,{CLASS|C}=classname] }
      { [,ALL|,A           [,LTPN=ltpname]          }
      {                   [,QT=sssss[.ttt]]        }
      {                   [,TPST=schedtype]       }
      {                   [,USERID=userid]        }
      [,L={a|name|name-a}]}
```

The parameters are:

SUMMARY or SUM or S

Indicates that the system is to use the SUMMARY format of output. The resulting message contains the number of classes, active and queued transactions, idle initiators, and total number of initiators. It also shows the global settings for the IBM-supplied APPC/MVS transaction scheduler, which are specified in ASCHPMxx parmlib member on the TPDEFAULT and SUBSYS keywords.

LIST or L

Indicates that the system is to use the LIST form of output. This output is the

same as the SUMMARY output, followed by a list of APPC/MVS transaction scheduler classes. Each entry in the list contains the name of a class, along with related information.

ALL or A

Indicates that the system is to use the ALL form of output. This output is the same as the LIST output, except that the system inserts a sublist after each entry in the list. The sublist contains information about each transaction program assigned to the APPC/MVS transaction scheduler class.

Note: For the output of the DISPLAY ASCH command, see the description of message ASB101I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

The following is a list of keywords that filter the displays. When you specify a filter keyword, the system displays only the data that meets the keyword’s criteria.

Notes:

1. The same keyword cannot be used twice with a single command.
2. A command line cannot exceed 126 characters in length.

ASID or A=asid

The address space identifier of the transaction program. The identifier is a one-to four-digit hexadecimal value.

CLASS or C=classname

The name of the APPC/MVS transaction scheduler class. The class name is one to eight alphanumeric (a-z, 0-9) or special (@,#,\$) characters.

LTPN=ltpname

The local transaction program name. This name is one to 64 alphanumeric (a-z, 0-9) or special (@,#,\$) characters. The name may also contain the characters in character set 00640, except for the following:

- comma(,) - used as a keyword delimiter and means that the preceding character is interpreted as the end of the transaction program name
- blank() - interpreted as the end of the command and means that the preceding character is interpreted as the end of the transaction program name
- asterisk(*) - used to filter the local transaction program name, it can only be used as the last character of the name. It causes the LTPN keyword filter to match every transaction program name which begins with the characters preceding the asterisk.

You can abbreviate the name of the local transaction program by entering the first part of the name, followed by an asterisk. For example, PROCESS* matches every local transaction program name that begins with the letters PROCESS.

You can also specify the asterisk as the last character of the displayable format of a SNA service transaction program name.

The displayable format of a SNA service transaction program name is in the form:

`-X'hh'ccc`

Where *hh* is the hexadecimal value for the first non-displayable character and *ccc* is a character string (one to three characters) from character set Type A. You can abbreviate the name of the SNA transaction program by entering the first part of the name, followed by an asterisk.

DISPLAY ASCH Command

Character sets 00640 and Type A are listed in *z/OS MVS Planning: APPC/MVS Management*.

QT=sssss[.ttt]

The queue time, in seconds, of a local transaction program waiting for initiation. *sssss* specifies the number of seconds, from 0 - 99999; *ttt* specifies the number of thousandths of a second, from .0 - .999. When you specify this keyword, the system displays only transaction programs that have been queued for an amount of time greater than or equal to the value you specify.

TPST=schedtype

The scheduling type of the transaction program. This keyword can have values of: STD, STANDARD, MT, or MULTITRANS.

USERID=userid

The userid of the transaction program that is running because of an allocate request. The system displays only initiators that are running programs on behalf of the userid you specify. The userid is one to ten alphanumeric (a-z,0-9) or special (@,#,\$) characters.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Displaying Page Data Set Information

Use the DISPLAY ASM command to identify the page data sets the system is currently using. You can request this information either for each data set of a given type, or for a specific data set.

If you request information by data set type, the following information is displayed in message IEE200I for each data set of the specified type that the system is currently using:

- Type of data set
- Percent full
- Status
- Device number
- Data set name

If you request information about the PLPA or common data set, or about a specific page data set, you receive all the preceding information, plus:

- Volume serial number
- Device type
- Data set size (in slots)
- Number of slots that are currently in use
- Number of slots that are currently available
- Number of permanent I/O errors that have occurred on the specified data set

DISPLAY ASM does not give you the level of detail that you need to tune the paging configuration; for this information, see "Page/Swap Data Set Activity Report" in *z/OS RMF Report Analysis*.

```
D ASM[,PLPA] [,L={a|name}name-a]
      ,COMMON
      ,LOCAL
      ,ALL
      ,PAGE=[dsname|ALL]
      ,PAGEDEL
```

ASM

The system is to display information about the page data sets the system is currently using. If you specify DISPLAY ASM with no operands, the system displays information about all page data sets that it is currently using and the status of the PAGEDEL command.

PLPA

Requests information about the PLPA page data set.

COMMON

Requests information about the common page data set.

LOCAL

Requests information about all local page data sets.

ALL

Requests information about all page data sets, and the status of the PAGEDEL command.

PAGE

Requests information about page data sets.

ALL

Requests information about all page data sets.

dsname

Requests information about the page data set named *dsname*.

PAGEDEL

Requests information about the PAGEDEL command, active or inactive.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display summary information about all page data sets, and the PAGEDEL command status, enter:

DISPLAY ASM,ALL or D ASM

Example 2

To display detailed information about the PLPA data set, enter:

D ASM,PLPA

DISPLAY CEE Command

Displaying the current system level Language Environment run-time options

Use the DISPLAY CEE command to list one or more parmlib members that can create the current system level Language Environment run-time options. The DISPLAY CEE command can also be used to display the contents of the table.

```
D CEE[{},CEEDOPT]      ]
      |,{CEECOPT}
      |,{CELQDOPT}
      [,L={a|name|name-a}]
```

CEEDOPT

Specifies the option to be used in a CICS environment

CEECOPT

Specifies the option to be used in a 31-bit run-time.

CELQDOPT

Specifies the option to be used in a 64-bit run-time.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

Example 1

```
D CEE
CEE3744I hh.mm.ss DISPLAY
CEE=(xx)
```

Where *xx* is the CEEPRM member suffix specified at IPL or with the SET CEE command.

Example 2

```
D CEE,CEEDOPT
CEE3745I hh.mm.ss DISPLAY
CEE=(xx)
```

```
Where set      Option-----
PARMLIB(CEEPRMxx)  ABPERC(NONE)
SETCEE command    ALL31(ON)
...
PARMLIB(CEEPRMxx)  XUFLOW(AUTO)
```

The option listed are only those specified in the SET CEE or SETCEE command.

Displaying CONTROL Command Functions

Use the DISPLAY C,K command to request a summary (message IEE162I) of the CONTROL command operands and the functions they perform.

```
D C,K[,L={a|name|name-a}]
```

C,K

A summary of CONTROL command operands is to be displayed.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display the CONTROL command operands and their functions in display area A of console named CON5, enter:

D C,K,L=CON5-A

Example 2

To display a summary of CONTROL command operands in display area A of console named CON10, enter:

D C,K,L=CON10-A

Displaying Attached Coupling Facility Information

Use the DISPLAY CF command to display storage and attachment information about coupling facilities attached to the system on which the command is processed.

```
| D CF[,CFNAME={(cfname[,cfname]...)}][,L={a|name|name-a}]
```

CF

Requests the system to display information about the coupling facilities that are attached to the system. If specified without further qualification, the system displays information about all coupling facilities that are attached.

CFNAME= or CFNM= cfname

Requests that information for one or more named coupling facilities be displayed.

cfname specifies the logical name of a coupling facility for which information is requested.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Displaying Console Group Definitions

Use this command to display the console group definitions in effect for the sysplex. The definitions, activated via a SET CNGRP command or the INIT statement in CONSOLxx, are obtained from the currently active CNGRPxx parmlib members.

There are three options for this command:

- Display all active console groups with their names.
- Display all console names associated with input group names.
- Display only console group names active in the sysplex.

DISPLAY CNGRP Command

```
D CNGRP[,{GROUP|G} [=({name[,name] ...})]] [,L={a|name|name-a}]
```

CNGRP

The system is to display information (message IEE679I) about the console groups currently defined to the system or sysplex. If you specify this keyword alone, the system displays all the group names and the console names associated with each group.

GROUP or G

The system is to display information on specific console groups. If GROUP is the last keyword in the command, then only the names of all active groups are displayed.

name[,name]

The system is to display all console names associated with each input group name. Valid group names are a maximum of 8 characters long.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To DISPLAY all active console group definitions with their names, enter:

```
DISPLAY CNGRP
```

Example 2

To DISPLAY the names of all active console groups, enter:

```
DISPLAY CNGRP, GROUP  
or  
DISPLAY CNGRP, G
```

Example 3

To DISPLAY the console names associated with the console groups NEWYORK and PHILLY, enter:

```
DISPLAY CNGRP, G=(NEWYORK,PHILLY)
```

Displaying Console Status Information

Use the DISPLAY CONSOLES command to display the status of all consoles or specified consoles in the sysplex, including SMCS. If you need information about extended MCS (EMCS) consoles, use the DISPLAY EMCS command to display information for extended MCS consoles.

See Figure 3-3 on page 3-23 and Figure 3-2 on page 3-18 for examples of the resulting display of the DISPLAY CONSOLES and DISPLAY CONSOLES,BACKLOG commands. The syntax of the command is:

```

D {CONSOLES}, {{ACTIVE|A}[,CA={name }] [,ROUT={NONE|ALL|rr }] }
{C } { {SS } { (name[,name]...) } { ((rr-ss) ) } }
{ {NACTIVE|N} [MSTR] { ((rr[,ss]...)) } }
{ {KEY [=key] } }

{ {BACKLOG|B} }

{ {MASTER|M} [,SYS=system name] }

{ * }

{ {LIST|L} }

{ {HARDCOPY|HC} [,SYS=system name] }

{CN=name [,ROUT={NONE|ALL|rr }] }
{ {((rr-ss) ) } }
{ {((rr[,ss]...)) } }
{ [,SYS=system name] }

{HCONLY }

{U={{ [/]devnum1[,/]devnum2]...}}
{ { { [/]lowdevnum-[/]highdevnum} } }
{ { [/]devnum } }

{MSTR[,SYS=system name] }

{SMCS }

[,L={a|name|name-a}] 
```

The parameters are:

CONSOLES or C

The system is to display console information in message IEE889I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books to see a description of the output.

ACTIVE or A

The system is to display the status of all active MCS and SMCS consoles.

Note: The ACTIVE parameter only works for extended MCS or system consoles when issued with the MSTR parameter. Use the KEY parameter for these consoles.

NATIVE or N

The system is to display the status of all MCS and SMCS consoles that are not active.

Note: The NACTIVE parameter only works for extended MCS and system consoles when issued with the MSTR parameter. Use the KEY parameter for these consoles.

ss

The system is to display the status of all allocatable subsystem consoles.

CA

The system is to display, for a sysplex, the console/system association list to match the specified keyword, ACTIVE or NACTIVE.

DISPLAY CONSOLES Command

name

The system is to display the specified console name(s) in the sysplex to match the specified keyword, ACTIVE or NACTIVE. A console name can be 2 to 8 characters in length.

ROUT

The system is to display the status of all MCS and SMCS consoles that receive messages identified by the routing code you specify.

NONE

The system is to display the status of all MCS and SMCS consoles to which no messages are routed by routing code.

rr The system is to display the status of MCS and SMCS consoles that accept messages with a routing code of *rr*.

(*rr*,*ss*...)

The system is to display the status of MCS and SMCS consoles that accept messages with routing codes listed as *rr*,*ss*, and so forth.

(*rr*-*ss*)

The system is to display the status of MCS and SMCS consoles that accept messages with routing codes in the range of *rr* to *ss*.

ALL

The system is to display the status of MCS and SMCS consoles that accept messages by routing codes.

SYS=*system-name*

The system is to display the status of consoles that are active or eligible to be activated on the system you specify, and that match the other specified parameters.

SYS is mutually exclusive with BACKLOG (B), HARDCOPY (HC), KEY, *, U=, HONLY, and MONLY.

MSTR

The system displays the status of the operators with master authority that match the specified keyword of ACTIVE, NACTIVE, or SS. MSTR is mutually exclusive with CA and U. MSTR can be issued with ROUT.

KEY

The system is to display a list of available class names of extended MCS consoles.

[=key]

The system is to display the list of active operators in the specified console class, where 'key' is a specific class name. See *z/OS MVS Planning: Operations* for more information.

CN=*name*

The system is to display the status of a console or consoles identified by console name.

U=

The system is to display the status of one or more consoles, identified by device numbers. A device number consists of 3 or 4 hexadecimal digits, optionally preceded by a slash (/).

[/]*devnum*

The system is to display the status of the console identified by device number *devnum*.

([/]devnum1,[/]devnum2...)

The system is to display the status of consoles identified by device numbers *devnum1*, *devnum2*, and so on.

([/]lowdevnum-[/]highdevnum)

The system is to display the status of the consoles identified by device numbers in the range of *lowdevnum-highdevnum*.

BACKLOG or B

The system is to display the status of all local consoles with a message backlog. It will list the information in descending order by quantity of backlogged messages.

MASTER or M

The system is to display the status of all consoles with master authority.

- * The system is to display the status of the console that issues the DISPLAY command.

LIST or L

The system is to display the status of consoles defined to the sysplex in CONSOLxx but not for extended MCS consoles.

HONLY

The system is to display information about messages in the hardcopy message set that are not directed to any console in the sysplex.

HARDCOPY or HC

The system is to display the following information about the hardcopy message set or the hardcopy medium:

- Whether the hardcopy medium is SYSLOG or OPERLOG
- Whether the hardcopy message set is to include operator commands, responses, and status displays
- The routing codes for messages the system is to include in the hardcopy message set
- The number of messages waiting to be sent to the hardcopy medium.

The DISPLAY CONSOLES command response (IEE889I) will display SYSLOG and OPERLOG status.

SMCS

Displays the status of the SMCS applications in the sysplex. The SMCS keyword is mutually exclusive with all other DISPLAY CONSOLES keywords.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display information about consoles named CON24 and TAPE, enter:

D C,CN=(CON24,TAPE)

Example 2

DISPLAY CONSOLES Command

To display information about all active consoles that receive messages with routing code 3, enter:

```
D C,A,ROUT=3
```

Example 3

To display information about all subsystem consoles that receive routing code 15, enter:

```
D C,SS,ROUT=15
```

Example 4

To display information about the console device number 81B, enter:

```
D C,U=81B
```

Example 5

To display information about the console device number 3480, enter:

```
D C,U=/3480
```

Example 6

To display the list of available class names (keys), enter:

```
D C,KEY
```

Example 7

To display information about all inactive consoles and have the output go to area A on the console named CON5, enter:

```
D C,N,L=CON5-A
```

Example 8

To display information about hardcopy processing on console CON13, area B, enter:

```
D C,HC,L=CON13-B
```

Displaying DIAG Parmlib Information

Use the DISPLAY DIAG command to display the current options that have been set through DIAGxx parmlib members.

```
D DIAG[,L={a|name|name-a}]
```

DIAG

The system displays information about the current options set in DIAGxx.
(Message IGV007I)

L=a, name, or name-a

Specifies the display area (a), console name (name), or both (name-a) where the display is to appear.

Example 1

To display DIAGxx parmlib information, enter:

D DIAG

Displaying Data Lookaside Facility Information

Use the DISPLAY DLF command to display the names of the data sets that are currently being processed as DLF objects (message ISG343I)

The complete syntax for the DISPLAY DLF command is:

```
D DLF[,RES={({qname|*}[,rname|,*])}][,HEX][,L={a|name|name-a}]
```

The parameters are:

RES=(qname[,rname])

A list of major names or resource information for the specified resource(s). Only resources that have at least one requestor are displayed.

Notes:

1. The recommended DISPLAY DLF syntax is:

DISPLAY DLF,RES=(SYSZSD0,*)

2. The parentheses around the resource name(s) in RES=(qname[,rname]) are required.

A resource name must consist of a qname (major name) and can include an rname (minor name). If you specify an asterisk (*) as the last character in the resource name, then the system treats the name as a generic name; the display includes all resources with names that match the portion of the name specified before the asterisk. For example, SYSV* indicates that set of resources whose names begin with SYSV. If you specify major name without a minor name, the system displays just a list of the specified major names of those resources that have requesters. You can specify a generic qname with a specific rname, and conversely, a specific qname with a generic rname.

Specify the HEX operand if you want the resource names to be displayed in EBCDIC and hexadecimal. Use it when you have resource names that contain characters that will not appear on your console (that is, those characters that are not defined in the figure, “;English (U.S) I/O Interface Code for 3277”, which appears in *IBM 3270 Information Display System*.)

How you specify qname (the major name) depends on the characters in the name.

qname

If qname only contains characters that are alphanumeric (A–Z and 0–9), national (#, @, and \$), and a period (.), specify either:

- 1–8 alphanumeric characters (a specific major name)
- 1–7 alphanumeric characters followed by an asterisk (*) (a generic major name)

'qname'

If qname consists of characters that can be displayed other than alphanumeric, national, or a period (excluding a single quote), use the form ‘qname’. The single quotes are required but do not count as part of the length specification for qname. For qname, specify either:

DISPLAY DLF Command

- 1–8 characters (a specific major name)
- 1–7 characters followed by an asterisk (*) after the closing single quote (a generic major name)

X'qname'

If qname contains hexadecimal values or a single quote, specify the name in hexadecimal in the form X'qname'. The prefix X and the single quotes enclosing qname are required but do not count as part of the length specification for qname. For qname, specify either:

- 2–16 hexadecimal digits (a specific major name)
- 2–14 hexadecimal digits followed by an asterisk (*) after the closing single quote (a generic major name)

- * If you want a list of the major names of all resources that have requesters, specify '*' to indicate a generic major name.

How you specify rname (the minor name) depends on the characters in the name.

rname

If rname contains characters that are alphanumeric (A–Z and 0–9), national (#, @, and \$), and/or a period (.), specify either:

- 1–52 alphanumeric characters (a specific minor name)
- 1–51 alphanumeric characters followed by an asterisk (*) (a generic minor name)

'rname'

If rname consists of characters that can be displayed other than alphanumeric, national, or a period (excluding a single quote), use the form 'rname'. The single quotes are required but do not count as part of the length specification for rname. For rname, specify either:

- 1–52 characters (a specific minor name)
- 1–51 characters followed by an asterisk (*) after the closing single quote (a generic minor name)

X'rname'

If rname contains hexadecimal values or a single quote, specify the name in hexadecimal in the form X'rname'. The prefix X and the single quotes enclosing rname are required but do not count as part of the length specification for rname. For rname, specify either:

- 2–104 hexadecimal digits (a specific minor name)
- 2–102 hexadecimal digits followed by an asterisk (*) after the closing single quote (a generic minor name)

- * If you want information on all resources, specify '*' to indicate a generic minor name.

HEX

Resource information is to be displayed in hexadecimal as well as EBCDIC.

L=a, name, or name-a

Specifies the display area (a), console name (name), or both (name-a) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display resource information about all resources that have requesters, enter:
 D DLF,RES=(*,*)

Performance Implication: This command gives you data about every allocated ENQ/RESERVE resource; therefore, there might be a very large display. If this command produces a large amount of output, the command output might fill WTO buffers, and degrade system response time. If the display exceeds the current supply of WTO buffers, an ABEND 09A with reason code 46FA will occur.

Example 2

To display resource information about all resources whose major name is SYSDSN, enter:

D DLF,RES=(SYSDSN,*)

Example 3

To display in EBCDIC and hexadecimal the outstanding ENQ/RESERVES that have a qname of SYSCTLG, enter:

D DLF,RES=(SYSCTLG,*) ,HEX

The display includes the hexadecimal representation of the resource name, SYSCTLG, with the hexadecimal representation under it:

```
SYSCTLG
EECEDC
2823337
```

Displaying Dump Options or Dump Data Set Status

Use the DISPLAY DUMP command to determine:

- Status and availability of pre- and automatically allocated dump data sets
- What dump mode and options are currently in effect
- The title and error-related data for pre- and automatically allocated dump data sets

```
D {DUMP},{{STATUS|ST|S} } }
{D   } {{OPTIONS|O} } }
{{TITLE|T} {,AUTODSN={aaa|ALL} } }
{{ERRDATA|ER|E} {,DSN={ALL|(ALL)
{nn|(nn[,nn]...)} } }
{nn-nn|(nn-nn[,nn-nn]...)} } }
{(nn[,nn]...,nn-nn[,nn-nn]...)} }

{,DUMPID={xxx|(yyy[,zzz]...)
{aaa-bbb|(aaa-bbb[,ccc-ddd]...)} }
{(yyy[,zzz]...,aaa-bbb[,ccc-ddd]...)} }

[,L={a|name|name-a}] }
```

DUMP or D

The system is to display dump information.

STATUS or ST or S

The system is to display (message IEE852I) a summary of:

DISPLAY DUMP Command

- Which SYS1.DUMP data sets are available and which are full
- How many dumps are captured in virtual storage and how much storage they occupy
- How much virtual storage is available for capturing additional dumps
- The status of automatic dump data set allocation
- What resources are defined for automatic dump data set allocation
- The naming convention currently in effect for automatically allocated dump data sets

OPTIONS or O

The system is to display:

- Coupling facility system failure dumping options
- Dump mode and options in effect for dump types:
 - SDUMP
 - SYSUDUMP
 - SYSMDUMP
 - SYSABEND

TITLE or T

The system is to display (message IEE853I) the dump data set name, title, and time of the dump for the captured dumps or dumps written to pre- or automatically allocated dump data sets as requested by the AUTODSN=, DSN=, and DUMPID parameters.

ERRDATA or ER or E

The system is to display (message IEE854I) error data for:

- Full direct access dump data sets that you specify in DSN=
- Automatically allocated dump data sets specified in AUTODSN=
- Captured dumps that you specify on the DUMPID parameter.

The error data for each full data set includes:

- Dump title
- Data set names for automatically allocated dump data sets
- Time of dump
- Error id, which includes the sequence number, the processor id, the ASID of the failing task, and the time stamp
- Abend code
- Reason code
- Module name
- Failing CSECT name
- Error PSW
- Translation exception address
- Abending program address
- Recovery routine address
- Registers at time of error

The SDWA furnishes most of the data that appears in the display caused by the DISPLAY DUMP,ERRDATA command. This means that if MVS/ESA is not in recovery mode, the display contains only the data set name, title, and time of the dump.

AUTODSN={aaa or ALL}

The system is to display the requested dump information about the dump data sets that were most recently allocated automatically. Only those dump data sets

allocated since the last IPL will be presented. Dump data sets created in a previous IPL will not be displayed by this command.

aaa

Specifies the number of data sets for which information is displayed. *aaa* must have a value from 1 to 100.

ALL

The system displays information for all dump data sets that were automatically allocated to a maximum of one hundred.

AUTODSN=, DSN=, and DUMPID= are mutually exclusive.

DSN={ALL or nn}

The system is to display dump information about all direct access dump data sets or specific pre-allocated direct access dump data sets. For DSN= you can specify:

- All data sets
- One or more single data sets
- One or more ranges of data sets

For specific data sets or ranges, **nn** must have a value from 00 to 99. When you specify a range of data sets, the first **nn** in the range must be less than or equal to the second **nn**.

If you specify **DSN=ALL**, then the system also displays information about the one dump data set that was most recently allocated automatically.

DSN=, AUTODSN=, and DUMPID= are mutually exclusive.

DUMPID=xxx or aaa- ddd

The system is to display the dump information about specific captured dumps waiting to be written to dump data sets, as denoted by the three decimal digit DUMPID. You can specify for DUMPID= one or more single captured dump identifiers and/or one or more ranges of captured dump identifiers. For any of these specifications, the value must be in the range of 000 to 999. When you specify a range of captured dump identifiers, the first identifier must be less than the second identifier. Multiple identifiers or ranges must be enclosed in parentheses and separated by commas.

DSN=, AUTODSN=, and DUMPID= are mutually exclusive.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To check the full or available status of all defined SYS1.DUMP data sets on both direct access and tape devices, enter:

D DUMP

Example 2

To display the dump title for direct access dump data sets 1, 5, 6, 7, 8, 9, 10, and 90, enter:

D D,T,DSN=(01,05-10,90)

DISPLAY DUMP Command

If any of these data sets are empty or undefined, the system tells you that dump data is not available for them.

Example 3

To display error data for all full direct access dump data sets *and* the most recent automatically allocated dump data set, enter:

```
D D,ER,DSN=ALL
```

Example 4

To display error data, if any, for direct access dump data sets 1-21, enter:

```
D D,ER,DSN=(01-21)
```

If any of these data sets are empty or undefined, the system tells you that dump data is not available for them.

Example 5

To see the dump modes and dump options in effect for each dump type, enter:

```
D D,O
```

Example 6

To DISPLAY the error data for captured dump 123, enter:

```
D D,ER,DUMPID=123
```

Example 7

To DISPLAY the titles of captured dumps 123 and 456, enter:

```
D D,T,DUMPID=123,456
```

Example 8

To DISPLAY the titles of all, or the one hundred most recent, automatically allocated dump data sets, enter:

```
D D,T,AUTODSN=ALL
```

Displaying Extended MCS Information

Use the DISPLAY EMCS command (instead of the DISPLAY CONSOLES command) to display information about extended MCS (EMCS) consoles.

When the system searches for any consoles you specify, it allows wildcard matching. CN, SYS, and KEY can include wildcard characters (*) and (?) that allow a single parameter to match many different actual conditions. For example, CN=AD? matches console names like AD1 or AD2 but not ADD1. CN=A* matches A1 or AD1 or ADD1.

The syntax for the DISPLAY EMCS command is:

```
D EMCS,{SUMMARY|S
          {INFO|I
          {FULL|F
          }[,L={a|name|name-a}]
          }{STATUS=A|N|L|B[{nn}]|ERR
          {ST
          }{CN=consname|*
          }{SYS=sysname|*
          }{KEY=keyname|*
          }{AUTH={ANY}
              {MASTER}
              {SYS}
              {IO}
              {CONS}
              {ALL}
              {INFO}
              {SYSONLY}
              {IOONLY}
              {CONSONLY}
              {ALLONLY}
              {INFOONLY}
              }{ATTR={ANY}
                  {YES}
                  {ROUT}
                  {HC}
                  {AUTO(YES|NO)}
                  {MN}
                  {NONE}
                  {INTIDS}
                  {UNKNIDS}
                  }{DOM={ANY}
                      {NORMAL}
                      {ALL}
                      {NONE}
                      {YES}
                      }
```

The parameters are:

EMCS

The system is to display console information about extended MCS (EMCS) consoles.

SUMMARY or S

The system is to display only the numbers and names for the consoles that meet the criteria.

INFO or I

The system is to display all console information, except statistics on the console's message data space, for the consoles that meet the criteria.

FULL or F

The system is to display all available information about the consoles that meet the criteria. Message data space statistics can only be displayed for consoles that are active on the system where the command is processed.

DISPLAY EMCS Command

The following keyword parameters define the criteria used to limit the number of consoles displayed.

STATUS=A|N|LIB[(*nn*)]|ERR

The system is to display information about extended MCS consoles according to console status:

- A** All extended MCS consoles that are active.
- N** All extended MCS consoles that are not active.
- L** Both active and inactive extended MCS consoles.

B[(*nn*)**]**

All consoles with a backlog of more than *nn* unretrieved delivered messages, where *nn* is a number from 1 to 999999. If you omit *nn*, the default is 10 unretrieved messages. Backlog information can only be displayed for consoles attached to the system that processes the command.

ERR

All consoles in an error state, such as consoles with queueing suspended. Error state information can only be displayed for consoles attached to the system that processes the command.

Note: Specifying B or ERR on STATUS forces the amount of information to be FULL.

CN=consname

The system is to display information according to console name. A console name can be from 1 to 8 characters. You can specify wildcard characters (*) and (?) in the console name.

CN=* is a special case because * is not a wildcard character. CN=* means that the system is to display information about this console, the console you are using to enter the command.

Note: Specifying CN=*, or a console name with no wildcard characters, automatically forces STATUS=L.

SYS=sysname

The system is to display information about any consoles that are active or eligible to be activated on the system you specify, and that match the other specified parameters. A system name can be from 1 to 8 characters. You can specify wildcard characters (*) and (?). The default is SYS=*, which matches all system names.

KEY=keyname

The system is to display information according to console key name, where *keyname* is the name your installation has assigned to a console group. (See *z/OS MVS Planning: Operations* for more information.) The name can be from 1 to 8 characters. You can specify wildcard characters (*) and (?). The default is KEY=*, which matches all console class names.

AUTH=

The system is to display information about consoles with a specific console command authority, which may be one of the following:

ANY

Consoles with any authority.

MASTER

Consoles with MASTER authority.

SYS

Consoles with at least SYS authority (meaning MASTER authority, SYS authority alone, or SYS combined with IO or CONS or both).

IO Consoles with at least IO authority.**CONS**

Consoles with at least CONS authority.

ALL

Consoles with at least ALL authority (SYS, IO, and CONS).

INFO

Consoles with at least INFO authority.

SYSONLY

Consoles with SYS authority only (not MASTER, CONS, or IO).

IOONLY

Consoles with IO authority only.

CONSONLY

Consoles with CONS authority only.

ALLONLY

Consoles with ALL authority only (meaning consoles with SYS, IO, and CONS authority, but not MASTER authority).

INFOONLY

Consoles with INFO authority only.

ATTR=

The system is to display information about extended MCS consoles that receive messages with a specific routing attribute, which may be one of the following:

ANY

Any consoles, regardless of routing attributes.

YES

Consoles that receive some type of unsolicited messages (either routing codes, hardcopy messages, AUTO(YES) messages, or MONITOR messages.)

ROUT

Consoles that receive any routing codes.

HC

Consoles receiving the hardcopy message set.

AUTO[(YES|NO)]

Consoles that are or are not receiving AUTO(YES) messages. The default is YES.

MN

Consoles receiving any type of MONITOR messages.

NONE

Consoles with no routing attributes.

INTIDS

Consoles receiving messages directed to console ID zero. If you do not specify, the system displays consoles with either Y or N value of INTIDS.

|
|
|

DISPLAY EMCS Command

UNKNIDS

Consoles receiving messages directed to the old 1-byte format of console IDs. If you do not specify, the system displays consoles with either Y or N value of UNKNIDS.

DOM=

The system is to display information about extended MCS consoles according to specific DOM attributes.

ANY

Any consoles, regardless of DOM attributes.

NORMAL

Only consoles defined with DOM(NORMAL).

ALL

Only consoles defined with DOM(ALL).

NONE

Only consoles defined with DOM(NONE).

YES

Consoles defined with either DOM(ALL) or DOM(NORMAL).

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example

Assume a single system where the system console is named SYS01, and there are two additional extended MCS consoles, named EMCS1 and EMCS2. If you issue the following command:

```
DISPLAY EMCS
```

The display includes only the console names (because SUMMARY is the default) of active extended MCS consoles (because STATUS=A is the default). The output of the command appears as follows:

```
IEE129I 13.35.15
DISPLAY EMCS DISPLAY EMCS
NUMBER OF CONSOLES MATCHING CRITERIA: 6
*DICNSY3 EMCS1 SY3 *ROUTEY3 EMCS2 *SYSLGY3
DISPLAY EMCS,INFO
```

The display includes all information except dataspace information (because you specified INFO) but will only include active consoles (because STATUS=A is the default). The output of the command looks like:

```
CNZ4101I 13.31.07 DISPLAY EMCS
DISPLAY EMCS,INFO
NUMBER OF CONSOLES MATCHING CRITERIA: 6
CN=*DICNSY3 STATUS=A CNID=01000001 KEY=None
SYS=SY3 ASID=0009 JOBNAME=----- JOBID=-----
HC=N AUTO=N DOM=ALL TERMNAME=*DICNSY3
MONITOR=-----
CMDSYS=SY3
LEVEL=ALL AUTH= INFO
MSCOPE=*ALL
```

```

ROUTCDE=ALL
INTIDS=N UNKNIDS=N
CN=EMCS1 STATUS=A CNID=01000002 KEY=EXAMPLE
SYS=SY3 ASID=0018 JOBNAME=EXTMCS JOBID=STC00029
HC=N AUTO=N DOM=NORMAL TERMNAME=EMCS1
MONITOR=-----
CMDSYS=SY3
LEVEL=ALL AUTH= MASTER
MSCOPE=*ALL
ROUTCDE=ALL
INTIDS=N UNKNIDS=N
CN=SY3 STATUS=A CNID=02000001 KEY=SYSCONS
SYS=SY3 ASID=0009 JOBNAME=----- JOBID=-----
HC=N AUTO=N DOM=NORMAL TERMNAME=SY3
MONITOR=-----
CMDSYS=SY3
LEVEL=ALL,NB AUTH= MASTER
MSCOPE=*ALL
ROUTCDE=NONE
AUTOACT=-----
INTIDS=N UNKNIDS=N
CN=*ROUTEY3 STATUS=A CNID=02000002 KEY=MVSROUTE
SYS=SY3 ASID=0009 JOBNAME=----- JOBID=-----
HC=N AUTO=N DOM=NONE TERMNAME=ROUTEALL
MONITOR=-----
CMDSYS=SY3
LEVEL=R,NB AUTH= CONS
MSCOPE=*ALL
ROUTCDE=NONE
INTIDS=N UNKNIDS=N
CN=EMCS2 STATUS=A CNID=02000003 KEY=EXAMPLE
SYS=SY3 ASID=0029 JOBNAME=EXTMCS JOBID=STC00030
HC=N AUTO=N DOM=NORMAL TERMNAME=EMCS2
MONITOR=-----
CMDSYS=SY3
LEVEL=ALL AUTH= MASTER
MSCOPE=*ALL
ROUTCDE=ALL
INTIDS=N UNKNIDS=N
CN=*SYSLGY3 STATUS=A CNID=03000001 KEY=SYSLOG
SYS=SY3 ASID=0009 JOBNAME=----- JOBID=-----
HC=N AUTO=N DOM=NONE TERMNAME=*SYSLGY3
MONITOR=-----
CMDSYS=SY3
LEVEL=ALL AUTH= MASTER
MSCOPE=SY3
ROUTCDE=NONE
INTIDS=N UNKNIDS=N

```

Displaying the Timer Synchronization Mode and ETR Ports

Use the DISPLAY ETR command to display the current timer synchronization mode and the status of the ETR ports as seen by MVS.

The DISPLAY ETR command issues the following information when the sysplex is running in STP mode:

- Timing mode
- CTN ID
- Time
- Node that is the source of the time, if applicable
- Redundant available timing links, if applicable

DISPLAY ETR Command

- | • Stratum level of the server
- | See message IEA386I for additional details.

The complete syntax for the DISPLAY ETR command is:

```
D ETR[,DATA] [,L={a|name|name-a}]
```

ETR

Displays the current ETR (external time reference) synchronization and the status of the ETR ports.

DATA

Displays the status, in detail, of each ETR port, giving the ETR network ID, ETR port number, and the ETR ID.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example

To display the current timer synchronization mode status and the ETR ports, enter:

```
D ETR
```

The status is shown in this display:

```
IEA282I  
hh.mm.ss ETR STATUS SYNCHRONIZATION MODE=mode CPC SIDE=id  
CPC PORT 0 CPC PORT 1  
op op  
enb enb
```

where the fields in the message are:

hh.mm.ss	is the current time in hours(hh), minutes(mm), seconds(ss)
MODE=mode	is the current synchronization mode, ETR or LOCAL
CPC SIDE=id	is the current CPC side id, 0 or 1
op	is the status of the port, operational or nonoperational
enb	is the status of the port, enabled or disabled

Displaying Global Resource Serialization Information

Use the DISPLAY GRS command to display information that helps you control the global resource serialization complex. The information includes:

- Configuration information:

The status of each system associated with the current global resource serialization complex is displayed. Some of the information displayed depends on whether you are running a global resource serialization ring or star complex. For instance, the status of the systems in the complex and the manner in which the systems are connected are different for a ring and a star complex.

- RNL information:
The contents of the RESERVE conversion, SYSTEMS exclusion, and SYSTEM inclusion resource name lists (RNLs).
- Resource information:
Information on resources for which there is contention or information about a specific resource.
- Information on resources that are delaying or suspending RNL changes.

If you do not code any keywords on the DISPLAY GRS command:

- The information displayed for a star complex is the same as if you entered DISPLAY GRS,SYSTEM.
- The information displayed for a ring complex is the same as if you entered DISPLAY GRS,SYSTEM, plus the configuration is displayed.
- Contention information:
A list of the units of work involved in contention for GRS-managed resources. This display can focus on the units of work that are waiting for the resources or those blocking the resources. In addition, the installation can display dependencies between requesters of GRS-managed resources.

When the keywords CONTENTION, RES=, RNL=, or any combination of them are specified together, the system will display a separate section for each keyword specified. Specifying the ALL keyword will override any parameters specified on the RNL= keyword.

The complete syntax for the DISPLAY GRS command is:

```
D GRS{[,SYSTEM|,LINK|,ALL|,A|,DELAY|,D|,SUSPEND|S|}
      [,CONTENTION|,C]
      [,RES=(qname|*[,rname|,*])]
      [,HEX]
      [,DEV=[/]devnum [,,{SUSPEND|S}]]
      [,RNL={CONVERSION|CON|C}]
          {ALL|A}
          {EXCLUSION|EXCL|E}
          {INCLUSION|INCL|I}

      ,{CONTENTION|C[,ENQ|,E] [,,{LATCH|L} [,,{JOBNAME|JOB}=jobname]] [,,HEX]}
      ,{,{LATCH|L} [,,{JOBNAME|JOB}=jobname] [,CONTENTION|,C]} [,,HEX]
          {CONTENTION|C}

      ,{ANALYZE|,ANALYSE|,AN,
          {BLOCKER|BLOCK
          {WAITER|WAIT[,SYSTEM|,SYS=sysname|*]
              [,ASID|,AS=asid]
              [,JOBNAME|,JOB=jobname]
              [,XJOBNAME|,XJOB=(jobname1,jobname2,...,jobname25)]
              [,XQNAME|,XQN=(qname1,qname2,...,qname25)]}
          }

      {DEPENDENCY|DEPEND|DEP[,SYSTEM|,SYS=sysname|*]
          [,ASID|,AS=asid][TCB=tcbaddr]
          [,JOBNAME|,JOB=jobname]
          [,XJOBNAME|,XJOB=(jobname1,jobname2,...,jobname25)]
          [,XQNAME|,XQN=(qname1,qname2,...,qname25)]}

      [,RES=(qname,rname)]
          {SCOPE|SCO=SYSTEM[S]|SYS[S]}
          {SYSTEM|SYS=sysname|*}
          [,COUNT|,CNT=nn][,DETAIL|,DET]

      [,L={a|name|name-a}]}
```

DISPLAY GRS Command

SYSTEM

System information is to be displayed. The SYSTEM operand produces a display only when a global resource serialization complex is active.

The display of system information includes:

- The system name (the name specified on the SYSNAME system parameter) of each system in the global resource serialization complex.
- The state of each system in the global resource serialization star complex:
 - Connecting: The system is processing the GRS=STAR parameter. It is not yet a member of the global resource serialization star complex.
 - Connected: The system is a member of the global resource serialization star complex.
 - Rebuilding: The system is a member of the global resource serialization star complex, but is currently rebuilding the global resource serialization lock structure, ISGLOCK. The system suspends any tasks that try to obtain any global resources.
- The state of each system in the global resource serialization ring complex:
 - Active: The system is presently serializing global resources. It is a member of the global resource serialization ring. If ACTIVE and WAIT appear, a global resource serialization command was issued but is waiting because another global resource serialization command is executing. If ACTIVE and VARY appear, the system is executing an internally-issued or operator-issued global resource serialization command.
 - Inactive: This system is not presently sending or receiving global requests in the global resource serialization ring. Any requests already held remain held, and any new requests remain pending until this system restarts back into the ring. The system suspends tasks that request global resources. The system can be used to restart the ring.
 - Quiesced: This system is not presently sending or receiving global requests in the global resource serialization ring. Any requests that are held remain held, and any new requests remain pending until this system restarts back into the ring. The system suspends tasks that request global resources. The system must restart back into an already active ring to resume global request processing, or it can be reactivated to restart a new ring if no other active systems exist.
 - Joining: This system is in the process of joining the global resource serialization complex.
 - Restarting: This system was an inactive or quiesced system and is now in the process of rejoining the global resource serialization ring.
 - Migrating: The system is participating in the process of migrating from a global resource serialization ring complex to a global resource serialization star complex.
- The communication status of each system in the global resource serialization ring or star complex:
 - The display for a ring complex shows the current settings for the following parameters:
 - The minimum RSA-message residency time in milliseconds.
 - The maximum tolerance time interval allowed for RSA-message to return to the system.
 - Whether synchronous reserve processing is activated for the current system.

- The display for a star complex shows the current settings for the following parameters:
 - The number of locks for the global resource serialization lock structure (ISGLOCK).
 - Which system is the contention notifying system.
 - Whether synchronous reserve processing is activated for the current system.

LINK

CTC link information is to be displayed. The LINK operand produces a display only when a global resource serialization complex is active.

When the global resource serialization ring complex is using XCF signalling, the display shows that XCF paths are used.

The display of global resource serialization CTC link information includes:

- The device number of each CTC link assigned to global resource serialization on this system
- The status of each global resource serialization CTC link, which can be any one of the following:
 - In use: This system is using this link to send messages to and receive RSA messages from the system at the other end of the CTC link.
 - Alternate: This system is not presently using this link to send or receive RSA messages, but it has the ability to do so. If your installation uses ring acceleration, global resource serialization might be using the link to send the ring acceleration signal.
 - Disabled: This system cannot use this link to send or receive messages.
 - Quiet: The system at the opposite end of the link does not respond.
- The system name (the name specified on the SYSNAME system parameter) of the system that last responded from the opposite end of the link.

ALL or A

The following information is to be displayed:

- System information
- CTC link information
- Resource contention information
- RNL change information
- The contents of all RNLs

Specify HEX if you want the resource names displayed in EBCDIC and in hexadecimal. If the global resource serialization complex is inactive, the display does not contain system information, CTC link information, or the contents of RNLs. If the system did not start or join an active global resource serialization ring at IPL, the display does not contain system, link, and RNL information. When an RNL change is in progress, the display shows DELAY and SUSPEND information.

CONTENTION or C

Resource contention information for the current global resource serialization complex is to be displayed. If a DISPLAY GRS,CONTENTION command is entered without the LATCH or ENQ operands, the system displays both ENQ and LATCH contention information. Specify HEX if you want resource names displayed in EBCDIC and in hexadecimal.

DELAY or D

Displays the jobs that are delaying an RNL change. The following information is displayed:

DISPLAY GRS Command

- Jobname
- ASID
- Resources held or waiting for

DEV=[/]devnum

Displays a list of non-converted RESERVE requests for the device identified by *devnum*. The device number consists of 3 or 4 hexadecimal digits, optionally preceded by a slash (/).

You can use this command to:

- Determine which jobs have RESERVE requests for a particular device.
- Whether this system has reserved the device.
- Help resolve reserve contention problems.

ENQ or E

Resource contention information for ENQs and RESERVEs is to be displayed. If you specify ENQ, you must also specify CONTENTION.

JOBNAME or JOB

Latch information for a particular job is to be displayed. Specify the name of a job that you suspect either owns a latch or has a pending request to obtain a latch. If JOBNAME is specified, LATCH must also be specified.

LATCH or L

Latch information is to be displayed. If you specify LATCH, you must also specify CONTENTION or JOBNAME or both:

- Specify LATCH,CONTENTION if you suspect that tasks or SRB routines are contending for a latch that is held by another task or SRB routine. The system displays information about latches that have contention (if any exist).
- Specify LATCH,JOBNAME to display information about latches that a specified job owns or is waiting for, regardless of whether contention for those latches exists.
- Specify LATCH,CONTENTION,JOBNAME to display information about latches that a specified job owns or is waiting for and for which contention exists.

The following are the various combinations of the CONTENTION, ENQ, JOBNAME, and LATCH parameters, and the information each produces:

CONTENTION

ENQ,LATCH,CONTENTION

Both ENQ and latch contention information.

ENQ,CONTENTION

ENQ contention information.

LATCH,CONTENTION

Latch contention information.

LATCH,JOBNAME

Latch information for a particular job name, if the job owns or waits for a latch

LATCH,CONTENTION,JOBNAME

Latch information for a particular job name, if the job owns or waits for a latch, and contention exists for the latch.

ENQ,LATCH,CONTENTION,JOBNAME

ENQ and latch contention information for a particular job name.

SUSPEND or S

Displays the jobs that are suspended, waiting for resources that are affected by the RNL change. The following information is displayed:

- Jobname
- ASID
- Resources the job is waiting for

Note: DELAY and SUSPEND do not support the HEX operand.

RNL=

The contents of one or all resource name lists (RNLs) in the current global resource serialization complex are to be displayed. The RNL operand produces a display only when a global resource serialization complex is active. Specify HEX if you want resource names displayed in EBCDIC and in hexadecimal. When the GRSRNL=EXCLUDE option is in effect, the display shows this option is being used.

CONVERSION or CON or C

The contents of the RESERVE conversion RNL are to be displayed.

EXCLUSION or EXCL or E

The contents of the SYSTEMS exclusion RNL are to be displayed.

INCLUSION or INCL or I

The contents of the SYSTEM inclusion RNL are to be displayed.

ALL or A

The contents of all RNLs are to be displayed.

RES=(qname[,rname])

A list of major names or resource information for the specified resource(s). Only resources that have at least one requestor are displayed.

A resource name must consist of a qname (major name) and can include an rname (minor name). If you specify an asterisk (*) as the last character in the resource name, then the system treats the name as a generic name; the display includes all resources with names that match the portion of the name specified before the asterisk. For example, SYSV* indicates that set of resources whose names begin with SYSV. If you specify major name without a minor name, the system displays just a list of the specified major names of those resources that have requesters. You can specify a generic qname with a specific rname, and conversely, a specific qname with a generic rname.

Specify the HEX operand if you want the resource names to be displayed in EBCDIC and hexadecimal. Use it when you have resource names that contain characters that will not appear on your console (that is, those characters that are not defined in the figure, “;English (U.S) I/O Interface Code for 3277,” which appears in *IBM 3270 Information Display System*).

Note: The parentheses around the resource name(s) in RES=(qname[,rname]) are required.

How you specify qname (the major name) depends on the characters in the name.

qname

If qname contains only characters that are alphanumeric (A-Z and 0-9), national (#, @, and \$), and a period (.), specify either:

- 1-8 alphanumeric characters (a specific major name)
- 1-7 alphanumeric characters followed by an asterisk (*) (a generic major name)

DISPLAY GRS Command

'qname'

If qname consists of characters that can be displayed other than alphanumeric, national, or a period (excluding a single quote), use the form 'qname'. The single quotes are required but do not count as part of the length specification for qname. For qname, specify either:

- 1-8 characters (a specific major name)
- 1-7 characters followed by an asterisk (*) after the closing single quote (a generic major name)

X'qname'

If qname contains hexadecimal values or a single quote, specify the name in hexadecimal in the form X'qname'. The prefix X and the single quotes enclosing qname are required but do not count as part of the length specification for qname. For qname, specify either:

- 2-16 hexadecimal digits (a specific major name)
- 2-14 hexadecimal digits followed by an asterisk (*) after the closing single quote (a generic major name)

- * If you want a list of the major names of all resources that have requesters, specify '*' to indicate a generic major name.

How you specify rname (the minor name) depends on the characters in the name.

rname

If rname contains characters that are alphanumeric (A-Z and 0-9), national (#, @, and \$), and/or a period (.), specify either:

- 1-52 alphanumeric characters (a specific minor name)
- 1-51 alphanumeric characters followed by an asterisk (*) (a generic minor name)

'rname'

If rname consists of characters that can be displayed other than alphanumeric, national, or a period (excluding a single quote), use the form 'rname'. The single quotes are required but do not count as part of the length specification for rname. For rname, specify either:

- 1-52 characters (a specific minor name)
- 1-51 characters followed by an asterisk (*) after the closing single quote (a generic minor name)

X'rname'

If rname contains hexadecimal values or a single quote, specify the name in hexadecimal in the form X'rname'. The prefix X and the single quotes enclosing rname are required but do not count as part of the length specification for rname. For rname, specify either:

- 2-104 hexadecimal digits (a specific minor name)
- 2-102 hexadecimal digits followed by an asterisk (*) after the closing single quote (a generic minor name)

- * If you want information on all resources, specify (*) to indicate a generic minor name.

HEX

Resource information is to be displayed in hexadecimal as well as EBCDIC. The HEX operand does not affect the SYSTEM or LINK operands.

ANALYZE or ANALYSE or AN

Displays an analysis of system contention. The scope of the analysis is based on the input specified by the command issuer, and can be:

- the entire sysplex

- one system
- one address space
- one task.

The default scope for the analysis is the entire sysplex.

BLOCKER or BLOCK

Displays an ordered list of the units of work blocking GRS-managed resources.

WAITER or WAIT

Displays an ordered list of the units of work waiting for ownership of GRS-managed resources.

DEPENDENCY or DEPEND or DEP

Displays the dependencies between the units of work and resources that are in contention.

SYSTEM or SYS = sysname or *

The scope of the analysis, where sysname is a valid system name in the sysplex, 1–8 characters, following sysname rules.

ASID or AS = asid

A valid hexadecimal address space id, 1–4 hexadecimal digits. ASID requires specification of a valid sysname; you may NOT specify ASID with JOBNAME or XJOBNAME..

JOBNAME or JOB = jobname

A valid JOBNAME, 1–8 characters, following jobname rules. JOBNAME requires specification of a valid sysname; you may NOT specify JOBNAME with ASID or XJOBNAME.

XJOBNAME or XJOB = (jobname1,jobname2,...,jobname25)

Jobname(s) to exclude from the analysis. You may specify up to 25 jobnames, and may enter an asterisk as a wildcard indicator as the last character of a jobname. You may NOT specify XJOBNAME with JOBNAME or ASID. If you specify XJOBNAME with XQNAME, exclusion will occur if either the jobnames or qnames match.

XQNAME or XQN = (qname1,qname2,...qname25)

Qname(s) (major names) to exclude from the analysis. You may specify up to 25 qnames, and may enter an asterisk (*) as a wildcard indicator as the last character of a qname. You may not specify XQNAME with RES. If you specify XQNAME with XJOBNAME, exclusion will occur if either the qnames or jobnames match.

TCB=tcbaddr

A valid TCB address, 1–8 hexadecimal digits. TCB is only valid with DEPENDENCY, SYSTEM, and ASID or JOBNAME, and requires specification of a valid ASID or jobname.

RES=(qname,rname)

The resource name with which to begin a dependency analysis. A resource name consists of a *qname* (major name) and an *rname* (minor name) of the 'nn' (see COUNT=) longest owners of the resource. If you omit the RES= keyword, the dependency analysis will display the *nn* longest waiters without regard to resource. You may not specify RES with XQNAME.

Note: When the ANALYZE keyword is specified, the HEX operand is not available for this keyword. *qname* indicates the resource major name and *rname* indicates the resource minor name with which to begin analysis. The valid characters are \$, ., @, and #.

DISPLAY GRS Command

SCOPE or SCO = SYSTEM(S) or SYS(S)

Indicates the scope of the resource that begins the dependency analysis. If you specify SCOPE=SYSTEM, then you must also specify with which system to associate the ENQ. You do this by using the SYSTEM keyword.

COUNT or CNT = nn

The maximum number of blockers or waiters (in decimal) to display. Valid values are from 1 to 99. The default value is 10. COUNT is valid with any combination of keywords.

DETAIL or DET

Specifies the more detailed form of message ISG349I. Not specifying DETAIL gives the shorter form of the output.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display resource information about all resources that have requesters, enter:

```
D GRS,RES=(*,*)
```

Performance Implication: This command gives you data about every allocated ENQ/RESERVE resource on all systems in the global resource serialization complex; therefore, there might be a very large display. If this command produces a large amount of output, the command output might fill WTO buffers, and degrade system response time. If the display exceeds the current supply of WTO buffers, an ABEND 09A with reason code 46FA will occur.

Global resource serialization might also truncate information that is displayed with ISG343I:

```
ISG343I 15.08.31 GRS STATUS  
NOT ENOUGH STORAGE TO COMPLETE THE REQUEST
```

Example 2

To display resource information about all resources whose major name is SYSDSN, enter:

```
D GRS,RES=(SYSDSN,*)
```

Example 3

To display the jobs delaying an RNL change, enter:

```
D GRS,DELAY
```

The system displays the following message:

```
ISG343I 01:32:21 GRS STATUS  
    DELAY REASONS ON SYSTEM SYSA  
    JOBNAME      ASID   QNAME   RNAME  
    USERJB1      0044   SYSDSN  SYS1.XYZ  
    CATALOG      0007   SYSDSN  SYS1.CHANGING.RESOURCE2
```

Example 4

To display the jobs suspended waiting for resources affected by an RNL change, enter:

```
D GRS,SUSPEND
```

The system displays the following message:

```
ISG343I 01:32:21 GRS STATUS
JOBS BEING SUSPENDED DUE TO RNL CHANGE ON SYSTEM SYSA
  JOBNAME      ASID  QNAME    RNAME
  USERNM3       0089  SYSDSN   SYS1.DATASET.CHANGING
  USERNM4       0245  SYSDSN   SYS1.XYZ.ABC
```

Example 5

To display in EBCDIC and hexadecimal the outstanding ENQ/RESERVES that have a qname of SYSCTLG, enter:

```
D GRS,RES=(SYSCTLG,*) ,HEX
```

The display includes the hexadecimal representation of the resource name, SYSCTLG, with the hexadecimal representation under it:

```
SYSCTLG
EEECEDC
2823337
```

To display in EBCDIC and hexadecimal the outstanding ENQ/RESERVES of a resource with a minor name, enter:

```
D GRS,RES=(GRJGQE07,*) ,HEX
```

The hexadecimal representation of the resource, GRJGQE07, would look like the following:

```
ISG343I 15.17.06 GRS STATUS
S=SYSTEMS GRJGQE07 TESTCASE#SERIALIZATION#ENQ
  CDDCDCFF ECEECCEC7ECDCDCECECDD7CDD
  79178507 35233125B2599139913965B558
SYSNAME      JOBNAME          ASID     TCBADDR  EXC/SHR  STATUS
S1           MAINAS11        001F     007DEB90 EXCLUSIVE  OWN
```

Example 6

To display latch contention information for all latches that currently exist, enter the following:

```
D GRS,LATCH,CONTENTION
```

If latch contention exists, the system displays the following message:

```
ISG343I 23.00.04 GRS LATCH STATUS 886
LATCH SET NAME: MY.FIRST.LATCHSET
CREATOR JOBNAME: APPINITJ CREATOR ASID: 0011
  LATCH NUMBER: 1
    REQUESTOR ASID  EXC/SHR  OWN/WAIT
    MYJOB1    0011  EXCLUSIVE  OWN
    DATAchg   0019  EXCLUSIVE  WAIT
    DBREC     0019  SHARED    WAIT
  LATCH NUMBER: 2
    REQUESTOR ASID  EXC/SHR  OWN/WAIT
    PEEKDAT1  0011  SHARED    OWN
    PEEKDAT2  0019  SHARED    OWN
    CHGDAT   0019  EXCLUSIVE  WAIT
LATCH SET NAME: SYS1.FIRST.LATCHSET
CREATOR JOBNAME: INITJOB2 CREATOR ASID: 0019
  LATCH NUMBER: 1
```

DISPLAY GRS Command

```
REQUESTOR ASID   EXC/SHR   OWN/WAIT
MYJOB2    0019    SHARED     OWN
LATCH NUMBER: 2
REQUESTOR ASID   EXC/SHR   OWN/WAIT
TRANJOB1  0019    SHARED     OWN
TRANJOB2  0019    EXCLUSIVE  WAIT
```

If no latch contention exists, the system displays the following message:

```
ISG343I 23.00.04 GRS LATCH STATUS 886
NO LATCH CONTENTION EXISTS
```

Example 7

To display latch information for job LISTCHK, enter:

```
D GRS,L,JOB=LISTCHK or D GRS,LATCH,JOBNAME=LISTCHK
```

If job LISTCHK owns or waits for a latch, the system displays the following message:

```
ISG343I 23.00.04 GRS LATCH STATUS 886
LATCH DISPLAY FOR JOB LISTCHK
LATCH SET NAME: SYS2.PAYROLLAPP.LATCHSET
CREATOR JOBNAME: INITJOB1  CREATOR ASID: 0011
LATCH NUMBER: 1
REQUESTOR ASID   EXC/SHR   OWN/WAIT
GETDAT1  0011    EXCLUSIVE  OWN
GETDAT2  0019    EXCLUSIVE  WAIT
SHOWDAT  0019    SHARED     WAIT
LATCH NUMBER: 2
REQUESTOR ASID   EXC/SHR   OWN/WAIT
LISTREC   0011    SHARED     OWN
FINDBLK   0019    SHARED     OWN
CHNGBLK   0019    EXCLUSIVE  WAIT
```

Example 8

To display information for all latches with contention involving job FINDREC, enter:

```
D GRS,L,C,JOB=FINDREC or
D GRS,LATCH,CONTENTION,JOBNAME=FINDREC
```

Example 9

To display contention information for ENQs and RESERVEs, enter:

```
D GRS,E,C or D GRS,ENQ,CONTENTION
```

Example 10

To display the contents of all resource name lists (RNLs) in the current global resource serialization complex, enter:

```
D GRS,RNL=ALL
```

The system displays the following message:

```
ISG343I 18.10.38 GRS STATUS
LIST TYPE QNAME  RNAME
INCL GEN SYSDSN
INCLPATT SYSDSN SYS1.*.*
EXCL SPEC SYSDSN PASSWORD
EXCL PATT SYSDSN SYS?.*.*LIB
EXCL PATT SYSDSN SYS?.TEST
```

```
EXCL PATT SYSDSN SYS1.*.LOGREC
EXCL PATT SYSDSN SYS1.*.MANX
EXCL SPEC SYSDSN SYS1.BROADCAST
CON PATT*      *
```

Note the use of wildcard characters in this example.

Displaying TSO/E Parmlib Information

Use the DISPLAY IKJTSO command to display the specifications in the active IKJTSOxx parmlib member.

```
| D IKJTSO[,statement-name][,L={a|name|name-a}]
```

IKJTSO

The system displays information from the active IKJTSOxx parmlib member.

statement-name

Displays only the specifications from an individual statement in the active IKJTSOxx parmlib member. If specified, *statement-name* can be one of the following:

ALL list of all the specifications in the active IKJTSOxx member of SYS1.PARMLIB

ALLOCATE

ALLOCATE command default data set status

AUTHCMD

list of authorized commands

AUTHPGM

list of programs that are authorized when invoked via the CALL command

AUTHTSF

list of programs that are authorized when invoked through the TSO/E service facility

CONSOLE

message processing defaults for the CONSOLE command and its services

HELP list of help data sets for different languages

NOTBKGND

list of commands not supported in the background

PLATCMD

list of commands that can run on the TSO/E command invocation platform

PLATPGM

list of programs that can run on the TSO/E command invocation platform when invoked through TSO/E service facility

SEND SEND, OPERATOR SEND, LISTBC, and BROADCAST command defaults

TEST list of additional commands and subcommands valid under TEST and TESTAUTH

DISPLAY IKJTSO Command

TRANSREC

TRANSMIT/RECEIVE command options and defaults

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display the current specifications of the SEND statement in the active IKJTSOxx parmlib member, enter:

```
D IKJTSO,SEND
```

The following is a sample of the information that the system displays.

```
IKJ738I TS0/E PARMLIB SETTINGS :  
SYS1.PARMLIB(IKJTSOAI) on volume PRMVOL  
Activated by **IPL** on 2001-10-29 at 11:55:48 from system OSV313  
Applies to : OSV313  
THE FOLLOWING ARE THE PARMLIB OPTIONS FOR SEND:  
  
OPERSEND(ON)  
USERSEND(ON)  
SAVE(ON)  
CHKBROD(OFF)  
LOGNAME(*)  
USEBROD(ON)  
MSGPROTECT(OFF)  
SYSPLEXSHR(OFF)  
OPERSEWAIT(ON)  
USERLOGSIZE(1,2)  
BROADCAST(DATASET(SYS2.BROADCAST)  
              VOLUME(BRDVOL) TIMEOUT(10) PROMPT)
```

Displaying I/O Configuration Information

Use the DISPLAY IOS,CONFIG command to display IOS-related configuration information.

```
D IOS,CONFIG[(EDT)|(HSA)|(ALL)][,L={a|name|name-a}]
```

IOS,CONFIG

The system displays information about the I/O configuration through message IOS506I.

EDT

Displays (message IOS506I) the jobs with outstanding binds on the primary Eligible Device Table (EDT), and, if applicable, on the secondary EDT.

HSA

Displays (message IOS506I) the amount of the hardware system area (HSA) that is available to perform configuration changes and the amount of space for each subchannel set within each channel subsystem.

ALL

Displays information (message IOS506I) about the I/O configuration and the amount of the hardware system area (HSA) that is available to perform configuration changes.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Displaying Captured UCB Information

Use the DISPLAY IOS,CAPTUCB command to display the current captured UCB protection status.

```
D IOS,CAPTUCB[,L={a|name|name-a}]
```

IOS,CAPTUCB

The system displays information about the current captured UCB protection status.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

The following example shows the message that is displayed in response to the command.

```
D IOS,CAPTUCB
IOS088I hh.mm.ss CAPTURED UCB DATA
CAPTURED UCB PROTECTION IS ENABLED|DISABLED
```

Displaying Dynamic Channel Path Management Information

Use the DISPLAY IOS,DCM command to display the current status of dynamic channel path management and to display the list of devices that do not have measurement blocks assigned.

```
D IOS,DCM[,L={a|name|name-a}]
```

IOS,DCM

The system displays information about dynamic channel path management.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

DISPLAY IOS,EKM Command

Displaying Encryption Key Manager (EKM) Status

Use the DISPLAY IOS,EKM command to display the current hostnames and connection numbers of encryption key management.

Use the DISPLAY IOS,EKM,VERIFY={PRIMARY|SECONDARY|ALL} to verify the connection of encryption key manager.

```
D IOS,EKM[,VERIFY={PRIMARY|SECONDARY|ALL}]
      [,L={a|name|name-a}]
```

IOS,EKM

Displays the host names of the primary and secondary encryption key managers, the maximum number of connections and the maximum number of permanent connections that can be used for encryption key management.

VERIFY=PRIMARY

Initiates an asynchronous test exchange with the primary encryption key manager to verify its usability.

VERIFY=SECONDARY

Initiates an asynchronous test exchange with the secondary encryption key manager to verify its usability.

VERIFY=ALL

Initiates an asynchronous test exchange with both the primary and secondary encryption key managers to verify their usability.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

For an EKM,VERIFY= command either an IOS631I message is issued indicating that the encryption key manager was successfully connected, or an IOS627E message is issued after several minutes indicating that there was a failure.

Example 1

To display the current encryption key manager settings, enter:

```
D IOS,EKM
```

The system displays the following message:

```
SY1  IOS099I 17.04.49 EKM HOSTS 833
PRIMARY HOSTNAME=text1
SECONDARY HOSTNAME=text2
MAX CONNECTIONS = dd1 PERMANENT CONNECTIONS = dd2
```

Example 2

To test the connectivity of the primary encryption key manager, enter:

```
D IOS,EKM,VERIFY=PRIMARY
```

If the encryption key manager connection is successfully verified, the system displays the following message:

IOS631I PRIMARY ENCRYPTION KEY MANAGER WAS SUCCESSFULLY CONNECTED

If the encryption key manager connection is not successfully verified and has failed, the system displays the following message:

IOS627E PRIMARY INTERFACE WITH ENCRYPTION KEY MANAGER CANNOT BE CONNECTED DUE TO *reason*

Displaying FICON Switch Data Information

Use the DISPLAY IOS,FICON command to display the current status (enabled|disabled) of FICON switch statistics gathering.

```
D IOS,FICON[,L={a|name|name-a}]
```

IOS,FICON

The system displays information about FICON switches.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Displaying IOS Group Information

Use the DISPLAY IOS,GROUP command to display a list of system names that all belong to the same IOS group.

```
D IOS,GROUP[,L={a|name|name-a}]
```

IOS,GROUP

The system displays information about systems in an IOS group.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Displaying IOS HYPERPAV Information

Use the DISPLAY IOS,HYPERPAV command to display the current HyperPAV enablement status.

```
D IOS,HYPERPAV[,L={a|name|name-a}]
```

DISPLAY IOS,HYPERPAV Command

IOS,HYPERPAV

The system displays information about the current HyperPAV enablement status.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Displaying MIDAW Facility Status

Use the DISPLAY IOS,MIDAW command to display the current status (enabled or disabled) of the modified indirect addressing word (MIDAW) facility.

```
D IOS,MIDAW[,L={a|name|name-a}]
```

IOS,MIDAW

The system displays status information about the MIDAW facility.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Displaying MIH and I/O Timing Limits

Use the DISPLAY IOS,MIH command to request a display of the current time intervals for the missing interrupt handler (MIH) or I/O timing (IOT) limits.

You can display the MIH time interval for all device classes, a specific device class, or one or more specific devices.

You can display the I/O timing limits for a device class or for one or more specific devices. You can request that the display be either formatted or unformatted.

The display, which is delivered by the WTO facility, is limited to 255 lines of output. If the limit is reached, the display is truncated. In this case, the last line of output is "... MAXIMUM DISPLAY OF 255 LINES EXCEEDED". Use the TIME, DEV, or DEVX parameter to request a selective display, to avoid missing data that meets the search criteria.

```
D IOS,MIH[,TIME={ALL|option}
      [
        [, {DEV }={([/]devnum[, [/]devnum1]...)
          [
            {DEVX } {[/]devnum-[/]devnum1[, [/]devnum2-[/]devnum3]...
          ]
        [
          {TDEV }
        [
          {TDEVX}
        [
          ,IOTHSWAP
      [
        ,L={a|name|name-a}
      ]]
```

The parameters are:

IOS,MIH

The MIH detects missing interrupt conditions. The I/O timing facility detects I/O requests that have exceeded the specified time limit. This command allows you to display the I/O timing limits or MIH time intervals established for the different devices on the system.

TIME=ALL

Displays the IOT and MIH time intervals established for all device classes and all individual devices.

TIME=option

Displays one of the following time intervals:

UREC

Displays the MIH time interval for the unit record device class.

TAPE

Displays the MIH time interval for the tape drive device class.

CTC

Displays the MIH time interval for the channel-to-channel device class.

COMM

Displays the MIH time interval for the communications device class.

CHAR

Displays the MIH time interval for the character reader device class.

GRAF

Displays the MIH time interval for the graphics device class.

DASD

Displays the MIH time interval for the DASD device class.

IOTDASD

Displays the I/O timing (IOT) time interval for the DASD device class.

USnn

Displays the MIH or IOT time interval for a user-specified class, where nn can be any two-digit number from 01 through 99 that matches a device group created by MIH or IOT processing. A user-specified device group is a set of devices associated with a specific time interval. The system creates this type of group and assigns the user class number (USnn) when either of the following is true:

- The MIH time interval is not equal to the time interval of its device class
Note: Some devices present their own MIH timeout values, via the primary/secondary MIH timing enhancement contained in the self-describing data for the device. If the primary MIH timeout value for the device does not equal the timeout value for the device class, and the device's timeout value has not been altered by the user, the system will create a user-specified class to contain the timeout value for the device. The user-specified class for these devices will be created at IPL (if the device is defined to be ONLINE), or at VARY ONLINE time.
- The IOT time interval is not equal to the time interval of its device class.

HALT

Displays the MIH time interval for monitoring halt (HSCH) and clear (CSCH) subchannel operations. This keyword is device independent; setting it affects all devices on the system.

DISPLAY IOS,MIH Command

MNTS

Displays the MIH time interval for monitoring *mount pending* conditions for DASD and TAPE devices. This keyword is device independent; setting it affects all devices on the system.

DEV= {[[/]devnum,[/]devnum1]...} {([/]devnum-[/]devnum1,[/]devnum2-[/]devnum3]...) }

Displays the MIH time interval for a specific device number or for a range of device numbers. You can specify one or more single device numbers, one or more ranges of device numbers, or all device numbers. When you specify a range of device numbers, the first device number in the range must be less than or equal to the second device number. If you specify only one device number, you can omit the parentheses.

The display is formatted.

A device number consists of 3 or 4 hexadecimal digits, optionally preceded by a slash (/).

DEVX= {[[/]devnum,[/]devnum1]...} {([/]devnum-[/]devnum1,[/]devnum2-[/]devnum3]...) }

Displays the MIH time interval for the specific defined device number or for a range of device numbers. You can specify one or more single device numbers, one or more ranges of device numbers, or all device numbers. When you specify a range of device numbers, the second device number in the range must be more than or equal to the first device number. If you specify only one device number, you can omit the parentheses.

The display is unformatted.

A device number consists of 3 or 4 hexadecimal digits, optionally preceded by a slash (/).

TDEV= {[[/]devnum,[/]devnum1]...} {([/]de vnum-[/]devnum1,[/]devnum2-[/]devnum3]...) }

Displays the I/O timing limit for the specific defined device number or for a range of device numbers. You can specify one or more single device numbers, one or more ranges of device numbers, or all device numbers. When you specify a range of device numbers, the second device number in the range must be more than or equal to the first device number. If you specify only one device number, you can omit the parentheses.

The display is formatted.

A device number consists of 3 or 4 hexadecimal digits, optionally preceded by a slash (/).

TDEVX= {[[/]devnum,[/]devnum1]...} {([/]devnum-[/]devnum1,[/]devnum2-[/]devnum3]...) }

Displays the I/O timing limit for the specific defined device number or for a range of device numbers. You can specify one or more single device numbers, one or more ranges of device numbers, or all device numbers. When you specify a range of device numbers, the first device number in the range must be less than or equal to the second device number. If you specify only one device number, you can omit the parentheses.

The display is unformatted.

A device number consists of 3 or 4 hexadecimal digits, optionally preceded by a slash (/).

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display the MIH and IOT intervals for all device classes and all individual devices, enter:

D IOS,MIH

Example 2

To display the MIH interval for the devices in address range 000 through 01A, enter:

D IOS,MIH,DEV=(000-01A)

The display is formatted:

```
IOS086I 14.34.51 MIH DEVICE TIMES
 0000=03:00,    0002=03:00,    0003=03:00,    0004=03:00,    0005=03:00,
 0006=03:00,    0007=03:00,    0008=03:00,    000A=03:00,    000B=03:00,
 000C=03:00,    000D=03:00,    000E=03:00,    000F=03:00,    0011=03:00,
 0012=03:00,    0013=03:00,    0014=03:00,    0015=03:00,    0016=03:00,
 0017=03:00,    0018=03:00,    0019=03:00,    001A=03:00.
```

Example 3

To display the MIH interval for the devices in address range 000 through 0D9, enter:

D IOS,MIH,DEVX=(000-0D9)

The display will be unformatted.

```
IOS086I 14.43.28 MIH DEVICE TIMES
(0000,0002-0008,000A-000F,0011-001D)=03:00, (001E-009D)=00:00,
(009E-009F)=03:00, (00A0-00A1)=00:15, (00BA-00BF)=00:00, (00C0-00C1)=
00:15, (00CA-00CF)=00:00.
```

Example 4

To display the MIH interval for the devices in address range ABC0 through ABCD, enter:

D IOS,MIH,DEV=(/ABC0-/ABCD)

Example 5

To display the I/O timing limit for the devices in address range 000-010, enter:

DISPLAY IOS,MIH,TDEVX=(000-010)

The system might display the following:

```
IOS086I 14.44.50 IOT DEVICE TIMES
 0002=00:10, (0003-0008,000A-000F)=00:00, 0010=00:15.
```

DISPLAY IOS,MIH Command

In this example, devices 000, 001, and 009 do not exist. The I/O timing limit for device 002 is 0 minutes and 10 seconds. The I/O timing limit for device 003-008 and 00A-00F is 0 minutes and 0 seconds. For device 010, the I/O timing limit is 0 minutes and 15 seconds.

Displaying IOS Storage Residency Information

Use the DISPLAY IOS,STORAGE command to display the storage residency of where IOS blocks are to be obtained (24 or 31 bit).

```
D IOS,STORAGE[,L={a|name|name-a}]
```

IOS,STORAGE

The system displays information about IOS storage residency.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Displaying the Devices Stopped by the IOACTION Command

Use the DISPLAY IOS,STOP command to identify the shared DASD that is currently stopped as result of the IOACTION STOP command.

```
D IOS,STOP[,L={a|name|name-a}]
```

IOS,STOP

The system displays information (message IOS610I) about the devices affected by the IOACTION STOP command. For example:

IOS610I IOACTION – THE FOLLOWING DEVICE(S) ARE IN THE STOPPED STATE:

420- 42F, 440- 44F, 470- 48F, 4A0- 4AF, 4C0- 4E7

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Displaying IPL Information

Use the DISPLAY IPLINFO command to display the following information:

- The date and time of the IPL
- The release level of the system
- The contents of parmlib members IEASYSxx and IEASYMxx
- LOADxx information used for the IPL
- The architecture level of the IPL
- The IODF (input/output definition file) device
- The IPL device and volume serial
- The status of MTL (manual tape library) tape devices

```
D IPLINFO [,L={a|name|name-a}]
```

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example

Assume a system has the following characteristics:

- IPL occurred on December 1, 2006 at 1:15 a.m.
- The release level is z/OS 1.8.0
- Parmlib member LOAD08, which resides on device ACB2, was used to IPL
- The system is running in z/Architecture mode
- The system treats MTL tape devices as MTL resident drives
- The system used system symbols definitions in IEASYMX6, IEASYMU6, IEASYM0L, and IEASYMR8.
- The operator entered IEASYSST and IEASYSLN in reply to the system parameters prompt.
- The IODF device resides on ACB2
- The IPL device is 3C2A and its volume serial is D83EL8

Issue the following command:

```
D IPLINFO
```

The system returns the following display, slightly modified to improve readability:

```
D IPLINFO
IEE254I 11.14.07 IPLINFO DISPLAY 350
SYSTEM IPLED AT 01.15.39 ON 12/01/2006
RELEASE z/OS 01.08.00 LICENSE = z/OS
USED LOAD08 IN SYS0.IPLPARM ON ACB2
ARCHLVL = 2 MTLSHARE = N
IEASYM LIST = (X6,U6,L,R8)
IEASYS LIST = (ST,LN) (OP)
IODF DEVICE ACB2
IPL DEVICE 3C2A VOLUME D83EL8
```

Displaying System Activity

Use the DISPLAY JOBS (or J or A or TS) command to display information about current system activity. The descriptions of messages IEE114I and IEE115I show the resulting display. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books to see the display. The syntax of this command is:

```
D {JOBS|J|A|TS} [, {LIST|L}, [{USERID=id}]
          |, {ALL|A}
          |, {jobname[.identifier]} | {(jobname)}
[,L={a|name|name-a}]
```

DISPLAY JOBS or J or A or TS Command

The parameters are:

JOBS or J or A or TS

The system is to display the following overview information about system activity:

- Number of active batch jobs
- Number of started tasks (MOUNT commands in execution are treated as started tasks)
- Number of logged-on time-sharing users
- Number of logged-on time-sharing users running under TSO/VTAM
- The maximum number of time-sharing users allowed to be logged on under TSO/VTAM
- Number of active system address spaces
- Number of active initiators including Advanced Program-to-Program Communication/MVS (APPC/MVS) transaction initiators
- Number of z/OS UNIX System Services address spaces

LIST or L

The system is to display detailed information for active jobs and started tasks (JOBS or J), logged-on time-sharing users (TS), active APPC/MVS transaction programs, or all three (A).

ALL or A

The system is to display more detailed information for active jobs and started tasks, logged-on time-sharing users, active APPC/MVS transaction programs, active initiators, and active system address spaces than that supplied by LIST.

jobname[.identifier] or (jobname)

The system is to display detailed information for active jobs, started tasks, logged-on time-sharing users, active APPC/MVS transaction programs, active initiators, and active system address spaces with the specified name. This information includes the data space names associated with the address space. If the specified name is the same as a valid secondary operand, you must enclose it in parentheses. (See Example 7.)

The specified name can be the name of a job, started task, APPC/MVS transaction program, APPC/MVS scheduler initiator, time-sharing user, or system address space. For a job, started task, APPC/MVS transaction program, APPC/MVS scheduler initiator, or system address space, the name can be 1 to 8 alphanumeric or national characters or 1 to 7 alphanumeric or national characters followed by an asterisk. For a time-sharing user, the name can be 1 to 7 alphanumeric or national characters or 1 to 6 alphanumeric or national characters you use the asterisk format, all jobs, APPC/MVS transaction programs, APPC/MVS scheduler initiators, started tasks, or time-sharing users with names that begin with the specified characters are displayed. See Example 5.

Notes:

1. You can use asterisk notation to display information about more than one job or started task. See "Using Wildcards to Display System Activity" on page 4-154 for more information.
2. The only time you may use an asterisk in the first position of a name is to specify the master scheduler address space: *MASTER*. For *MASTER*, the name must be the complete address space name.

3. Started task names can come from a variety of sources. The name of a started task depends on whether the JOBNAME keyword was specified on the START command.

If JOBNAME was specified, *jobname* is the name assigned to the started task.

If JOBNAME was not specified and the source JCL for the started task is:

- A ***job***, the system uses the jobname provided with the JCL JOB statement.
- A ***procedure***, the system uses the member name as the jobname.

Refer to “Displaying Started Task Status” on page 4-157 for information on determining the jobname and identifier of currently active started tasks.

identifier

The started task identifier. You can use asterisk notation to display information about more than one job or started task. See “Using Wildcards to Display System Activity” on page 4-154 for more information.

USERID=userid

A filter to display only the work executing on behalf of *userid*. This userid may be specified on the *USER=* keyword in JCL, or the userid that requested that a transaction occur.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Information for the LIST Operand

The system displays detailed information when you specify LIST or L. The detailed information is displayed after the overview information. Table 4-12 shows the operands that you can combine with LIST or L and the detailed information that results from each combination.

Table 4-12. Displaying System Activity: Information for the LIST Operand

Primary Operand	Information for the LIST Operand
JOBS or J	List of active jobs, including, for each job: <ul style="list-style-type: none"> • Jobname, APPC/MVS transaction program, started task • Stepname • Procedure stepname or requesting userid • Type of job • Address space status • Central (real) address range (V=R only)
TS	List of logged-on time-sharing users, including, for each user: <ul style="list-style-type: none"> • Userid • Address space status
A	List of all active units of work, including: <ul style="list-style-type: none"> • For active jobs, started tasks, and APPC/MVS transaction programs all of the information listed for JOBS or J • For logged-on time-sharing users, all of the information listed for TS

DISPLAY JOBS or J or A or TS Command

The system displays more detailed information when you specify ALL or A than when you specify LIST or L. The detailed information is displayed after the overview information. Table 4-13 shows the operands that you can combine with ALL or A and the detailed information that results from each combination.

Table 4-13. Displaying System Activity: Information for the ALL Operand

Primary Operand	Information for the ALL Operand
JOBS or J	List of active jobs, including, for each job: <ul style="list-style-type: none"> • Jobname (APPC/MVS transaction program name), started task • Stepname • Procedure stepname or requesting userid • Type of job • Address space identifier • Address space status • Program event recording (PER) activity • Number of outstanding step-must-complete requests • Processor affinity • Elapsed time since initiation • Accumulated processor time • Work unit identifier • Transaction requestor's userid • Central (real) address range (V=R only)
	Workload management information: <ul style="list-style-type: none"> • Workload associated with the address space • Service class associated with the address space • Resource group associated with the service class. "N/A" is displayed if there is no resource group assigned to the service class • Whether the address space has been quiesced by the RESET command • Whether the address space is a server • The service class period
TS	List of logged-on time-sharing users, including, for each user: <ul style="list-style-type: none"> • Address space status • Address space identifier • Program event recording (PER) activity • Number of outstanding step-must-complete requests • Processor affinity • Elapsed time since LOGON • Accumulated processor time • Work unit identifier
	Workload management information: <ul style="list-style-type: none"> • Workload associated with the address space • Service class associated with the address space • Resource group associated with the service class. "N/A" is displayed if there is no resource group assigned to the service class • Whether the address space has been quiesced by the RESET command • Whether the address space is a server • The service class period

DISPLAY JOBS or J or A or TS Command

Table 4-13. Displaying System Activity: Information for the ALL Operand (continued)

Primary Operand	Information for the ALL Operand
A	List of all active units of work, including: <ul style="list-style-type: none"> • For each active job, started task, APPC/MVS transaction program, and APPC/MVS scheduler initiator: <ul style="list-style-type: none"> – All of the information listed for JOBS or J • For each time-sharing user: <ul style="list-style-type: none"> – All of the information listed for TS • For each active system address space: <ul style="list-style-type: none"> – Name – Stepname – Procedure stepname

The system displays the most detailed information when you supply a specific name. This additional information may be useful to the system programmer for diagnostics. The detailed information is displayed after the overview information. Table 4-14 shows the operands that you can combine with a specific name and the detailed information that results from each combination.

Table 4-14. Displaying System Activity: Information for a Specific Name

Primary Operand	Information for the NAME Operand
JOBS or J	List of active jobs for the specific name: <ul style="list-style-type: none"> • Jobname, APPC/MVS transaction program name, initiator address space name • Stepname • Procedure stepname or requesting userid • Type of job • Address space identifier • Address space status • Program event recording (PER) activity • Number of outstanding step-must-complete requests • Processor affinity • Elapsed time since initiation • Accumulated processor time • Work unit identifier • Transaction requestor's userid • Central (real) address range (V=R only) • Central (real) address of address space number second table (ASTE) • Data space names and the data space ASTEs
	Workload management information: <ul style="list-style-type: none"> • Workload associated with the address space • Service class associated with the address space • Resource group associated with the service class. "N/A" is displayed if there is no resource group assigned to the service class • Whether the address space has been quiesced by the RESET command • Whether the address space is a server • The service class period

DISPLAY JOBS or J or A or TS Command

Table 4-14. Displaying System Activity: Information for a Specific Name (continued)

Primary Operand	Information for the NAME Operand
TS	List of logged-on time-sharing users with the specific name: <ul style="list-style-type: none"> • Address space status • Address space identifier • Program event recording (PER) activity • Number of outstanding step-must-complete requests • Processor affinity • Elapsed time since LOGON • Work unit identifier • Accumulated processor time • Central (real) address of address space number second table (ASTE) • Data space names and the data space ASTEs
	Workload management information: <ul style="list-style-type: none"> • Workload associated with the address space • Service class associated with the address space • Resource group associated with the service class. "N/A" is displayed if there is no resource group assigned to the service class • Whether the address space has been quiesced by the RESET command • Whether the address space is a server • The service class period
A	List of all active units of work for the specific name, including: <ul style="list-style-type: none"> • For an active job, active APPC/MVS transaction program name, initiator address space and started task: <ul style="list-style-type: none"> – All of the information listed for JOBS or J • For a time-sharing user: <ul style="list-style-type: none"> – All of the information listed for TS • For an active system address space: <ul style="list-style-type: none"> – Name – Stepname – Procedure stepname

Using Wildcards to Display System Activity

You can use the asterisk (*) wildcard to display information about more than one job or started task. A trailing asterisk (*) indicates that a DISPLAY command applies to *all* jobs or started tasks that match a leading character string. The DISPLAY JOBS, J, A, or TS command supports only a *trailing* asterisk. You cannot specify an asterisk in other character positions in job or started task names.

For example, you can enter the following command to display information about all jobs and started tasks beginning with the characters X11:

D A,X11*

You can also use the asterisk wildcard to specify both a job name and identifier. The system displays information about all jobs and started tasks that match the combinations of characters that precede one or more asterisks.

For example, you can enter the following command to pass a two-digit value to all jobs with names that begin with J22 and identifiers that begin with X11:

D A,J22*.X11*

Remember the following rules when using the asterisk wildcard in the DISPLAY JOBS, J, A, or TS command:

DISPLAY JOBS or J or A or TS Command

- If you specify both the *jobname* and *identifier* values, you cannot specify a single asterisk for *both* values.

For example, to display information about all jobs with names beginning with J22, you can specify a single asterisk on the *identifier* to indicate a *wildcard*:

D A,J22*.*

If you were to remove the J22 characters from the above command, it would not be valid. You cannot specify *.* without a leading character string on the *jobname* parameter, the *identifier* parameter, or both.

- A slash (/) cannot precede an identifier that contains an asterisk.

The following tables describe how the asterisk wildcard works with DISPLAY JOBS, J, A, or TS. Table 4-15 shows examples of START commands used to start jobs. The third and fourth columns show the associated jobnames and identifiers.

Table 4-15. Examples of START Commands to Start Jobs

Job Number	START Command	Jobname	Identifier
1	START YZ	YZ	YZ
2	START WX.YZ	WX	YZ
3	START WX.YZ1	WX	YZ1
4	START WX1.YZ1	WX1	YZ1
5	START WX, JOBNAME =WX1	WX1	WX1
6	START WX, JOBNAME =WX2	WX2	WX2
7	START WX, JOBNAME =YZ	YZ	YZ
8	START Q.YZ3	Q	YZ3
9	START WX.R1	WX	R1
10	START WX, JOBNAME =YZ4	YZ4	YZ4

Table 4-16 shows examples of DISPLAY commands. (The examples apply to DISPLAY JOBS, J, A, or TS, although only DISPLAY A commands are used in this figure.) The numbers in the second column indicate the jobs in Table 4-15 that apply to each DISPLAY command.

Table 4-16. Examples of DISPLAY Commands

DISPLAY Command	Displayed Jobs
D A,YZ	1, 7
D A,WX.YZ	2
D A,WX.YZ*	2, 3
D A,YZ.*	1, 7
D A,WX*	2, 3, 4, 5, 6, 9
D A,YZ*	1, 7, 10
D A,WX*.YZ	2
D A,WX*.YZ*	2, 3, 4
D A,*.YZ*	1, 2, 3, 4, 7, 8, 10
D A,*.YZ	1, 2, 7
D A,WX*.*	2, 3, 4, 5, 6, 9
D A,WX.*	2, 3, 9

DISPLAY JOBS or J or A or TS Command

Table 4-16. Examples of DISPLAY Commands (continued)

DISPLAY Command	Displayed Jobs
D A,*	Not valid (can be done using D A,ALL)
D A,*.*	Not valid (can be done using D A,ALL)

The following are examples of various forms of the DISPLAY JOBS, J, A, or TS command:

Example 1

To display detailed information about all active units of work, enter:

D A,L

Example 2

To display detailed information about active jobs, enter:

D J,L

Example 3

To display more detailed information about active jobs, enter:

D J,A

Example 4

To display detailed information about any active time-sharing user with the name WAGNERJ, enter:

D TS,WAGNERJ

Example 5

To display detailed information about all active jobs, started tasks, time-sharing users, or address spaces with the name beginning with D96, enter:

D A,D96*

Example 6

To display detailed information about the master scheduler address space, enter:

D A,*MASTER*

Example 7

To display detailed information about any active time-sharing user with the name LIST, enter:

D TS,(LIST)

Example 8

To display detailed information about all jobs or APPC/MVS transaction programs running for user WANDA, enter:

D J,L,USERID=WANDA

Example 9

DISPLAY JOBS or J or A or TS Command

To display detailed information about any jobs or APPC/MVS transaction programs named PHONE running for user WANDA, enter:

```
D J,PHONE,USERID=WANDA
```

Example 10

To display detailed information about all initiator address spaces beginning with INIT, enter:

```
D A,INIT*
```

Example 11

To display detailed information about started task X11 which has a job name of AOR2, enter:

```
D A,AOR2.X11
```

Example 12

To display detailed information about all started tasks with the job name AOR2, enter:

```
D A,AOR2.*
```

Example 13

To display detailed information about all started tasks with a job name of AOR2 and identifiers that start with T1, enter:

```
D A,AOR2.T1*
```

Example 14

To display detailed information about all started tasks with job names that start with AOR and identifiers that start with T1, enter:

```
D A,AOR*.T1*
```

Example 15

To display detailed information about all started tasks with identifiers that start with T1, enter:

```
D A,*.T1*
```

Displaying Started Task Status

The displayed output for a display command depends on whether the started task source JCL is a JOB or a cataloged procedure and whether you use the **JOBNAME** parameter on the START command.

- **JOBNAME parameter.** This parameter on the START command names the started task. (See “Starting a System Task from a Console” on page 4-575 for additional information.)
- **Membername.** If you do not use the **JOBNAME** parameter on the START command and the source JCL is a procedure, the system automatically assigns the member name as the jobname.
- **Jobname within the source JCL.** If you do not use the **JOBNAME** parameter on the START command and the source JCL for the started task is a job, a job name is assigned based on the job name on the JOB statement.

More on the DISPLAY A Command

- **Identifier.** If you use the identifier on the START command, you can identify the started task by both the identifier and the jobname that was assigned by the system.

If you issue a DISPLAY A,ALL command, the system will display status about all started tasks. In the following examples, the source JCL is provided and examples of the changes in the output are provided.

For the purposes of understanding the display output fields, the following illustration indicates what each column represents in the examples that follow. Note, however, that the illustration has been slightly modified to improve the readability and the column identifiers have been added as pointers; the actual display output will not appear as in the illustration that follows:

SYS1 D A,WTOR				
SYS1 IEE115I 16.33.03 1996.308 ACTIVITY 048				
JOBS	M/S	TS USERS	SYSAS	INITS
00000	00006	00001	00016	00002
jobname	identifier	stepname		
WTOR	WTOR	ONLYSTEP	OWT S	A=0019

Figure 4-3. Display Output Illustration (Column Descriptions)

In the illustration:

- WTOR is the *jobname*
- WTOR is also the *identifier*
- ONLYSTEP is the *stepname*

Output (When the Member Contains a Procedure)

The following examples indicate what the different commands will generate in displayed output when the started task source JCL is a procedure (SYS1.PROCLIB member named WTOR) as follows:

```
//ONLYSTEP EXEC PGM=WTOR,PARM='HELLO'
```

For the example START WTOR where only the *membername* was specified (neither *JOBNAME* nor *identifier* were specified), only the *membername* appears in the output.

SYS1 D A,WTOR				
SYS1 IEE115I 16.33.03 1996.308 ACTIVITY 048				
JOBS	M/S	TS USERS	SYSAS	INITS
00000	00006	00001	00016	00002
WTOR	WTOR	ONLYSTEP	OWT S	A=0019

Figure 4-4. Display Output from D A,WTOR (Membername)

For the example START WTOR. IDENTIFY (*membername* and *identifier* were specified), the *membername* and *identifier* appear in the output.

```
SYS1 D A,WTOR
SYS1 IEE115I 16.34.57 1996.308 ACTIVITY 083
JOBS M/S TS USERS SYSAS INITS
00000 00006 00001 00016 00002
WTOR IDENTIFY ONLYSTEP OWT S A=0019
```

Figure 4-5. Display Output for D A,WTOR (Membername and Identifier)

For the example START WTOR,JOBNAM=WTORNAME (*membername* and *JOBNAME* were specified), only the job name appears in the output.

```
SYS1 D A,WTORNAME
SYS1 IEE115I 16.36.46 1996.308 ACTIVITY 118
JOBS M/S TS USERS SYSAS INITS
00000 00006 00001 00016 00002
WTORNAME WTORNAME ONLYSTEP OWT S A=0019
```

Figure 4-6. Display Output for D A,WTOR (Membername and JOBNAME)

Output (When the Member Contains a Job)

The following examples indicate how the different displayed output appears for a started task (SYS1.STCJOBS member named SYM1) with source JCL of a JOB, given three different START commands. The source JCL of SYM1:

```
//SYMTEST JOB 'accounting_info',MSGLEVEL=(1,1)
//STEP1 EXEC PGM=WTOR,PARM='HELLO',TIME=1
```

For the example START SYM1, where only the *membername* is specified (neither *JOBNAME* nor *identifier* was specified), only the job name (provided in the member) appears in the output.

```
SYS1 D A,SYMTEST
SYS1 IEE115I 16.20.14 1996.308 ACTIVITY 811
JOBS M/S TS USERS SYSAS INITS
00000 00006 00001 00016 00002
SYMTEST SYMTEST STEP1 OWT S A=0019
```

Figure 4-7. Display Output from D A,SYM1

For the example START SYM1.IDENTIFY (*membername* and *identifier* were specified), both the job name (in the member) and the *identifier* (specified in the command) appear in the output.

```
SYS1 D A,SYMTEST
SYS1 IEE115I 16.22.24 1996.308 ACTIVITY 832
JOBS M/S TS USERS SYSAS INITS
00000 00006 00001 00016 00002
SYMTEST IDENTIFY STEP1 OWT S A=001A
```

Figure 4-8. Display Output from D A,SYMTEST

For the example START SYM1,JOBNAM=SYMBOLS (*membername* and *JOBNAME* were specified), only the job name specified in the command appears in the output.

DISPLAY LLA Command

```
SYS1 D A,SYMBOLS
SYS1 IEE115I 16.23.41 1996.308 ACTIVITY 856
JOBS    M/S    TS USERS    SYSAS    INITs
00000   00006   00001   00016   00002
SYMBOLS SYMBOLS STEP1    OWT S    A=001A
```

Figure 4-9. Display Output from D A,SYMBOLS

Displaying Library Lookaside Information

Use the DISPLAY LLA command to display information about library lookaside, and to display a list of all the libraries that LLA is managing.

The syntax of this command is:

```
D LLA[,L={a|name|name-a}]
```

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example

If you enter a D LLA command the format of the output is:

```
CSV600I 12.38.09 LLA DISPLAY 529
EXITS: CSVLLIX1 - INACTIVE CSVLLIX2 - ON
VLF: ACTIVE GET LIB ENQ: YES SEARCH FAIL COUNT: 0
LNKLST SET: IPL
12 LIBRARY ENTRIES FOLLOW
ENTRY  L F R P  LIBRARY NAME
      1 L          SYS1.CSSLIB
      2 L          SYS1.MIGLIB
      3 L          TCPIP.SEZALOAD
      4 L          SYS1.VTAMLIB
      5 L          SYS1.CMDLIB
      6          IMSVS.TEST.PGMLIB
      7 L          SYS1.MIGLIB
      8 L          SYS1.LINKLIB
      9 L          TCPIP.SEZSLNK2
     10 L          SYS1.SORTLPA
     11 F          SYS1.JOBLIB
     12 L          SYS1.SCBDHENU
```

The following describes some of the CSV600I output fields. For a complete description of all of the output fields, see message CSV600I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

ENTRY

The entry number of the library being displayed. This does not relate to the order in which the libraries were specified or are processed.

L or LNKLST

The LNKLST status of the library being displayed. L is one of the following:

L The library is in the current LNKLST.

A The library is in an active, not current, LNKLST.

(blank)

The library is not in the LNKLST.

F or FREEZE

The FREEZE state of the library being displayed. F is one of the following:

F The library is in freeze state.

(blank)

The library is not in freeze state.

R or REMOVE

The REMOVE status of the library being displayed. R is one of the following:

R The library was requested to be removed.

(blank)

The library was not requested to be removed.

P or PDSE

The indicator of whether or not the library is a partitioned data set extended, or PDSE. P is one of the following:

P The library is a PDSE.

(blank)

The library is not a PDSE.

Displaying the System Logger and its Log Streams

You can use the DISPLAY LOGGER command to display the status of the system logger, individual log streams, or one or all log streams from a sysplex view.

Note: You can use the asterisk as a wildcard character with the DISPLAY LOGGER command; specify an asterisk (*) as the search argument or specify an asterisk as the last character of a larger search argument. If used, the wildcard must be the last character in the search argument, or the only character.

Restrictions

- Do not use the same parameter twice within a single command.
- Do not exceed a command line length of 128 characters.

Syntax

```
D LOGGER[,{STATUS|ST}
      [,{CONNECTION|CONN|C}[,LSNAME|LSN=logstreamname] ]
      [,Jobname|JOB|J=mvsjobname]
      [,,{SUMM|S } ]
      {Detail|D}
      [,DASDONLY ]]

      [,,{LOGSTREAM|L}[,LSName=logstreamname ] ]
      [,STRNAME|STRN=structurename]
      [,DASDONLY ]]

      [,,{STRUCTURE|STR}[,STRNAME|STRN=structurename] ]
      [,L={a|name|name-a} ]
```

DISPLAY LOGGER Command

The parameters are:

Status or ST

Display the current operational status of the system logger. Status is the default if you specify no parameters. Possible status values are:

NOT AVAILABLE FOR IPL - XCFLOCAL MODE

Because the sysplex is in XCF local mode, the system logger is not available for the life of this IPL.

NOT AVAILABLE FOR IPL - NOSTART SPECIFIED

Because IXGLOGR=NOSTART is specified on LOGR SUBSYS, the system logger is not available for the life of this IPL.

INITIALIZING

The system logger is in the process of initializing.

INITIALIZING - DRXRC CONVERSION NOT DONE

The system logger is in the process of initializing; however, DRXRC conversion has not been done.

INITIALIZING - LOGR CDS IS NOT AVAILABLE

The system logger is in the process of initializing; however, the LOGR couple data set is not available.

ACTIVE

The system logger is running.

ACTIVE - DRXRC CONVERSION NOT DONE

The system logger is running; however, DRXRC conversion has not been done.

ACTIVE - LOGR CDS IS NOT AVAILABLE

The system logger is running; however, the LOGR couple data set is not available.

NOT ACTIVE

The system logger has terminated.

Connection or Conn or C

Display all log streams with one or more connections to the system(s) from which you issued the command.

You can use the following filters to limit or change the information displayed. (Note that if you use the SYSPLEX filter, the view of the output is changed to the systems or resources that are connected to the log stream, a sysplex view.)

LSName or LSN = *logstreamname*

This filter requests a display of all actively connected log streams that match the specified log stream name.

Jobname or JOB or J = *mvsjobname*

This filter requests a display of all log streams with one or more connections to the specified jobname.

Summ or S, or Detail or D

These two mutually exclusive parameters are valid only when preceded by the specification of the LSName parameter or the Jobname parameter (or both) as part of the **Connection** display. **Summ** (summary), the default, displays a condensed overview of the requested information. **Detail** produces a more detailed report.

SYSPLEX

This filter requests to change the view of the output for the display logger command **Connection** option from a system view to a sysplex view. If you use the LSName | LSN filter to narrow the information to search for and display, the system displays information about systems and resources connected to the log stream. Otherwise, the display will show all log streams with one or more connections on the sysplex.

DASDONLY

This filter requests a display of all log streams with a DASDONLY configuration.

Logstream or L

Display log stream sysplex information.

LSName or LSN = *logstreamname*

This filter requests a display of all defined log streams that match the specified log stream name.

STRName or STRN = *structurename*

This filter requests a display of all log streams on the sysplex that are defined to a structure that matches the specified structure name.

DASDONLY

This filter requests a display of all log streams that match other filters that have a DASDONLY configuration.

STRucture or STR

Sort by structure name and display all log streams defined to any structure on a sysplex.

STRName or STRN = *structurename*

This filter requests a sort by structure and display of all log streams on the sysplex defined to the specified structure name.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Examples**Example 1**

Display the current operational status of the System Logger.

`DISPLAY LOGGER,STATUS`

Example 2

Display all log streams with one or more connections for the system that match the log stream name starting with the letters *logstr*.

`DISPLAY LOGGER,CONN,LSN=logstr*`

Example 3

Display all log streams with at least one active connection in the sysplex that matches the log stream name starting with the letters *logstr*.

`DISPLAY LOGGER,CONN,SYSPLEX,LSN=logstr*`

DISPLAY LOGGER Command

Example 4

Display all defined log streams for the sysplex that match the log stream name of *loga* and structure names that start with *list*.

```
DISPLAY LOGGER,L,LSN=loga,STRN=list*
```

Example 5

Display all defined log streams for the sysplex that start with *logstr* and have a DASD only configuration.

```
DISPLAY LOGGER,L,LSN=logstr*,DASDONLY
```

Example 6

Display all defined log streams for the sysplex and sort by structure name starting with the letters *list*.

```
DISPLAY LOGGER,STR,STRN=list*
```

Example 7

The following command displays details of all the log streams on the system with connections to the specified job name:

```
DISPLAY LOGGER,CONN,JOBNAME
```

The output appears as follows:

CONNECTION INFORMATION BY LOGSTREAM FOR SYSTEM SY1			
LOGSTREAM	STRUCTURE	#CONN	STATUS
NICKJ.TEST.LOGSTREAM	LIST01	00002	DISCONNECT PENDING
DUPLEXING: STAGING DATA SET			
STGDSN: IXGLOGR.NICKJ.TEST.LOGSTREAM.PLEX1			
VOLUME=ALL001 SIZE=0040000 % IN-USE=30			
GROUP: PRODUCTION			
DISCONNECT PENDING FOR 0003 MINUTES			
FORCE DISCONNECT IN PROGRESS			
JOBNAME: LOGTEST1 ASID: LOGTAS1			
R/W CONN: 0003			
RES MGR./CONNECTED: no			
IMPORT CONNECT: no			
JOBNAME: LOGTEST2 ASID: LOGTAS2			
R/W CONN: 0003			
RES MGR./CONNECTED: no			
IMPORT CONNECT: no			
NUMBER OF LOGSTREAMS: 000001			

Example 8

The following command displays inventory information by log stream for either the sysplex (DISPLAY LOGGER,CONN,SYSPLEX) or system (DISPLAY LOGGER,L):

```
DISPLAY LOGGER,CONN,SYSPLEX  
DISPLAY LOGGER,L
```

The output appears as follows:

INVENTORY INFORMATION BY LOGSTREAM			
LOGSTREAM	STRUCTURE	#CONN	STATUS
NICKJ.TEST.LOGSTREAM1	LIST01	00000	AVAILABLE
NICKJ.TEST.LOGSTREAM2	LIST02	00003	IN USE

```

SYSNAME: SY1
 DUPLEXING: LOCAL BUFFERS
SYSNAME: SY2
 DUPLEXING: STAGING DATA SET
SYSNAME: SY3
 DUPLEXING: LOCAL BUFFERS
GROUP: PRODUCTION
NICKJ.TEST.LOGSTREAM3      LIST03          00001     IN USE
SYSNAME: SY2
 DUPLEXING: LOCAL BUFFERS
GROUP: TEST

NUMBER OF LOGSTREAMS: 000003

```

Example 9

The following command displays the status of system logger:

```
DISPLAY LOGGER,ST
```

The output appears as follows:

```

SYSTEM LOGGER STATUS
SYSTEM   SYSTEM LOGGER STATUS
-----
SY1      ACTIVE
          LOGR CDS IS NOT AVAILABLE
          LOGGER SERVICES DISABLED FOR GROUP: TEST

```

Example 10

The following command displays the log streams defined to any structure in the sysplex:

```
DISPLAY LOGGER,STR
```

The output appears as follows:

```

INVENTORY INFORMATION BY STRUCTURE
STRUCTURE      GROUP      CONNECTED
-----
LIST01         TEST
              OA08553.LOG.STREAM    NO
LIST02
  *NO LOGSTREAMS DEFINED*
LONG_STRUCT_NAME  PRODUCTION
              NICKJ.TEST.LOG.STREAM    NO
              NICKJ.TEST.LOG.STREAM2  YES

NUMBER OF STRUCTURES: 3

```

Displaying the Logrec Recording Medium

Use the DISPLAY LOGREC command to display the current logrec error and environmental record recording medium and any alternate medium, if available. The DISPLAY LOGREC command produces the following:

- The current logrec error recording medium (either the name of a log stream, the name of a logrec data set, or IGNORE).
- The alternate recording medium, if a logrec data set has been defined.
- Both current and alternate recording medium status.

Once the system processes the command, it issues message IFB090I to the console from which the command was issued or to a specified console. Use LookAt

DISPLAY LOGREC Command

(see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books to see the description of message IFB090I, which shows the resulting display.

Restrictions

- Do not use the same keyword more than once within a single command.
- Do not exceed the maximum command line length of 124 characters.

Syntax

```
D LOGREC[,CURRENT|CURR] |{DATASET|DSN}|{ALL|A}  
[ ,L={a|name|name-a}]
```

The parameters are:

CURRENT or CURR

Indicates that the system is to display the current logrec medium. CURRENT is the default. The possible current mediums are as follows:

- LOGSTREAM, which displays the log stream name and status.
- DATASET, which displays the logrec data set name and status.
- IGNORE, which indicates that there is no logrec medium.

DATASET or DSN

Indicates that the system is to display only the logrec data set name and status. If a data set name is displayed, then it is defined. The displayed data set, however, might not be the current logrec recording medium. To determine the current recording medium, use the CURRENT option. If there is no data set defined, the system displays the text NOT DEFINED.

ALL or A

Indicates that the system is to display all, both current and alternate, logrec medium and data set names and status.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example

To display information for all the logrec medium settings:

```
DISPLAY LOGREC,ALL
```

Displaying System Configuration Information

D M=ESTOR command no longer supported:

The D M=ESTOR command is not supported on systems at z/OS V1R7 and higher.

Use the DISPLAY M command to display the status of sides, processors, ICRFs, channel paths, devices, and central storage, or to compare the current hardware configuration to the configuration in a CONFIGxx parmlib member.

The DISPLAY M command can accept the subchannel set number to qualify the input device number. The output of message IEE097I includes the applicable subchannel set number.

When you specify a device number that could be mistaken for the device name, precede the device number with a slash. The slash is optional with a 3-digit device number.

```
D M[=CHP[(xx)|(xx-xx)|(list)]
     |=CONFIG[(xx)]
     |=CPUAD|CPU][(x)|(list)]
     |=CU(xxxx)
     |=DEVICE|DEV][([/]devnum)|([/]lowdevnum-[/]highdevnum)|(list)]
     |=DEVICE|DEV|([/]devnum,(chp))
     |=DEVICE|DEV|([/]devnum),chp
     |=HIGH
     |=HSA
     |=SIDE[(id)]
     |=STORAGE|STOR|([dddmM-dddmM)|(list)|(E[=id])]
     |=SWITCH(sssss [,pp[-pp] [,pp[-pp]]...])
     |=(parm[,parm]...)

[,L={a|name|name-a}]
```

- M** The system is to display information about the system configuration. When you enter DISPLAY M with no operands, the system displays the starting address and length of each portion of the hardware system area (HSA) and the status of all processors, ICRFs, central storage, channel paths, and devices, depending on the type of processor or processor complex.

If the processor complex is partitioned, the system does not provide information about resources that are not part of the configuration on which you issue the command. Message IEE174I gives you the status of resources on the side from which you issue the command and tells you that information about the other side is unavailable. If you are running your processor complex in single-image mode with all resources in one side offline, message IEE174I identifies the other side as being offline but gives you the information about those resources. For example, to partition a processor complex, you configure offline the resources on one side. To verify that those resources are offline, issue the DISPLAY M=SIDE command. The display lists the side as offline and gives the status of the resources.

CHP

The system is to display the online and offline status of channel paths. If you do not specify any channel path, the system displays the status of all channel paths, as well as a status of either “managed and online” or “managed and offline” as part of the support of dynamic channel path management. For a description of the display format, see message IEE174I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

(xx)

A single channel path identified by *xx*. The channel path identifier can have a value from 0 to FF.

(xx-xx)

A range of channel path identifiers. The starting and ending channel path identifiers can have a value from 0 to FF.

DISPLAY M Command

(list)

One or more single channel path identifiers, or a combination of single channel path identifiers and ranges of channel path identifiers, each separated by a comma.

CONFIG[(xx)]

The system is to display the differences between the current configuration and the configuration described in parmlib member CONFIGxx. If you omit xx, the system assumes that you mean CONFIG00.

For a description of the display format, see message IEE097I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

You can also start this function from the HCD dialog. For details refer to the section “Process Display M=CONFIG(xx) Command” in *z/OS HCD User’s Guide*.

CPUAD or CPU

The system is to display the online or offline status of one or more processors and any ICRFs attached to those processors. For a description of the display format, use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books to see message IEE174I.

If you do not specify any processor identifiers, the system displays the online or offline status of all processors and any ICRFs attached to them. Whether you specify a processor identifier or not, the system displays “N” when a processor is neither online or offline, but is recognized by the machine.

Note: When you issue the DISPLAY M=CPU command from a PR/SM partition, the system displays the status for the logical processors and ICRFs defined to the partition.

(x) A single processor identified by processor identifier in hexadecimal format.

(list)

One or more processor identifiers, each separated by a comma.

CU

The system is to display the information for a specific control unit. For a description of the display format, see message IEE174I.

(xxxx)

The control unit number.

Note: The D M=CU command does not support displaying information for CTC control units.

DEVICE or DEV

The system is to display the number of online channel paths to devices or a single channel path to a single device.

For a description of the display format, see message IEE174I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

([I/]devnum)

A single device number.

([/]lowdevnum-[/]highdevnum)

The lower device number *lowdevnum* and the upper device number *highdevnum* of a range of devices.

([/]devnum,(chp))

A single device number and single channel path identifier.

(([/]devnum),chp)

A single device number and single channel path identifier.

Device numbers and ranges can be specified in any combination.

A device number consists of 3, 4, or 5 hexadecimal digits, optionally preceded by a slash (/). A channel path identifier can have a value from 0 to FF. In the 5 digit format, sddd, s is the subchannel set identifier and dddd is the device number.

HIGH

The system is to display the highest possible central storage in decimal M bytes (megabytes) or when the value is greater than 16383 decimal, in hexadecimal M bytes. Each address indicates the amount of storage available at system initialization. For a description of the display format, see message IEE174I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

HSA

The system is to display the starting address and length of each portion of the hardware system area (HSA). For a description of the display format, see message IEE174I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

SIDE[(*id*)]

The system is to display the resources installed in side (physical partition) *id*, whether the resources are online or offline, and whether the side is online, offline, or unavailable. If the processor complex is partitioned and the specified side is part of another configuration, no information is provided. If the processor complex is running in single-image mode and you do not specify an *id*, the system displays both sides. If the command is issued from MVS running in a partition, no information is provided.

For a complete description of the display format of DISPLAY M=SIDE, see message IEE174I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

STORAGE or STOR

The system is to display the status of central storage. The display includes storage offline, storage waiting to go offline and reconfigurable storage sections. For storage waiting to go offline, the system displays:

- The address space identifier (ASID)
- The jobname of the current user of the storage
- The amount of unassigned storage in offline storage elements
- The amount of storage that belongs to another configuration

STORAGE also indicates if a given range of central storage contains data that is shared through the use of the IARVSERV macro.

In this display, storage offline does not include the hardware save area (HSA). To find the location and length of the HSA, enter DISPLAY M=HSA.

If you do not specify (*ddddddX-ddddddX*), (*list*), or (E[=*id*]), the system displays the status of all central storage. For a description of the display

DISPLAY M Command

format, see message IEE174I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

(*ddddddX-dddddxX*)

The starting and ending addresses of a range in central storage for which you want the status display. Specify up to five decimal digits followed by a multiplier (M-megabytes, G-gigabytes, T-terabytes, P-petabytes) for each address. The starting and ending addresses (*ddddddX*) must each be on a valid storage boundary and cannot exceed 16383P. The starting and ending addresses must not be the same.

Instead of specifying the range using decimal numbers, you may specify it in hexadecimal, with or without a multiplier, in the format X'xxxxxx'-X'xxxxxx'. For example:

- X'123456789A00000'-X'123456789B00000'
- X'123'M-X'124'M

You may use underscores in any hexadecimal specification for better clarity. Underscores in the specification are ignored during processing.

(*list*)

One or more address ranges (in decimal), each separated by a comma.

(E[=id])

The system is to display the status of the requested storage element. The display includes the amount of storage (in megabytes) the system owns in each online storage element, the amount of storage available to be configured online, whether the storage element is online or offline. If you omit the *id*, the system displays this information for all installed storage elements.

Note: If the processor complex is partitioned and the specified storage element is part of another configuration, no information is provided.

SWITCH(*ssss [,pp[-pp] [,pp[-pp]]...]*)

The system is to display the status of a specific switch, switch port, or list of switch ports.

For a description of the display format, see message IEE174I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

ssss

The device number of the switch device.

[,pp[-pp] [,pp[-pp]]...]

The port address or port address list.

(*parm[,parm]...*)

The system is to display the status of each resource you specify as *parm*. The list of *parms* you specify within the parentheses may contain any combination of CHP, CPU, DEV, HIGH, HSA, STOR(E[=id]), and STOR. You must separate the resources in the list with commas and you must enclose the list in parentheses. Do not use blanks within the parentheses and do not try to specify CONFIG in the list.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display the online or offline status of all devices on channel path 01, enter:

D M=CHP(01)

Example 2

To display the following:

- The online or offline status of all processors
- The number of online channel paths to each device
- The highest central storage address available
- The status of central storage

enter:

D M=(CPU,DEV,HIGH,STOR)

Example 3

To display the number of megabytes of storage the system owns in storage element 0 and the status of the storage element, enter:

D M=STOR(E=0)

Example 4

To display the number of megabytes of storage the system owns in each storage element and the status of each element, enter:

D M=STOR(E)

Example 5

To display the status of all processors, the status for channel paths 1, 3, 4, 5, and the high storage addresses for central storage, enter:

D M=CPU
D M=CHP(01,03-05)
D M=HIGH
or
D M=(CPU,CHP(01,03-05),HIGH)

Displaying MVS Message Service Status and Languages

Use the DISPLAY MMS command to request a display of the current status of the MVS message service (MMS) and a display of the languages that are currently available.

The complete syntax for the DISPLAY MMS command is:

D MMS[,L={a|name|name-a}]

DISPLAY MMS Command

MMS

Displays the status of the MVS message service and the list of the languages that are currently available.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example

To display the status of the MVS message service and the current languages, enter:

D MMS

The status display is in the following format:

```
CNLS026I 13.25.02 MMS DISPLAY
PARMLIB MEMBER = MMSLSTE2
LAST REFRESH WAS AT 10:21 on 04/24/89
CODE  CONFIG      OBJECT
ENU   CNLENU01    SYS1.MSG.ENG
JPN   CNLJPN02    SYS1.MSG.JAPAN
DEU   CNLDEU01    SYS1.MSG.GERMAN
EXIT 01 -  OURMMS1
```

When the MVS message service is not active, the system issues message IEE294I.

Displaying Message Suppression, Retention, Color, Intensity, and Highlighting Options

Use the DISPLAY MPF (message processing facility) command to display information about message processing and presentation that is contained in the MPFLSTxx parmlib member or members currently in effect.

The MPF parameter on the INIT statement in the CONSOLxx parmlib member or the SET MPF=xx command activates and deactivates an MPFLSTxx member or members.

The syntax of the DISPLAY MPF command is:

```
D MPF[,{MSG|M}][,L={a|name|name-a}]
     |,{COLOR|C}
     |,CMD
```

MPF

The system is to display information about message processing and presentation. If you do not use operands on the DISPLAY MPF command, the system displays the following information:

- Which messages are being suppressed by MPF
- Which action message are not being retained by the action message retention facility
- Which installation exits receive control for selected messages
- The status of the general WTO installation exit IEAVMXIT

- Whether this message is automated by MPF
- The MPFLSTxx member that identifies the message ID, color attribute, or command installation exit definition
- What color, intensity, and highlighting capabilities are in effect
- The status of the command installation exit routines specified in parmlib member MPFLSTxx
- The current installation options for handling foreign messages

|
Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books to see a description of the output in message IEE677I.

MSG or M

The system is to display information on all messages that are defined in the current MPFLSTxx member:

- Which messages are being suppressed by MPF
- Which action messages are not being retained by the action message retention facility
- Which installation exits receive control for selected messages
- The status of the general WTO installation exit IEAVMXIT
- Whether this message is automated by MPF
- The MPFLSTxx member is automated by MPF
- The MPFLSTxx member that identifies the message ID

COLOR or C

The system is to display:

- What color, intensity, and highlighting capabilities are in effect

CMD

The system is to display:

- The status of the command installation exit routines specified in parmlib member MPFLSTxx

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display information about the message IDs and WTOR installation exits for all consoles, enter:

D MPF,M

Example 2

To display the color, intensity, and highlighting specifications for all consoles, enter:

D MPF,C

Example 3

DISPLAY MPF Command

To display the message processing and presentation information for all consoles on console 2, area A, enter:

```
D MPF,L=CON2-A
```

Example 4

To display the status of the command installation exits, enter:

```
D MPF,CMD
```

Displaying z/OS UNIX System Services Status

The MVS operator can use the DISPLAY command to obtain:

- z/OS UNIX System Services status information (for example, active or terminating, shutting down or restarting).
- File system information.
- z/OS UNIX System Services process information for address spaces, including an indication of which processes are registered as permanent or blocking.
- The current setting for all statements in the BPXPRMxx member (or members) of SYS1.PARMLIB, which is used by z/OS UNIX. This includes system-wide and process limits, their high-water marks, and current usage.
- Information about multiple parmlib members.
- Information about each physical file system that is currently part of the z/OS UNIX System Services configuration.
- Routing information from the common inet (CINET) prerouter tables.
- Thread-level information for any thread that is in a byte-range lock wait.

You can use this command to display address space information for a user who has a process that is hung or that is waiting to become a process. You can also use the information returned from this command to determine how many address spaces a given TSO/E user ID is using, whether an address space is using too many resources, and whether a user's process is waiting for a z/OS UNIX kernel function to complete.

The syntax for the DISPLAY OMVS command is:

```
D OMVS[ {,SUMMARY|S} ]  
  ,{ASID|A}=ALL  
  ,{ASID|A}=asid  
  ,{ASID|A}=DUBW  
  ,U=userid  
  ,{PID}=processid[,BRL]  
  ,{FILE|F[,{NAME|N}=filesystem ][,CAPS|C} }  
    ,{,OWNER|O}=systemname  
    ,{,EXCEPTION|E}  
    ,{,TYPE|T}=type]  
  ,{VSERVER|V}  
  ,{PFS|P}  
  ,{CINET|CI}=All|TPname  
  ,{OPTIONS|O}  
  ,{LIMITS|[,{PID=ProcessId][,RESET]}  
  ,[MEMLIMIT[=maxmemlimit  
  ,{SER}  
  ,ACTIVATE=SERVICE      |,{WAITERS|W}  
  ,MF[{=ALL|A}]  
  ,MF={PURGE|P}]  
  [,L={a|name|name-a}]  
  {SOCKETS|SO}
```

SUMMARY or S

Displays status of z/OS UNIX processes, file systems, and servers (for example, active or terminating) and the BPXPRMxx parmlib member specified during initialization or specified by the SET OMVS= z/OS UNIX System Services command.

ASID or A=ALL

Displays process information for all z/OS UNIX System Services address spaces.

ASID or A=asid

Displays process information for the specified hexadecimal address space ID (ASID). If the specified ASID is not a z/OS UNIX System Services address space, an error message is issued.

ASID or A=DUBW

Displays process information for all address spaces waiting to be dubbed a z/OS UNIX System Services process. After message BPXP022E is issued to indicate one or more jobs are waiting for z/OS UNIX System Services availability, you can issue D OMVS,A=DUBW to display all jobs waiting to be dubbed.

U=userid

Displays process information for all processes associated with the specified TSO/E user ID. Use this operand when a user requests that a hung process be canceled. You can display all processes owned by the user and find the address space ID (ASID) of the process that needs to be canceled. Then use the CANCEL command to cancel the address space.

PID=processid

Displays thread information for the processid that is specified in decimal numbers. In a sysplex environment, you must issue the D OMVS,PID= command from the system on which the specified process is running. See “Example 13” on page 4-186.

BRL

Displays thread-level information for any thread that is in a byte-range lock wait. You can specify this operand with the PID operand. See “Example 13” on page 4-186.

FILE or F

Displays a list of file systems that z/OS UNIX System Services is currently using, including the following:

- The status of each file system.
- The date and time that the file system was mounted.
- The latch number for the file system.
- The quiesce latch number for the file system, or 0 if the file system has never been quiesced by UNIX System Services.

You can limit the amount of information displayed by specifying one of the following keywords:

NAME or N=filesystem

Displays information about the specified file system or file systems. You can use one wildcard character (*) in the file system specified. For example, ZOS18.*.HFS or ZOS.L*.HFS. Note that specifying D OMVS,F,NAME=* results in the system displaying all file systems, which is the same output as if you specified D OMVS,F.

DISPLAY OMVS Command

OWNER or O=systemname

Displays information for the file systems owned by the specified system name. Specifying D OMVS,F,OWNER displays all the file systems that are owned by this system.

EXCEPTION or E

Displays file systems in an exception state, such as a file system that is quiesced, unowned, or in recovery.

TYPE or T=type

Displays all file systems of the specified PFS type.

CAPS or C

Displays variable data containing lowercase letters in uppercase.

VSERVER or V

Displays process information for all processes that have been defined as servers that use the virtual file system (VFS) callable services API.

CINET or CI = ALLtpname

Displays the Common Inet routing information for all of the active transport providers in use by the common inet prerouter. The transport providers were specified with the SUBFILESYSTYPE statements in the BPXPRMxx profile or specified with the SETOMVS command. The network routing information was specified in the appropriate data set for the transport provider. When the name (*tpname*) of an active transport provider is specified, the command displays the Common Inet routing information for that specific transport provider.

OPTIONS or O

Displays the current settings of the options that

- (a) were set during initialization in the parmlib member BPXPRMxx or by a SET OMVS or SETOMVS command after initialization, and that
- (b) can be altered dynamically via a SET OMVS or SETOMVS command.

Note that if you issue the D OMVS,O command while OMVS is shutdown, the system will attempt to display the OMVS parmlib options that were last in effect when OMVS was active. However, it is possible that some option values are unavailable and may not have values displayed.

PFS or P = Physical File System

Displays information about each physical file system that is currently part of the z/OS UNIX System Services configuration. The physical file systems were specified in the BPXPRMxx profile, or with the SETOMVS command, or are an internal part of z/OS Unix System Services.

LIMITS or L

Displays information about current z/OS UNIX System Services parmlib limits, their high-water marks, and current system usage. When the PID= keyword is specified, LIMITS displays high-water marks and current usage for an individual process.

MEMLIMIT=maxmemlimit

Displays the maximum amount (*maxmemlimit*) of allocated, non-shared, 1-megabyte storage segments above the bar allowed for the address space. This command displays the value as nnnnnnnnnnC, where C is the associated denomination (or multiplier) value as set by the SETOMVS command as shown in Table 4-17 on page 4-177.

Table 4-17. Denomination (multiplier) character used for various OMVS commands

Denomination Value	1-Character Abbreviation	Bytes
null	n/a	1
Kilo	K	1,024
Mega	M	1,048,576
Giga	G	1,073,741,824
Tera	T	1,099,511,627,776
Peta	P	1,125,899,906,842,624

RESET

Resets the high-water mark for a system limit to 0.

SER

Reports serialization for all in-use, shared memory mutexes (mutual exclusion locks) and condition variables. Each mutex and condition variable is identified by the shared memory ID and the location of the shared memory object.

If the object is in an above-the-bar shared-memory segment, the location information indicates the address of the mutex or condition variable. If it is in a below-the-bar segment, the location information indicates the offset within the shared-memory segment. The offset is displayed, in this case, because each address space sharing a below-the-bar segment can map it at a different virtual address. For each mutex, the output shows the owner's TCB address, process ID, and ASID and the same orf those waiting for access, if the system can determine that information.

For each condition variable, the output shows the same information for the waiting task of the condition variable and additionally identifies the associated mutex. User data is displayed for each owner and waiting task of a mutex or condition variable. In the case where LE is the caller of BPX1SMC, the user data represents the address of the LE DSA data area for the waiting or owning task.

ACTIVATE=SERVICE

Specifies that all the dynamically activated service items are to be displayed. Dynamically activated service consists of SMP/E installable service for the z/OS UNIX kernel and logical file system (LFS) components that was activated with the F OMVS,ACTIVATE=SERVICE command. (See “Recycling z/OS UNIX System Services (z/OS UNIX)” on page 4-333).

The service items are displayed in the order they were activated, with the most recent set of activated service items being displayed first. The most recent set of service items, which are shown as the highest numbered set of service items, are the highest level of service items activated for z/OS UNIX.

The display includes the following information:

- The library and volume from which each set of service was activated.
- The amount of ECSA and OMVS address space storage consumed by all dynamically activated service items. Note that the amount of storage consumed will not decrease if you deactivate service items (F OMVS,DEACTIVATE=SERVICE), because the modules containing the deactivated service items remain in storage. See “Example 15” on page 4-188.

This command will not display deactivated service items.

DISPLAY OMVS Command

WAITERS or W

Displays information about delays caused by the following conditions:

- Mount latch contention
- Outstanding unprocessed sysplex messages
- File system latch contention
- Other reasons

You can use the information displayed to figure out which tasks are hung, and why they are waiting. See "Example 16" on page 4-188

MF

MF=ALL I A

Displays information about move or mount failures:

- Enter MF to display information about the last 10 or less move or mount failures.
- Enter MF=ALL or MF=A to display information about the last 50 or less move or mount failures.

The system issues message BPXO058I to display the information about mount failures. See "Example 17" on page 4-190.

MF=PURGE I P

Allows you to purge the saved information about mount failures displayed in message BPXO058I.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

SOCKETS I SO

Displays the following information about each AF_UNIX socket:

- The owner's userid
- The socket name
- The name of the peer socket

See "Example 18" on page 4-190.

Example 1

To display process information for all z/OS UNIX System Services address spaces, enter:

```
DISPLAY OMVS,A=ALL
```

z/OS UNIX System Services status information (OMVS ACTIVE) appears before the process information. See the appropriate UNIX System Services book for an explanation of the data filled in below the headers, such as a state of MKI.

```
BPX0040I 14.31.40 DISPLAY OMVS 018
OMVS      000E ACTIVE          OMVS=(93)
USER      JOBNAME  ASID      PID      PPID STATE   START     CT_SECS
IBMUSER   BPXOINIT 0013      1        0 MKI    11.02.40     .037
LATCHWAITPID= 0 CMD=BPXPINPR
 SERVER=Init Process          AF=      0 MF=65535 TYPE=FILE
MEGA      MEGA    001A    16777218      1 1RI    11.18.17     .634
LATCHWAITPID= 0 CMD=OMVS
MEGA      MEGA    001A    16777219  16777218 1CI    11.18.25     .634
LATCHWAITPID= 0 CMD=sh -L
```

Example 2

To display z/OS UNIX System Services process information on all z/OS UNIX System Services address spaces owned by user ID MEGA, enter:

```
DISPLAY OMVS,U=MEGA
```

z/OS UNIX System Services status information (OMVS ACTIVE) appears before the process information.

```
BPX0040I 14.34.15 DISPLAY OMVS 021
OMVS    000E ACTIVE          OMVS=(93)
USER    JOBNAME ASID        PID      PPID STATE   START     CT_SECS
MEGA    MEGA    001A    16777218      1 1RI    11.18.17    .634
      LATCHWAITPID=      0 CMD=OMVS
MEGA    MEGA    001A    16777219  16777218 1CI    11.18.25    .634
      LATCHWAITPID=      0 CMD=sh -L
```

Example 3

To display z/OS UNIX System Services process information for the address space with ASID equal to 001A, enter:

```
DISPLAY OMVS,ASID=1A
```

z/OS UNIX System Services status information (OMVS ACTIVE) appears before the process information.

```
BPX0040I 14.36.04 DISPLAY OMVS 024
OMVS    000E ACTIVE          OMVS=(93)
USER    JOBNAME ASID        PID      PPID STATE   START     CT_SECS
MEGA    MEGA    001A    16777218      1 1RI    11.18.17    .634
      LATCHWAITPID=      0 CMD=OMVS
MEGA    MEGA    001A    16777219  16777218 1CI    11.18.25    .634
      LATCHWAITPID=      0 CMD=sh -L
```

Example 4

To display detailed file system information on currently mounted files, enter:

```
DISPLAY OMVS,FILE
```

z/OS UNIX System Services status information appears before the file system information.

```
d omvs,f
BPX0045I 11.40.13 DISPLAY OMVS 217
OMVS    000E ACTIVE          OMVS=(AW)
TYPENAME DEVICE -----STATUS----- MODE  MOUNTED  LATCHES
HFS      6 ACTIVE           READ   04/08/11  L=18
      NAME=ZOS17.LPP.HFS
      PATH=/usr/lpp
HFS      5 ACTIVE           READ   04/08/11  L=17
      NAME=ZOS17.NLS.HFS
      PATH=/usr/lib/nls
HFS      4 ACTIVE           READ   04/08/11  L=16
      NAME=ZOS17.MAN.HFS
      PATH=/usr/man
HFS      3 ACTIVE           RDWR  04/08/11  L=15
      NAME=ZOS17.VAR.HFS
      PATH=/SYSTEM/var
HFS      2 ACTIVE           RDWR  04/08/11  L=14
      NAME=ZOS17.ETC.HFS
```

DISPLAY OMVS Command

```
PATH=/SYSTEM/etc
HFS          1 ACTIVE
NAME=ZOS17.ROOT.HFS
PATH=/
```

RDWR 04/08/11 L=13
11.38.19 Q=0

If AUTOMOVE was specified with a system list in the SETOMVS or chmount command, the list of candidates to take ownership of the file system is displayed under the file system owner.

For zFS file systems, the display includes an aggregate file system name indicating membership in a data set containing multiple file systems. Aggregates provide member file systems with a common pool of disk space.

Note: File systems can have a status of NOT ACTIVE if they were mounted under a physical file system (PFS) that has terminated, such as is possible with zFS, TFS, or NFS Client. These file systems cannot be made active again without unmounting and remounting them after the PFS is restarted. If the file systems are remounted, they will appear active with a new device number.

Example 5

To display process information for all processes that have been defined as a server, enter:

```
DISPLAY OMVS,V
```

z/OS UNIX System Services status information (OMVS ACTIVE) appears before the file system information.

```
BPX0040I 14.38.46 DISPLAY OMVS 030
OMVS    000E ACTIVE      OMVS=(93)
USER    JOBNAME ASID    PID      PPID STATE   START   CT_SECSS
IBMUSER BPXINIT 0013    1        0 MKI    11.02.40   .0373
LATCHWAITPID= 0 CMD=BPXPINPR
SERVER=Init Process     AF=      0 MF=65535 TYPE=FILE
```

Example 6

To display all options set during initialization by the parmlib member BPXPRMxx or with the SET command, enter:

```
DISPLAY OMVS,O
d omvs,o
BPX0043I 13.10.16 DISPLAY OMVS 066
OMVS    000D ETC/INIT WAIT  OMVS=(M7)
CURRENT UNIX CONFIGURATION SETTINGS:
MAXPROCSYS = 256 MAXPROCUSER = 16
MAXFILEPROC = 256 MAXFILESIZE = NOLIMIT
MAXCPUTIME = 1000 MAXUIDS = 200
MAXPTYS = 256
MAXMMAPAREA = 256 MAXASSIZE = 209715200
MAXTHREADS = 200 MAXTHREADTASKS = 1000
MAXCORESIZE = 4194304 MAXSHAREPAGES = 4096
IPCMMSGQBYTES = 2147483647 IPCMSGQNUM = 10000
IPCMMSGNIDS = 500 IPCSEMNIDS = 500
IPCSEMNOPS = 25 IPCSEMNSEMS = 1000
IPCSHMMPAGES = 25600 IPCSHMNIDS = 500
IPCSHMNSEGS = 500 IPCSHMPAGES = 262144
SUPERUSER = BPXROOT FORKCOPY = COW
STEPLIBLIST =
USERIDALIASTABLE=
```

```

SERV_LINKLIB = POSIX.DYNSEVR.LOADLIB BPXLK1
SERV_LPALIB = POSIX.DYNSEVR.LOADLIB BPXLK1
PRIORITYPG VALUES: NONE
PRIORITYGOAL VALUES: NONE
MAXQUEUEDSIGS = 1000 SHRLIBRGNsize = 67108864
SHRLIBMAXPAGES = 4096 VERSION = /
SYSCALL COUNTS = NO TTYGROUP = TTY
SYSPLEX = NO BRLM SERVER = N/A
LIMMSG = NONE AUTOCVT = OFF
RESOLVER PROC = DEFAULT
AUTHPGMLIST = NONE
SWA = BELOW

```

Note: The SYSPLEX (YES) option indicates the system is in a sysplex and is using the shared file system capability. You cannot dynamically change the SYSPLEX parameter through SETOMVS or SET OMVS. For more information, see the chapter on shared file system in *z/OS UNIX System Services Planning*.

To display the current setting of the options that were set during initialization by the parmlib member BPXPRM93 or with the SET OMVS or SETOMVS command and that can be altered dynamically by either of those commands, enter:

```

DISPLAY OMVS,0
BPX0043I 11.08.44 DISPLAY OMVS 962
OMVS 000E ACTIVE OMVS=(93)
z/OS UNIX CURRENT CONFIGURATION SETTINGS:
MAXPROCSYS = 256 MAXPROCUSER = 16
MAXFILEPROC = 256 MAXFILESIZE = NOLIMIT
MAXCPUTIME = 1000 MAXUIDS = 32
MAXRTYS = 256 MAXPTYS = 256
MAXMMAPAREA = 4096 MAXASSIZE = 41943040
MAXTHREADS = 200 MAXTHREADTASKS = 50
MAXCORESIZE = 4194304 MAXSHAREPAGES = 131072
IPCMMSGQBYTES = 262144 IPCMSGQNUM = 10000
IPCMMSGNIDS = 500 IPCSEMNIDS = 500
IPCSEMNOPS = 25 IPCSEMNSEMS = 25
IPCSHMMPPAGES = 256 IPCSHMNIDS = 500
IPCSHMNSEGS = 10 IPCSHMSPAGES = 262144
SUPERUSER = BPXROOT FORKCOPY = COW
STEPLIBLIST =
USERIDALIASABLE=
PRIORITYGOAL VALUES: NONE
MAXQUEUEDSIGS = 1000
SYSCALL COUNTS = NO TTYGROUP = TTY
AUTHPGMLIST = /etc/authfile

```

Example 7

To display the thread information for process id 1, enter:

```

DISPLAY OMVS,PID=1
BPX0040I 11.13.40 DISPLAY OMVS 971
OMVS 000E ACTIVE OMVS=(93)
USER JOBNAME ASID PID PPID STATE START CT_SECS
IBMUSER BPX0INIT 0013 1 0 MKI 11.02.40 .037
LATCHWAITPID= 0 CMD=BPXPINPR
SERVER=Init Process AF= 0 MF=65535 TYPE=FILE
THREAD_ID TCB@ PRI_JOB USERNAME ACC_TIME SC STATE
04B9267800000000 009DEA70 OMVS .028 WAT W
04B92F2000000001 009DE8D8 .003 VRT Y
04B937C800000002 009DE278 OMVS .002 KIN K

```

DISPLAY OMVS Command

Example 8

To display information about each physical file system that is currently part of the z/OS UNIX System Services configuration when the physical file systems are specified in the BPXPRMxx profile, enter:

D OMVS,P

```
BPX0046I 14.35.38 DISPLAY OMVS 092
OMVS      000E ACTIVE          OMVS=(33)
PFS CONFIGURATION INFORMATION
  PFS TYPE      DESCRIPTION          ENTRY      MAXSOCK   OPNSOCK  HIGHUSED
    TCP      SOCKETS AF_INET        EZBPFINI  50000     244      8146
    UDS      SOCKETS AF_UNIX        BPXTUINT  64         6        10
    HFS      LOCAL FILE SYSTEM    GFUAINIT
    BPXFTCLN  CLEANUP DAEMON     BPXFTCLN
    BPXFTSYN  SYNC DAEMON       BPXFTSYN
    BPXFPINT  PIPE              BPXFPINT
    BPXFCIN   CHAR SPECIAL      BPXFCIN
    NFS      REMOTE FILE SYSTEM  GFSCINIT

  PFS NAME      DESCRIPTION          ENTRY      STATUS    FLAGS
    TCP41     SOCKETS            EZBPFINI  ACT       CD
    TCP42     SOCKETS            EZBPFINI  ACT
    TCP43     SOCKETS            EZBPFINI  INACT    SD
    TCP44     SOCKETS            EZBPFINI  INACT

PFS PARM INFORMATION
  HFS      SYNCDEFAULT(60) FIXED(50) VIRTUAL(100)
           CURRENT VALUES: FIXED(55) VIRTUAL(100)
  NFS      biod(6)
```

The information displayed is:

PFS TYPE

For each FILESYSTYPE statement, the data specified with the TYPE operand is displayed.

PFS DESCRIPTION

A brief description of the physical file system.

ENTRY

The name of the load module specified with the ENTRYPOINT operand on the FILESYSTYPE or SUBFILESYSTYPE statements.

MAXSOCK

This is the MAXSOCKETS operand of a NETWORK statement for a sockets physical file system. It specifies the maximum number of sockets that can be open at one time for the address family.

OPNSOCK

OPEN SOCKETS: The number of sockets that are currently opened for this sockets physical file system.

HIGHUSED

The highest number of sockets that have been in use at one time for each of the configured address families.

PFS NAME

For each SUBFILESYSTYPE statement, the transport provider specified with the NAME operand is displayed.

STATUS

The status of each PFS specified with the SUBFILESYSTYPE statement: ACT = ACTIVE, INACT = INACTIVE.

FLAGS

Additional information for each PFS that was defined with the SUBFILESYSTYPE statement:

- CD** Current Default transport provider. The system is currently using this PFS as the default transport provider although it wasn't specified as the default with the SUBFILESYSTYPE statement.
- SD** Specified Default transport provider. This PFS was specified as the default transport provider with the SUBFILESYSTYPE statement. Currently, however, it is not being used as the default.
- SC** Specified Current default transport provider. This PFS was specified as the default transport provider with the SUBFILESYSTYPE statement and the system is currently using it as the default.

PARM INFORMATION

Data specified with the PARM operand on the FILESYSTYPE or SUBFILESYSTYPE statements is displayed. For the file system, in addition to the IPL settings specified with PARM, the current settings for the FIXED and VIRTUAL PARMs are displayed.

Notes:

1. Although you may specify up to 1024 bytes of parameter information in the BPXPRMxx profile, only the first 165 bytes of parameter information is displayed.
2. If a dash ('-') should appear as the first character for any PFS name, it means the PFS is dead.

Example 9

To display the Common Inet routing information when there are three active transport providers, enter the following. If internet protocol version 6 (IPv6) is in use, 16-byte IP addresses will display where appropriate. IPv6 data displays after IPv4 data.

```
D OMVS,CINET
BPX0047I 12.01.33 DISPLAY OMVS 285
OMVS      000E ACTIVE          OMVS=(QY)
IPV4 HOME INTERFACE INFORMATION
TP NAME   HOME ADDRESS      FLAGS
TCPIPZ1   001.001.001.001
TCP1      003.003.003.003
TCPIPZ1   006.007.008.009
TCP1      044.044.044.044

IPV4 HOST ROUTE INFORMATION
TP NAME   HOST DESTINATION    METRIC
TCPIPZ1   001.001.001.001      0
TCP1      003.003.003.003      0
TCP1      127.000.000.001      0
TCPIPZ1   127.000.000.001      0

IPV4 NETWORK ROUTE INFORMATION
TP NAME   NET DESTINATION     NET MASK      METRIC
TCPIPZ1   001.000.000.000     255.000.000.000      0
TCP1      003.000.000.000     255.000.000.000      0

IPV6 HOME INTERFACE INFORMATION
TP NAME   HOME ADDRESS      FLAGS
TCP1      0022:0022:0022:0022:0022:0022:0022:0022      DRS
TCPIPZ1   0000:0000:0000:0000:0000:0000:0000:0009      DRS
TCP1      0021:0021:0021:0021:0021:0021:0021:0021      DRS
```

DISPLAY OMVS Command

IPV6 HOST ROUTE INFORMATION		
TP NAME	HOST DESTINATION	METRIC
TCP1	0022:0022:0022:0022:0022:0022:0022:0022	0
TCP1	0001:0000:0000:0000:0000:0000:0000:0005	0
TCPIPZ1	0000:0000:0000:0000:0000:0000:0000:0008	0
TCP1	0000:0000:0000:0000:0000:0000:0000:0001	0
TCPIPZ1	0000:0000:0000:0000:0000:0000:0000:0001	0
TCPIPZ1	0000:0000:0000:0000:0000:0000:0000:0009	0
TCP1	0001:0000:0000:0000:0000:0000:0000:0007	0
TCP1	0021:0021:0021:0021:0021:0021:0021:0021	0

IPV6 NETWORK ROUTE INFORMATION		
TP NAME	NET DESTINATION	METRIC
TCPIPZ1	432B:0055:0066:0099:0099:0033:0000:0000/090	0
TCPIPZ1	0000:0000:0099:0044:0055:0077:0099:0066/060	0
TCP1	0000:0000:002E:002E:002E:002E:002E:056	0
TCPIPZ1	0000:0000:0011:0014:0014:0013:0013:0013/090	0
TCPIPZ1	0000:0000:0031:0031:0031:0031:0043:0044/056	0
TCP1	0000:0000:002B:002B:002B:002B:002B:090	0
TCP1	0000:0000:002D:002D:002D:002D:002D:060	0
TCP1	0000:0000:002C:002C:002C:002C:002C:090	0
TCP1	0000:0000:002F:002F:002F:002F:002F:100	0
TCPIPZ1	0000:0000:0033:0033:0033:0033:0033:100	0

The information displayed is:

TP NAME

The name of the transport provider for which the information is being displayed.

HOME ADDRESS

The internet protocol (IP) address of the transport provider. 16-bytes display for IPv6.

HOST DESTINATION

When a transport provider is connected to a host, the host IP address is displayed. 16-bytes display for IPv6.

NET DESTINATION

When a transport provider supplies network routing information to the Common Inet Pre-Router, the network destination address is the IP address of a network that can be accessed through the transport provider. 16-bytes display for IPv6. For IPv6, a PREFIX LENGTH follows a slash at the end of the net destination IP address. This value specifies how many of the leftmost contiguous bits comprise the prefix.

NET MASK

A mask that is applied to destination IP addresses to separate the network number from the host number.

METRIC

When selecting a route, if two transport providers can access the same route, the Common INET (CINET) Prerouter selects the route with the best metric. The lower the number, the better the metric. The metric 0 = a direct connection.

FLAGS

DRS = Default Routes Supported: When the Common Inet Pre-Router cannot find a specified IP address in its routing tables, it passes the request to a transport provider that supports default routes. If no transport provider supports default routes, the request is rejected with **ENETUNREACH**.

Note: When the cinet is not installed, similar routing information can be obtained by using the **netstat TC *tpname* gate** command or the **onetstat -p *tpname* -r** command.

Example 10

To display information about current system-wide parmlib limits, enter:

```
DISPLAY OMVS,L
```

```
BPX0051I 14.05.52 DISPLAY OMVS 904
OMVS    0042 ACTIVE          OMVS=(69)
SYSTEM WIDE LIMITS:      LIMMSG=SYSTEM
                         CURRENT HIGHWATER   SYSTEM
                         USAGE   USAGE     LIMIT
MAXPROCSYS           1        4       256
MAXUIDS              0        0       200
MAXPTYS              0        0       256
MAXMMAPAREA          0        0       256
MAXSHAREPAGES        0        10      4096
IPCMMSGNIDS         0        0       500
IPCSEMNIDS          0        0       500
IPCSHMNIDS          0        0       500
IPCSHMSPAGES        0        0      262144 *
IPCMMSGQBYTES       ---      0      262144
IPCMMSGQNUM          ---      0      10000
IPCSHMPAGES          ---      0       256
SHRLIBRGNSIZE        0        0      67108864
SHRLIBMAXPAGES       0        0       4096
```

An * displayed after a system limit indicates that the system limit was changed via a SETOMVS or SET OMVS= command.

Note: Although IPCMSGQBYTES, IPCMSGQNUM, and IPCSHMPAGES are displayed in the output of the D OMVS,L command, these resources are not monitored and no resource messages are issued.

Example 11

To display information about current parmlib limits for a process with a PID of 33554434, enter:

```
DISPLAY OMVS,L,PID=33554434
```

```
d omvs,l,pid=33554434
BPX0051I 14.06.49 DISPLAY OMVS 907
OMVS    0042 ACTIVE          OMVS=(69)
USER    JOBNAME ASID        PID      PPID STATE   START     CT_SECS
WELLIE1 WELLIE1 001C      33554434      1 IRI    14.04.38     .015
LATCHWAITPID=          0 CMD=EXEC
PROCESS LIMITS:      LIMMSG=SYSTEM
                         CURRENT HIGHWATER   PROCESS
                         USAGE   USAGE     LIMIT
MAXFILEPROC          0        1      256,1000
MAXFILESIZE          ---      ---      NOLIMIT
MAXPROCUSER          1        4       16
MAXQUEUEDSIGS        0        0      1000
MAXTHREADS           0        0       200
MAXTHREADTASKS       0        0       50
IPCSHMNSEGS          0        0       10
MAXCORERESIZE        ---      ---      4194304,NOLIMIT
```

An * displayed after a process limit indicates that the limit was changed, either directly, with a SETOMVS,PID= command; or indirectly, by a global change of this value with a SETOMVS command.

DISPLAY OMVS Command

The values displayed are in the same units as the values used in the SETOMVS command. For example, MAXFILESIZE is displayed in units of 4KB.

Notes:

1. Although MAXFILESIZE and MAXCORESIZE are displayed in the output, their current and high-water usage are not monitored, and no resource messages are issued for these resources.
2. The MAXPROCUSER limit is based on UID, as opposed to PID, value. The current and high-water usage values reflect all values for all processes that have the same UID as the UID for the specified PID.
3. For UID=0, there is no limit on MAXPROCUSER. When the **PID=** value in the DISPLAY command is for a process with UID=0, the process limit appears as unlimited. For example:

```
MAXPROCUSER          4           11           NOLIMIT
```

4. MAXCORESIZE, MAXFILESIZE, and MAXFILEPROC each have hard and soft limits. (See the documentation for the C-RTL function **setrlimit()** in *z/OS XL C/C++ Run-Time Library Reference*.) When the hard and soft limits are the same, only one value is displayed. When the limits are different, both values are displayed: first the soft limit and then the hard limit, separated by a comma.

In the preceding example, MAXFILEPROC has a hard limit of 1000 and a soft limit of 256. For MAXFILESIZE, the soft limit is equal to the hard limit and is unlimited. For MAXCORESIZE, the soft limit is 4,194,304 and the hard limit is unlimited.

Example 12

If the SETOMVS command is issued to change the value of MAXFILEPROC to 256, the information displayed is:

	CURRENT USAGE	HIGHWATER USAGE	PROCESS LIMIT
MAXFILEPROC	0	0	256 *
.			
.			
.			

If the process changes its soft limit for MAXFILEPROC to 100 (using the **setrlimit()** function), the information displayed is:

	CURRENT USAGE	HIGHWATER USAGE	PROCESS LIMIT
MAXFILEPROC	0	0	100,256
.			
.			
.			

Example 13

To display thread-level information for any thread that is in a byte-range lock wait. enter:

```
D OMVS,PID=16777219,BRL
BPX0040I 13.50.54 DISPLAY OMVS 042
OMVS      000E ACTIVE             OMVS=(99)
USER      JOBNAME   ASID      PID      PPID STATE   START     CT_SECS
WELLIE0  WELLIE0  0015    16777219  16777218 1CI   14.11.53      .703
LATCHWAITPID=          0 CMD=sh -L
```

```

THREAD_ID      TCB@     PRI_JOB  USERNAME   ACC_TIME SC STATE
250640E00000002 009C8550 OMVS           .124 RED C
BRLWAIT DEV=00000001 INO=0000002E FILE=PoughkeepsiePho+ PID=12345678
The blocking process is on system: SY2

```

The information displayed in BRLWAIT DEV=00000001 INO=0000002E FILE=PoughkeepsiePho+ PID=12345678 is:

DEV

The device number of the file's mounted file system.

INO

The inode number of the file, as shown by **ls -i**.

FILE

Up to 16 characters of the filename of the file that is being locked. If the filename has more than 16 characters, the first 15 are displayed, followed by a plus sign (+).

PID

The process ID of another process that is blocking this process from obtaining the lock. Usually this is the owner (or one of the owners) of a lock on the same range, but sometimes it is another process that is also waiting.

The system also displays the name of the system for where the blocking process is, when the following conditions occur:

- The command is issued in a sysplex configuration.
- The blocking process is from a system in the sysplex that is different from the system where the command is issued.

Example 14

To display a report of serialized, in-use, shared memory mutexes and condition variables, enter:

```

D OMVS,SER
BPX0057I 08.51.42 DISPLAY OMVS 284
OMVS      000E ACTIVE          OMVS=(6D)
                      UNIX SERIALIZATION REPORT
NO RESOURCE CONTENTION EXISTS

```

The output from D OMVS,SER will be as follows when there is contention:

```

BPX0057I 08.51.42 DISPLAY OMVS 284
OMVS      000E ACTIVE          OMVS=(6D)
                      UNIX SERIALIZATION REPORT
RESOURCE    #1

NAME=SHARED MUTEX  DATA: SHMID=00000648 OFFS/ADDR=0000000000002428
JOBNAME ASID TCB      PID     USER DATA      EXC/SHR OWN/WAIT
DOMINO1 013A 008EF190 16777220 0000000024780148 EXC      OWN
DOMINO2 02B2 008FA190 16908357 0000000024825220 EXC      WAIT
DOMINO3 0206 008FF458 16973924 0000000024824778 EXC      WAIT
RESOURCE    #2

NAME=SHARED CONDVAR  DATA: SHMID=00000648 OFFS/ADDR=0000000000002458  JOBNAME ASID TCB      PID
DOMINO2 02B6 008FA190 16908357 0000000024825220 EXC      WAIT
DOMINO3 0206 008FF458 16973924 0000000024824778 EXC      WAIT
RESOURCE #0002 IS LOCKED BY:
NAME=SHARED MUTEX  DATA: SHMID=00000648 OFFS/ADDR=0000000000002428

```

DISPLAY OMVS Command

The information displayed is:

NAME= SHARED MUTEX / CONDVAR

An indication of whether the object is a mutex (MUTEX) or condition variable (CONDVAR).

Example 15

To display information about all the dynamically activated service items, enter:

```
D OMVS,ACTIVATE=SERVICE
BPX0059I 08.51.42 DISPLAY OMVS 284
OMVS      000E ACTIVE          OMVS=(6D)
              DYNAMIC SERVICE ACTIVATION REPORT
SET #3:
LINKLIB=SYS1.DYNLIB.PVT           VOL=BPXLK1
LPALIB=SYS1.DYNLIB.LPA           VOL=BPXLK1
  OA12345  OA23456  OA34567  OA45678  ANLATC1
SET #2:
LINKLIB=SYS1.DYNLIB.PVT           VOL=BPXLK1
LPALIB=SYS1.DYNLIB.LPA           VOL=BPXLK1
  OA02001  OA02002  OA02003  OA02004  OA02004  OA02005
  OA02007  OA02008  OA02009
SET #1:
LINKLIB=SYS2.DYNLIB.PVT           VOL=BPXLK1
LPALIB=SYS1.DYNLIB.LPA           VOL=BPXLK1
  OA01001  OA01002  OA01003
ECSA STORAGE: 1268496          OMVS  STORAGE: 4768248
```

This display output shows that the service items (such as OA12345) are listed in groups based on when they were activated. The displayed information includes the library and volume from which each set of service was activated. At the end of the report, the output shows the amount of ECSA and OMVS address space storage consumed by all dynamically activated service items. Note that the amount of storage consumed will not decrease if you deactivate service items because the modules containing the deactivated service items remain in storage. For example, let us say you back off the most recently dynamically activated service (Set 3) shown in the output above with the following command:

```
F OMVS,DEACTIVATE=SERVICE
```

Next, you enter the display command again, to see the following output:

```
D OMVS,ACTIVATE SERVICE
BPX0059I 08.58.26 DISPLAY OMVS 296
OMVS      000E ACTIVE          OMVS=(6D)
              DYNAMIC SERVICE ACTIVATION REPORT
SET #2:
LINKLIB=SYS1.DYNLIB.PVT           VOL=BPXLK1
LPALIB=SYS1.DYNLIB.LPA           VOL=BPXLK1
  OA02001  OA02002  OA02003  OA02004  OA02005  OA02006
  OA02007  OA02008  OA02009
SET #1:
LINKLIB=SYS2.DYNLIB.PVT           VOL=BPXLK1
LPALIB=SYS1.DYNLIB.LPA           VOL=BPXLK1
  OA01001  OA01002  OA01003
ECSA STORAGE: 1268496          OMVS  STORAGE: 4768248
```

Note that the service items in Set #3 are no longer shown because they have been deactivated, but the total ECSA and OMVS storage consumed has not decreased.

Example 16

DISPLAY OMVS Command

To display information about waiters caused by mount latch contention, outstanding sysplex messages, file system contention or other conditions, enter:

D OMVS,W

```

SY1 D OMVS,W
SY1 BPX0063I 12.39.07 DISPLAY OMVS 426
OMVS    000E ACTIVE          OMVS=(QY)

MOUNT LATCH ACTIVITY:
USER   ASID   TCB           REASON               AGE
-----
HOLDER:
OMVS    000E  008E9828     Inact Cycle          00.01.18
      IS DOING: XPFSS VfsInactCall / XSYS Message To: SY2
      FILE SYSTEM: ZOS17.SY2.ETC.HFS
      HOLDING: File System Latch 123 EXCL

WAITER(S):
OMVS    000E  008D97C8     FileSys Quiesce      00.00.05
OMVS    000E  008E9B58     FileSys Sync          00.01.10

OUTSTANDING CROSS SYSTEM MESSAGES:
SENT SYSPLEX MESSAGES:
USER   ASID   TCB   FCODE MEMBER  REQID   MSG TYPE   AGE
-----
MEGA   0025 008DD218 0008 SY2      01000038 LookupCall 00.03.08
      FILE: somedirname          (12,456)
      HOLDING: File System Latch 333 SHR
TC0    0026 008E6E88 1011 SY1      0100003A Quiesce 00.00.05
      HOLDING: File System Latch 456 EXCL
OMVS   000E 008E9828 0804 SY2      01000039 VfsInactCall 00.01.18
      HOLDING: File System Latch 27 EXCL

RECEIVED SYSPLEX MESSAGES:
FROM   FROM           FROM
ON TCB ASID   TCB   FCODE MEMBER  REQID   MSG TYPE   AGE
-----
008D97C8 0026 008E6E88 1011 SY1      0100003A Quiesce 00.00.05
      IS DOING: Mount Latch Wait
008D1238 0022 008E6E00 0003 SY3      01000123 Read    00.07.25
      IS DOING: ZFS      Read
      FILE: thefilename          (44,1234)
      FILE SYSTEM: ZOS17.SY2.VAR.HFS
      HOLDING: File System Latch 33 SHR
FILE SYSTEM LATCH ACTIVITY:
USER   ASID   TCB           SHREXCL             AGE
-----
Latch 432 FILE SYSTEM: THE.FILESYS.NAME
HOLDER(S):
User10  0044  00880460      SHR                00:12:08
      IS DOING: NFS ReadCall
      FILE: somefilename          (88,1234)
User11  0045  00880460      SHR                00:15:58
      IS DOING: NFS ReadCall
      FILE: somefilename          (88,1234)
WAITER(S):
OMVS    000E  008E9B58      EXCL               00.01.10
Latch 678 FILE SYSTEM: ANOTHER.FILESYS.NAME
HOLDER(S):
OMVS    000E  00820420      EXCL               00:12:08
      IS DOING: ZFS SyncCall / Osi_Wait
WAITER(S):
User12  0022  008D97C8      SHR                00.00.05
User15  0072  008E9B58      SHR                00.01.10

OTHER WAITING THREADS:
USER   ASID   TCB           PID        AGE
-----
```

DISPLAY OMVS Command

```
USER01 0021 00908070 1234      00:12:41
  IS DOING: NFS Readdir / Running
  FILE: nfsdirname          (33,5432)
  FILE SYSTEM: HOST12.AJAX.DIRECTORY
  HOLDING: File System Latch #123 SHR
USER03 0041 00908070 786534    00:12:41
  IS DOING: BRLM Wait
  FILE: FileNameIsHere      (22,845)
  FILE SYSTEM: AJAX.DS88.ZFS
USER04 0051 00908070 15        00:00:49
  IS DOING: File Latch Wait-Latch 1379
  FILE: somefilename         (88,1234)
  FILE SYSTEM: HOST12.AJAX.DIRECTORY
  HOLDING: File System Latch #123 SHR
USER05 0071 00908070 378992    00:08:51
  IS DOING: ZFS Write / OSI_WAIT
  FILE: zfsfilename
  FILE SYSTEM: AJAX.DS23.ZFS
```

Example 17

To display information about the last 10 or less mount or move failures, enter:

```
D OMVS,MF
SY1 d omvs,mf
SY1 BPX0058I 11.22.20 DISPLAY OMVS 480
OMVS 000D ACTIVE OMVS=(MN,ZS)
SHORT LIST OF FAILURES:
TIME=16.24.40 DATE=2003/11/18      MOVE RC=0489 RSN=1278054D
  NAME=ZOS16.SY1.HFS
  PATH=/SY1
  SYSNAME=SY3
TIME=11.22.07 DATE=2003/11/18      MOUNT RC=0099 RSN=C5C7082A
  NAME=MY.HFS
  TYPE=HFS
  PATH=/SY1/tmp
TIME=21.58.17 DATE=2003/11/17      MOVE RC=0079 RSN=119E04B7
  NAME=*
  SYSNAME=SY9
TIME=11.54.04 DATE=2003/11/25      MOVE RC=0079 RSN=119E04B7
  PATH=/SY2
  SYSNAME=CAT
TIME=11.52.15 DATE=2003/11/25      MOVE RC=0079 RSN=119E04B7
  NAME=ZOS16.SY2.HFS
  SYSNAME=DOG
```

Example 18

To display information about each AF_UNIX socket, enter:

```
D OMVS,SOCKETS
BPX0060I 17.12.57 DISPLAY OMVS
OMVS 000D ACTIVE OMVS=(6F,JB)
AF_UNIX Domain Sockets
JOBNAME ID PEER ID STATE READ WRITTEN
-----
TCPICS 00000003 00000000 LISTEN 00000345
  Socket name: /var/sock/SYSTCPNCN.TCPICS
TCPICS 0000002A 00000022 ACP 000012AB 00054C2A
  Socket name: /var/sock/SYSTCPNCN.TCPICS
  Peer name: /tmp/sock1
NETVIEW 00000022 0000002A CONN 00054C2A 000012AB
  Socket name: /tmp/sock1
  Peer name: /var/sock/SYSTCPNCN.TCPICS
USER14 00000006 00000000 DGRAM 00000480 00000ABC
  Socket name: /tmp/test.sock
```

```

TCPICS 00000037 00000034 ACP      00000012 0000034C
       Socket name: /var/sock/SYSTCPN.TCPICS
JLB     00000034 00000037 CONN    0000034C 00000022
       Peer name: /var/sock/SYSTCPN.TCPICS

```

Displaying Operator Information (OPDATA)

Use the DISPLAY OPDATA command to display operator information (OPDATA). Depending on the operands specified, the display may represent either sysplex-wide data or system-unique data.

The syntax of the DISPLAY OPDATA command is:

```
D {OPDATA|O} [,PREFIX      ] [,L={a|name|name-a}]
      [,TRACKING|TR]
      [,MONITOR|MN] [,FULL]
```

OPDATA or O

The system displays operator information.

PREFIX

The system displays (message IEE603I) sysplex-wide information about the command prefixes defined for the subsystems in the sysplex. This is the default if no other operands are specified. See *z/OS MVS Planning: Operations* for more information.

TRACKING or TR

The status of the Console ID Tracking facility is displayed (message CNZ1001I), along with any recorded instances. See *z/OS MVS Planning: Operations* for more information.

MONITOR or MN

The system is to display through message CNZ1100I the enablement status of the monitoring facility for all message types supported, including whether each of these monitor message types are sent to the system log/operlog. The system also displays the number of consoles and, if applicable, TSO/E users that have requested to receive specific message types.

FULL

Instead of displaying the number of consoles and TSO/E users that have requested to receive specific message types, the system lists the names of those consoles. If there is any TSO/E user information to display, an additional section listing the user names will be included.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display information about the command prefixes current and active on the sysplex, enter:

```
D O,PREFIX
```

Example 2

DISPLAY OPDATA Command

To display information about the Console ID Tracking facility on this system, enter:
D 0,TRACKING

Example 3

To display information about the enablement status of the monitoring facility for all monitor message types supported, enter:

D 0,MONITOR

Displaying PARMLIB Information

Use the DISPLAY PARMLIB command to display:

- The parmlib data sets and volume serial numbers that are defined in LOADxx.
- The parmlib data sets and volume serial numbers that are defined in the MASTER JCL (when there are no LOADxx parmlib statements).

Note: If you did not specify SYS1.PARMLIB in the parmlib concatenation, the system automatically adds it to the end of the parmlib concatenation.

```
D PARMLIB [,ERRORS|E] [,L={a|name|name-a}]
```

ERRORS or E

Parmlib data sets and volume serial numbers that were defined in LOADxx PARMLIB statements but were not found.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

Assume a system has the following characteristics:

- Parmlibs STEVE.PARMLIB1 and STEVE.PARMLIB2 were used to IPL the system.
- Both parmlib data sets reside on volume D72665 and were specified on the PARMLIB statement in LOADxx.

D PARMLIB

The system returns the following display, slightly modified to improve readability:

```
SYS1      D PARMLIB
SYS1      IEE251I 16.41.05 PARMLIB DISPLAY 048
PARMLIB DATA SETS SPECIFIED
AT IPL
ENTRY    FLAGS     VOLUME   DATA SET
1          S         D72665  STEVE.PARMLIB1
2          S         D72665  STEVE.PARMLIB2
3          D         DEFVOL  PARMLIB
```

Example 2

Assume a system has the following characteristics:

- Parmlib data sets STEVE.PARMLIB1 and STEVE.PARMLIB2 were used to IPL the system.
- SYS1.PARMLIB was also specified on the PARMLIB statement of LOADxx.

D PARMLIB

The system returns the following display, slightly modified to improve readability:

```
SYS1      D PARMLIB
SYS1      IEE251I 16.41.05 PARMLIB DISPLAY 048
PARMLIB DATA SETS SPECIFIED
AT IPL
ENTRY    FLAGS     VOLUME   DATA SET
1        S          D72665   STEVE.PARMLIB1
2        S          DEFVOL   PARMLIB
3        S          D72665   STEVE.PARMLIB2
```

Example 3

Assume a system has the following characteristics:

- The default parmlib data set is used to IPL the system.
- The following output could mean one of the following:
 - There were parmlib data sets specified in LOADxx but they were not found.
Issue the D PARMLIB,ERRORS command.
 - No parmlib data sets were specified in either the LOADxx member or on the IEFPARM DD statement in Master JCL.

D PARMLIB

The system returns the following display, slightly modified to improve readability:

```
SYS1      D PARMLIB
SYS1      IEE251I 16.41.05 PARMLIB DISPLAY 048
PARMLIB DATA SETS SPECIFIED
AT IPL
ENTRY    FLAGS     VOLUME   DATA SET
1        D          DEFVOL   PARMLIB
```

Example 4

Assume a system has the following characteristics:

- There was no parmlib data set in LOADxx.
- The default parmlib data set is used to IPL the system.
- Parmlib data sets STEVE.PARMLIB3 and STEVE.PARMLIB4 were found in the IEFPARM DD statement of Master JCL.

D PARMLIB

The system returns the following display, slightly modified to improve readability:

```
SYS1      D PARMLIB
SYS1      IEE251I 16.41.05 PARMLIB DISPLAY 048
PARMLIB DATA SETS SPECIFIED
AT IPL
ENTRY    FLAGS     VOLUME   DATA SET
1        D          DEFVOL   PARMLIB

MASTER PROCESSING USING THE FOLLOWING PARMLIBS
ENTRY    FLAGS     VOLUME   DATE SET
1        S          D72666   STEVE.PARMLIB3
2        S          D72666   STEVE.PARMLIB4
```

Example 5

To display the parmlib data sets defined but not found, enter:

DISPLAY PARMLIB Command

```
D PARMLIB,ERRORS
```

Assume a system has the following characteristics:

- Parmlib data sets STEVE.PARMLIB5 and STEVE.PARMLIB6 were specified in LOAD xx PARMLIB statements, but they were not found.

```
D PARMLIB,ERRORS
```

The following illustration is slightly modified from what the user sees in order to improve readability in this documentation.

```
SYS1      D PARMLIB,ERRORS
SYS1      IEE251I 16.41.05 PARMLIB, ERRORS 048
PARMLIB DATA SETS SPECIFIED BUT NOT FOUND
ENTRY    FLAGS   VOLUME   DATA SET
 1        S        D72666  STEVE.PARMLIB5
 2        S        D72666  STEVE.PARMLIB6
```

Example 6

To display the parmlib data sets defined after a SETLOAD command update:

```
D PARMLIB
```

Assume a system has the following characteristics:

- At 11.05.14 on 9/13/96, a SETLOAD command was issued.
- The SETLOAD command used LOADPL which was found in data set SYS1.PARMLIB on volume CTDSD1.
- The LOADPL member has 3 parmlib statements:
 1. RELSON.MACLIB
 2. SYS1.PARMLIB
 3. RELSON.PARMLIB
- All 3 parmlib data sets in LOADPL reside on volume CTDSD1.
- The cataloged SYS1.PARMLIB data set is added to the end of the parmlib concatenation by default.

Note: The cataloged SYS1.PARMLIB data set, which is not the same data set as the data set SYS1.PARMLIB on volume CTDSD1, is automatically added to the end of the parmlib concatenation by default (because it was not explicitly stated in the parmlib concatenation).

```
D PARMLIB
```

The following illustration is slightly modified from what the user sees in order to improve readability in this documentation.

```
SYS1      D PARMLIB
SYS1      IEE251I 16.41.04 PARMLIB DISPLAY 048
PARMLIB DATA SETS SPECIFIED
AT 11.05.14 ON 09/13/1996
ENTRY    FLAGS   VOLUME   DATA SET
 1        S        CTDSD1  RELSON.MACLIB
 2        S        CTDSD1  PARMLIB
 3        S        CTDSD1  RELSON.PARMLIB
 4        D        CATALOG  PARMLIB
```

Displaying Commands Defined for PFKs

Use the DISPLAY PFK command to display the PFK definitions in effect for a specified console, the PFK definitions in a specified PFK table, or the PFK tables that are available.

```
D PFK[,CN=name] [,L={a|name|name-a}]
|,{TABLE|T} [=nnnnnnnn]
```

PFK

The system displays information about the PFKs (message IEE235I). Unless you specify otherwise on the CN= name operand, the PFK information refers to the console from which you issue the command.

Note: Only D PFK with the T or TABLE option is valid from extended consoles.

Any other specification of the D PFK command has no effect on extended MCS consoles or on system consoles, and is not valid for managing these consoles.

TABLE or T

Requests PFK definitions in a specific PFK table or lists all names of PFK tables that are available to be displayed.

nnnnnnnn

Requests PFK definitions in the PFK table named *nnnnnnnn*. If you omit =*nnnnnnnn*, the system displays the list of PFK tables available.

CN=*name*

Requests the PFK definitions for the console called *name*.

L=*a, name, or name-a*

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display will appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display PFK definitions for the console on which you issue the command, enter:

D PFK

Example 2

To display all available PFK tables, enter:

D PFK,T

Example 3

To display the contents (PFK definitions) of the PFK table named MVSCMDS, enter:

D PFK,T=MVSCMDS

Displaying Registered Products

Use the DISPLAY PROD command to display information about products that have been registered or display the product enablement policy. You can also use the command to determine the state (enabled, disabled, not defined, or not found) that, according to the current policy, exists for a specific product or set of products.

When the system searches for any products you specify, it allows wildcard matching. OWNER, NAME, FEATURENAME, and ID can include wildcard

DISPLAY PROD Command

characters (*) and (?) that allow a single parameter to match many different actual conditions. For example, OWNER(AD?) matches owner names like AD1 or AD2 but not ADD1. OWNER(A*) matches A1 or AD1 or ADD1.

The complete syntax for the DISPLAY PROD command is:

```
D PROD,{REGISTERED|REG|STATE|STATUS}  
[,OWNER(o)][,NAME(n)][,FEATURENAME(fn)][,ID(id)][,ALL][,L={a|name|name-a}]
```

Note: This command requires a /* */ around comments. Refer to “System Command Formats” on page 4-13 for further information.

PROD

Displays information about registered products or the product enablement policy.

REGISTEREDIREG

Displays information about any matching products that have registered as running on the system.

STATE

Displays information about the enablement state, defined in the enablement policy, for any matching products.

STATUS

For the product entry that is the best match for the product you specify, displays information about the enablement policy entry that the system would use if the product attempted to register.

If you specify STATUS, the system does not use wildcard matching; the wildcard characters (*) and (?) receive no special treatment.

OWNER(o)

Specifies the owner for the products to be displayed. You can specify wildcard characters (*) and (?). The default is OWNER(*), which matches all product owners unless you specified STATUS.

NAME(n)

Specifies the name of the products to be displayed. You can specify wildcard characters (*) and (?). The default is NAME(*), which matches all product names unless you specified STATUS.

FEATURENAME(fn)

Specifies the feature name of the products to be displayed. You can specify wildcard characters (*) and (?). The default is FEATURENAME(*), which matches all feature names unless you specified STATUS.

ID(i)

Specifies the identifier for the products to be displayed. You can specify wildcard characters (*) and (?). The default is ID(*), which matches all product identifiers unless you specified STATUS.

ALL

Specifies that all matching products, including those that registered with Ifaedreg_Type_NoReport, are to be displayed. Unless you specify ALL, products that registered with Ifaedreg_Type_NoReport are not displayed, even if they match the other criteria.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

Example 1

If you enter the following command:

```
D PROD,REG
```

the response has the following format:

IFA111I 11.15.28 PROD DISPLAY 358			
S OWNER	NAME	FEATURE	VERSION ID
E IBM CORP	z/OS	z/OS	01.08.00 5694-A01
E IBM CORP	z/OS	DFSMSTVS	**.**.** 5694-A01
E IBM CORP	z/OS	INFOPRINT SERVER	**.**.** 5694-A01
E IBM CORP	z/OS	JES2	01.08.00 5694-A01
E IBM CORP	z/OS	RACF	**.**.** 5694-A01
E IBM CORP	z/OS	RMF	**.**.** 5694-A01
E IBM CORP	z/OS	Security Server	**.**.** 5694-A01
E IBM CORP	z/OS	SDSF	**.**.** 5694-A01
E IBM CORP	z/OS	TCP/IP BASE	**.**.** 5694-A01

Displaying Entries in the List of APF-Authorized Libraries

You can use the DISPLAY PROG,APF command to display one or more entries in the list of APF-authorized libraries. (APF means authorized program facility.) Each entry in the APF list display contains:

- An entry number
- The name of an authorized library
- An identifier for the volume on which the authorized library resides (or *SMS*, if the library is SMS-managed).

You can issue the DISPLAY PROG,APF command from a console with INFO authority.

The complete syntax for the DISPLAY PROG,APF command is:

D PROG,APF[,ALL]	[,L={a name name-a}]
,DSNAME=libname ,ENTRY=xxx ,ENTRY=(xxx-yyy)	

Note: This command requires a /* */ around comments. Refer to “System Command Formats” on page 4-13 for further information.

PROG,APF

Displays libraries in the APF list. The parameters that follow this parameter determine the display information. If no parameters follow this parameter, the system displays all libraries in the APF list.

Both DISPLAY PROG,APF and DISPLAY PROG,APF,ALL display *all* libraries in the APF list.

ALL

Displays all libraries in the APF list. Both DISPLAY PROG,APF and DISPLAY PROG,APF,ALL display *all* libraries in the APF list.

DISPLAY PROG,APF Command

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

DSNAME=libname

Displays all entries for this library name. If an asterisk follows *libname*, the system displays all entries for all libraries beginning with *libname*. DSNAME can be an alias for the library name.

This function does not map an alias to the actual library name. Therefore, if you specify an alias, only the entry for the alias is displayed. Similarly, when you specify an actual library name as input, no entries for the library's aliases are displayed.

ENTRY=xxx

Displays the library entry for the specified decimal entry number. The order of the libraries in the APF list is not necessarily the order in which they were added. Use this parameter to limit the display to a specific library.

ENTRY=(xxx-yyy)

Displays all library entries in the range beginning with decimal entry number *xxx* and ending with decimal entry number *yyy*. The order of the libraries in the APF list is not necessarily the order in which they were added. Use this parameter to limit the display to a subset of the entire list of libraries.

Example

If you enter the command **D PROG,APF** the output appears in the following format:

```
CSV450I 13.25.02 PROG,APF DISPLAY
FORMAT=DYNAMIC
ENTRY VOLUME DSNAME
 1 580130 SYS1.LINKLIB
 2 580130 SYS1.SVCLIB
 3 617680 SYS1.ACCTG.DATA
 4 *SMS*  SYS1.MASAL.JOBS
```

Displaying Dynamic Exits

Use the DISPLAY PROG,EXIT command to display exits that have been defined to the dynamic exits facility or have had exit routines associated with them.

The complete syntax for the DISPLAY PROG,EXIT command is:

```
D PROG,EXIT,{ {EXITNAME|EX|EN}=exitname } [,DIAG]
{ {EXITNAME|EX|EN}=exitname* }
{ {MODNAME|MOD}=modname }
{ [ALL] [,IMPLICIT|,IMP] }
[,L={a|name|name-a}]
```

Note: This command requires a /* */ around comments. Refer to "System Command Formats" on page 4-13 for further information.

PROG,EXIT

Displays the names of exits that have been defined to the dynamic exits facility, had exit routines associated with them, or had their attributes changed.

ALL

Displays the names of all the exits that have been defined to the dynamic exits facility, have had exit routines associated with them, or have had their attributes changed.

EXITNAME= or EX= or EN=exitname

Displays the names of all exit routines associated with the named exit, along with status information about the exit. The exit routines are displayed in the order in which they are invoked by dynamic exits services.

If no exit routines are associated with a particular exit, the system issues message CSV463I.

EXITNAME= or EX= or EN=exitname*

Displays the names of exits that both:

- Have a name that matches *exitname*. The trailing asterisk “*” is a wildcard that is used to match patterns.
- Are defined or have had an exit routine associated with them.

DIAG

An optional keyword that specifies diagnostic information for the exit specified by EXITNAME=*exitname*. The CSV464I. The message displays information about the state of the exit, the entry point address of the exit routine, the load point address of the exit routine module, the length of the exit routine module, and jobname. For the sample output, see page 4-200.

MODNAME= or MOD=name

Displays the names of the exits with which the specified exit routine is associated. You can use this information before replacing an exit routine to ensure that the exit routine is not defined to any exits.

IMPLICIT or IMP

Displays the names of exits that have been implicitly defined. An exit is implicitly defined when:

- You add exit routines to an exit before the exit is defined
- You set attributes using the ATTRIB parameter of the SETPROG EXIT command before defining the exit.

You can use this parameter to determine whether exit routines were improperly added to an exit that might never be defined. Issue SETPROG EXIT,UNDEFINE,EXITNAME=*exitname* to have the system remove the improper definition of that exit.

Both IMPLICIT and ALL,IMPLICIT display the names of all the exits that have been implicitly defined.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

Example 1

To display exits that have an exit name starting with “IEF”, and either are defined or have had an exit routine associated with them, enter:

```
D PROG,EXIT,EXITNAME=IEF*
```

The output appears in the following format:

DISPLAY PROG,EXIT Command

```
CSV460I 17.01.16 PROG,EXIT DISPLAY 710
EXIT          DEF EXIT          DEF EXIT          DEF
IEF_ALLC_OFNL  E   IEF_SPEC_WAIT    E   IEF_VOLUME_ENQ    E
IEF_VOLUME_MNT E   IEFDB401      E
```

Example 2

To display all exit routines associated with exit SYS.IEFU84 along with status information about the SYS.IEFU84, enter:

```
D PROG,EXIT,EXITNAME=SYS.IEFU84
```

The output appears in the following format:

```
CSV461I 17.04.20 PROG,EXIT DISPLAY 725
EXIT          MODULE  STATE MODULE  STATE MODULE  STATE
SYS.IEFU84     IEFU84    A   MYIEFU84    I
```

Example 3

To display exit names that are associated with exit routine IEFU84, enter:

```
D PROG,EXIT,MODNAME=IEFU84
```

The output appears in the following format:

```
CSV462I 17.05.33 PROG,EXIT DISPLAY 731
MODULE  IEFU84
EXIT(S) SYS.IEFU84      SYSSTC.IEFU84
```

Example 4

To display information about the exit entry point address, the load point address of the exit routine module, and other diagnostic information for exit routine SYS.IEFU84, enter:

```
D PROG,EXIT,EXITNAME=SYS.IEFU84,DIAG
```

The output appears in the following format:

```
CSV464I 11.45.00 PROG,EXIT DISPLAY
EXIT SYS.IEFU84
MODULE  STATE EPADDR LOADPT LENGTH  JOBNAME
IEFU84  A     12345678 00000000 00000000 *
```

Displaying LNKLST Information

Use the DISPLAY PROG,LNKLST command to display information about the LNKLST set. The command provides information about LNKLST sets for the LNKLST concatenation and associated jobs.

The complete syntax for the DISPLAY PROG,LNKLST command is:

```
D PROG,LNKLST[,NAME=[lnklstname|CURRENT]
[,NAMES]
[,USERS,[CURRENT|NOTCURRENT|NAME=lnklstname]]
[,ASID=asid]
[,JOBNAME=jobname]
[,L={a|name|name-a}]
```

Note: This command requires a /* */ around comments. Refer to "System Command Formats" on page 4-13 for further information.

PROG,LNKLST

Displays information about the LNKLST concatenation and jobs associated with it. When the LNKLST is authorized by default, the APF authorization status provided is only applicable when the data set is referenced independently of the LNKLST.

NAME=CURRENT**NAME=*lnklstname***

Displays the data sets for the specified LNKLST set or concatenation.

If you specify CURRENT, the system displays information for the current LNKLST set that has been activated as the LNKLST concatenation.

For *lnklstname*, you must specify a valid 1 to 16 character name of a LNKLST set defined to the system.

Default: NAME=CURRENT is the default. If you omit this parameter, the system displays information for the current LNKLST concatenation.

NAMES

Displays the name of each LNKLST set defined to the system.

USERS,CURRENT

Displays a list of address spaces that use the current LNKLST set.

Default: CURRENT is the default. If you omit this parameter, the system displays a list of address spaces for the current LNKLST set.

USERS,NOTCURRENT

Displays a list of address spaces that use any LNKLST set besides the current LNKLST set.

USERS,NAME=*lnklstname*

Displays a list of address spaces that use the LNKLST set specified by NAME=*lnklstname*.

For *lnklstname*, you must specify a valid 1 to 16 character name defined of a LNKLST set defined to the system.

ASID=*asid*

Displays the LNKLST set in use by the address space for the specified ASID.

JOBNAME=*jobname*

Displays the LNKLST set in use by the specified job. The system provides information for any job that matches *jobname*. *jobname* can include wildcard characters (*) or (?).

L=*a, name, or name-a*

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

Example 1

To display information for the LNKLST concatenation (defined as LNKLST1 in PROGxx and activated at IPL), enter:

```
D PROG,LNKLST
```

The output appears in the following format. For a description of the output fields, use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books to see message CSV470I.

DISPLAY PROG,LNKLST Command

```
CSV470I 01.00.00
LNKLST DISPLAY
LNKLST SET LNLKST1 LNKAUTH
=APFTAB
ENTRY APF VOLUME DSNAME
1 A DRV602 SYS1.LINKLIB
2 A DRV602 SYS1.MIGLIB
3 A DRV602 SYS1.CSSLIB
4 SMS MY.LINKLIB
```

Example 2

To display the LNLKST set associated with the job that matches the jobname MYJOB, enter:

```
D PROG,LNLKST,JOBNAME=MYJOB
```

The output appears in the following format:

```
CSV473I 02.15.00 LNLKST DISPLAY
LNKLST SET ASID JOBNAME
MYLNKLST 0018 MYJOB
```

Displaying LPA Information

Use the DISPLAY PROG,LPA command to display the entry point, load point, and size of modules in the LPA, and to display the minimum amount of CSA and ECSA that must remain after dynamically adding a module to the LPA.

The complete syntax for the DISPLAY PROG,LPA command is:

```
D PROG,LPA{,MODNAME=modname}{,L={a|name|name-a}}
{,CSAMIN}
```

Note: This command requires a /* */ around comments. Refer to “System Command Formats” on page 4-13 for further information.

MODNAME=

Displays entry point, load point, and length information about the LPA module. You can use MOD and MODULE as synonyms of MODNAME.

modname

is the 1-8 character LPA module name. If the last character of the modname is an asterisk (*), it will be treated as X'CO'.

CSAMIN

Displays the current CSA and ECSA minimum values.

L=a, name, or name-a

Specifies the display area (a), console name (name), or both (name-a) where the display is to appear.

Displaying System Requests

Use the following form of the DISPLAY command to display outstanding messages requiring operator action. These messages include WTOR messages, action messages saved by AMRF, action messages issued by the communications task, and action messages that were not displayed on all necessary consoles. You can request that the system display:

DISPLAY R Command

- The immediate action messages (descriptor codes 1 or 2), eventual action messages (descriptor code 3), and critical eventual action messages (descriptor code 11)
 - The device numbers of devices waiting for mount requests to be fulfilled
 - The device numbers of devices waiting for operator intervention
 - The status of the action message retention facility
 - An alphabetical list of keynames of outstanding action messages
 - The messages issued by a specified system
 - The messages that await operator response at a specified console
 - The messages that have specific routing codes

The following list summarizes how you use the operands on the DISPLAY command to retrieve and display information:

- The U operand displays information about devices and units.
 - The I, E, CE, R, and M operands display outstanding action messages.
 - The LIST, L, ALL, and A operands display combinations of the above.
 - The KEY operand displays an alphabetical list of keynames of outstanding action messages.

The resulting display is described in messages IEE112I (successful result) or IEE312I (unsuccessful result). Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books to see explanations of those messages.

```

D R[,U]
      ,KEY[,SYS=sysname][,CN=(ALL)]
      [,I      ][,msgformat][,MSG=msgid][,SYS=sysname][,KEY=keyname]
      [,        ][,JOB=jobname]
      ,E      [,CN={name|(ALL)}][,ROUT={ALL|(rrr[,sss]...)}
      ,CE                                {(rrr-sss[,rrr-sss]...)}
      ,R
      ,M
      ,{LIST|L}
      ,{ALL|A}
      ,
      [,L={a|name|name-a}]

      (See Note)

```

Note: If you supply all commas between DISPLAY R and the operands that have equal signs, you get default values. However, supply only one comma before the L operand, even if you omit the preceding operands. For example, DISPLAY R,I,L=2B.

Because TSO consoles in OPERATOR mode do not route messages by console IDs or routing codes, do not use the ROUT= or CN= operands on DISPLAY R commands issued from these consoles.

R

The system is to display information about outstanding action messages (that is, messages with descriptor codes 1, 2, 3, or 11), WTORs, and devices awaiting mount requests to be fulfilled, and units requiring intervention. Information includes either the text of all outstanding action messages and

DISPLAY R Command

WTORs, a summary of keynames associated with the outstanding action messages, or device numbers. The system also displays a number that represents the total of all outstanding WTORs or action messages.

If the issuing console has master authority, the system displays, on the issuing console, all outstanding WTORs. Otherwise, unless you specify the CN parameter on the command, the system displays information about only those messages that appeared on the console that issues the DISPLAY R command.

Consoles of some subsystems, such as NetView, must specify the CN=(ALL) parameter to ensure displaying all outstanding requests.

Optional subparameters are:

- I** Display the texts and message identification numbers of all outstanding immediate action messages (descriptor codes 1 or 2).
- E** Display the texts and message identification numbers of all outstanding eventual action messages (descriptor code 3).

CE

Display the texts and identification numbers of all outstanding critical eventual action messages (descriptor code 11).

- R** Display the texts and message identification numbers of all messages awaiting replies (WTORs).
- M** Display the texts and message identification numbers of all immediate action, eventual action, and critical eventual action messages, and messages awaiting replies.

LIST or L or ALL or A or blank

Display the texts and message identification numbers of all immediate action, eventual action, and critical eventual action messages and messages awaiting replies. Also display the device numbers of devices with unfulfilled mount requests and any units requiring operator intervention.

- U** Display the device numbers of devices with unfulfilled mount requests and any units requiring operator intervention.

msgformat

Specifies the information that is to accompany messages when they are displayed on a console. The possible values of *msgformat* are:

- J** Display the message text with the jobname or job ID of the message issuer. If JES3 is the primary subsystem and is running in XCF-local mode, this option displays the jobname, but not the job ID.

JN

Display the message text with only the job name of the message issuer. If JES3 is the primary subsystem and is running in XCF-local mode, JN has the same effect as J.

- M** Display only the text of each message.
- S** Display the message text, the name of the system that sent the message, and the jobname or job ID of the message issuer. If JES3 is the primary subsystem and is running in XCF-local mode, this option displays the jobname, but not the job ID.

SN

Display the system name and the jobname of the message issuer. If JES3 is the primary subsystem and is running in XCF-local mode, SN has the same effect as S.

T Display the message text with the time stamp, the name of the system that sent the message, and the jobname or job ID of the message issuer. If JES3 is the primary subsystem and is running in XCF-local mode, this option displays the jobname, but not the job ID.

TN

Display the message text with the time stamp, the name of the system that sent the message, and the jobname of the message issuer. If JES3 is the primary subsystem and is running in XCF-local mode, TN has the same effect as T.

The format of a message that includes all message format options is:

Time stamp System name Jobname/id Message text

Default: For MCS, SMCS and extended MCS consoles, the default message format differs depending on the primary subsystem. If it is JES2, the default format is defined by the MFORM setting for the console. (You can use the CONTROL S command (K S,MFORM) to change the MFORM setting.) If it is JES3, the default message format option is **S**. You can use the CONTROL command to change the default for MCS and SMCS consoles.

M is the default message format option for extended MCS consoles. To change the default value for extended MCS consoles, use the RACF command, ALTUSER userid OPERPARM(MFORM(T,S,J,M,X)). See *z/OS Security Server RACF Command Language Reference* for more information.

MSG=msgid

The text of any action message awaiting a reply is to be displayed if the message identifier begins with the one to ten characters specified by *msgid*. Specify a trailing asterisk (*) wildcard to request messages for all message identifiers that match a leading character string.

JOB=jobname

The system requests the messages that are identified by a one to eight-character jobname. Specify a trailing asterisk (*) wildcard to request messages for all jobnames that match a leading character string. For example, enter the following command to display outstanding messages requiring operator action for all jobnames that begin with the characters TSO1:

D R,JOB=TSO1*

SYS=sysname

The system is to display messages that have appeared at the system named *sysname* or, if you also specify KEY, the keynames of messages issued at the system with this name.

If you issue DISPLAY R,M,SYS=*sysname* from a console on the JES3 global, you get all messages for the system named *sysname*. (Note that the system retrieves the same information if you issue the DISPLAY R,L,SYS=*sysname* command. You cannot retrieve unit information from another system.)

If you issue DISPLAY R,M without the SYS=*sysname* operand, at an MCS or SMCS console that also controls the JES3 global, the system displays all outstanding messages for that system and for the local systems attached to

DISPLAY R Command

it. If you issue the same command at a MCS or SMCS console that also controls a JES3 local or JES2 system, the system displays the messages only for that system.

KEY

The system displays an alphabetical list of keynames associated with outstanding messages. The system also displays the total number of messages for each keyname.

KEY=*keyname*

The system requests those messages that are identified by a one to eight-character keyname, such as those messages issued by the specified dynamic support program (DSP) of JES3.

KEY=MOUNT

The system displays outstanding tape mount requests.

CN

The system displays a set of messages and device numbers of devices awaiting mount requests to be fulfilled, and units requiring intervention, or, if you also specify KEY, a list of outstanding keynames of messages that appear at a specified console or all consoles. These messages include those directed by routing code and those directed by console id.

If you omit the CN operand, the default is the current console on which you enter the D R command.

name

Requests those outstanding action messages that the system directed to the console with the name *name*.

(ALL)

Requests the outstanding action messages that the system directed to all consoles. The parentheses are required.

ROUT

The system displays only the outstanding action messages that have the specified routing codes. The system rejects the ROUT operand if you also request a summary of keynames.

ALL

Requests messages with any routing code.

(rrr[sss]...)

Requests messages with one or more routing codes.

(rrr-sss)[,(rrr-sss)]...

Requests messages within a range of routing codes. When you specify a range of routing codes, the first *rrr* in the range must be less than or equal to the second *sss*.

NONE

Requests only those messages that the system directs to the console by console id.

L=a, *name*, or *name-a*

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display the identification numbers and texts of all unanswered system requests, the device numbers of all devices waiting for mount requests to be fulfilled, and the device numbers of all devices waiting for operator intervention in area A of the console named CON2, enter:

D R,L,L=CON2-A or D R,A,L=CON2-A

Example 2

To display the identifications and texts of all unanswered requests for operator action or reply that begin with identifier IEF in the first available area of the console through which the command is entered, enter:

D R,MSG=IEF

Example 3

To display the total number (and not the texts) of outstanding action messages, WTORs, devices awaiting mount requests to be fulfilled, and units requiring intervention, enter:

D R

Example 4

To display all outstanding action and WTOR messages that have routing codes 1-12, enter:

DISPLAY R,M,ROUT=(1-12)

If the console is defined to receive, for example, only routing codes 1 and 2, the display includes only messages with those routing codes.

Example 5

To display a summary of all keynames currently active and the number of outstanding messages associated with each keyname, enter:

DISPLAY R,KEY

Example 6

To display the text of the outstanding message associated with the keyname TAPE listed in response to the command in Example 5, enter:

DISPLAY R,KEY=TAPE

Example 7

To display the system names, job id's and message texts of all messages issued at any system within a JES3 complex, enter from a console with master authority:

DISPLAY R,A,S

Example 8

To display all outstanding messages issued on system SY2:

DISPLAY R,M,SYS=SY2

DISPLAY R Command

The system also displays numbers of devices that await mount requests and units requiring intervention.

Example 9

To display all outstanding messages directed specifically to the console named CON02, excluding messages that were directed to the console by default, enter:

```
DISPLAY R,L,CN=CON02
```

The system also displays numbers of devices that await mount requests and units requiring intervention.

Example 10

To display all outstanding messages directed specifically to the console named CON04 by any routing code, enter:

```
DISPLAY R,ROUT=ALL    issued from console CON04
```

or

```
DISPLAY R,ROUT=ALL,CN=CON04    issued from another console
```

Example 11

To display all outstanding messages with their jobnames, enter:

```
DISPLAY R,L,JN
```

Example 12

To display all outstanding messages whose jobnames begin with the characters CICS, enter:

```
DISPLAY R,L,JOB=CICS*
```

or

```
DISPLAY R,L,JN,JOB=CICS*
```

Displaying Resource Recovery Services (RRS) Information

Use the DISPLAY RRS command to display status information about RRS coordinated transactions to the system console and SYSLOG. You can also use automation to parse the output and trigger alerts.

The command supports the following parameters:

- **UR** - Display unit of recovery information, in either summary or detailed format. You can filter the output information with the optional keyword filter parameters described below.
- **RM** - Display resource manager information, in either summary or detailed format. You can filter the output information with the optional keyword filter parameters described below.

```

D RRS[,UR[ ,SUMMARY|,SUM|,S|,DETAILED|,D]uroptions]
    [,RM[ ,SUMMARY|,SUM|,S|,DETAILED|,D]rmoptions]
    [,L=a|name|name-a]

uroptions:
[,URID=ur-identifier]
[,STATE=FLT|SCK|PRP|DBT|CMT|BAK|END|OLA|CMP|FGT]
[,SYSNAME=system-name]
[,GNAME=logging-group-name]

rmoptions:
[,RMNAME=resource-manager-name]
[,SYSNAME=system-name]
[,GNAME=logging-group-name]

```

RRS

The system is to display status information about RRS coordinated transactions to the system console and SYSLOG.

UR

Indicates that the system is to display information about RRS coordinated transactions.

SUMMARY or SUM or S

Indicates that the system is to use the summary form of the output (ATR601I). The resulting message contains a list of RRS coordinated transactions that were selected through the optional filter parameters.

A summary UR entry includes the following information:

- UR identifier
- System name
- RRS logging group name
- UR state
- UR type - protected or unprotected
- Comment - comments about this UR

DETAILED or D

Indicates that the system is to use the DETAILED form of the output (ATR603I) . The resulting message contains the detailed information for the particular transaction as indicated by the **URID=** parameter.

With this output form, The **URID=** parameter is required and only one particular UR can be selected (no wildcards are allowed). SYSNAME and GNAME can not use wildcards.

A detailed UR entry includes all the information in a summary report, plus the following information:

- A list of expression of interests that are associated with the transaction. For each expression of interest: URI token, resource manager name, resource manager role, interest type, interest status, interest state, exit status, and duration.

RM

Indicates that the system is to display information about resource managers that are currently active or were previously active with RRS.

SUMMARY or SUM or S

Indicates that the system is to use the summary form of the output (ATR602I) . The resulting message contains a list of resource managers

DISPLAY RRS Command

that were selected through optional keyword filter parameters. A summary RM entry includes the following information:

- Resource manager name
- Resource manager state
- System name
- RRS logging group name

DETAILED or D

Indicates that the system is to use the **DETAILED** form of the output (ATR604I). The resulting message contains the detailed information for the then specified resource manager as indicated by the **RMNAME=** parameter.

With this output form, the **RMNAME=** parameter is required and only one particular RM can be selected (no wildcards are allowed).

SYSNAME and GNAME can not use wildcards.

A detailed RM entry includes all the information in a summary report, plus the following information:

- RM token
- A list of URs associated with the resource manager. The information displayed for each UR is similar to the UR summary information.

URID=ur-identifier

A UR identifier used to limit the number of URs returned for a UR request. For a SUMMARY request, this parameter is optional. If specified, wildcards are allowed (* or ?). If not specified, the UR identifier is not used to filter the returned UR information. For a DETAILED request, this parameter is required and can not contain any wildcards.

STATE=FLTISCKIPRPIDBTICMTIBAKIENDIOLAICMPIFGT]

The UR state of the UR(s) to be returned. If not specified, URs in any state are returned.

RMNAME=resource-manager-name

A resource manager name used to limit the number of RMs returned for a RM request. For a SUMMARY request, this parameter is optional. If specified, wildcards are allowed (* or ?). If not specified, the resource manager name is not used to filter the returned RM information. For a DETAILED request, this parameter is required and can not contain any wildcards.

GNAME=logging-group-gname

An RRS logging group name used to limit the amount of returned information. Wildcards (* or ?) are allowed for only the summary form of output. The default is the logging group for the current system.

SYSNAME=system-name

A system name used to limit the amount of returned information. Wildcards (* or ?) are allowed for only the summary form of output. The default is the current system.

L=a,name, or name-a

Indicates the display area (**a**), console name (**name**), or both the console name and the display area (**name-a**) where the display will be presented. If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Displaying RTLS Information

Use the DISPLAY RTLS command to display the current status of the run-time library services (RTLS) environment. The command provides information about the physical and logical libraries in use, the users of the logical libraries, and the cache use for a given library or for all libraries.

When the system searches for information you specify, it allows wildcard matching. LIBRARY, VERSION, MODULE, and JOBNAMES can include wildcard characters (*) and (?) that allow a single parameter to match many different conditions. For example, LIBRARY=/* allows you to request information about all defined libraries. To request information about all modules, specify MODULE=*. To request information about all modules with three-character names beginning with M and ending with D (such as MAD, MBD, and MCD), specify MODULE=M?D.

The syntax for the DISPLAY RTLS command is:

```
D RTLS[,NAMES[,LIBRARY=lname[,VERSION=ver] [,CURRENT|,SEQNUM=num|,ALL]
      [,PHYSICAL,LIBRARY=pname[,CURRENT|,SEQNUM=num|,ALL] [,MODULE=mod|,LOGICAL]
      [,LOGICAL{,LIBRARY=lname[,VERSION=ver] [,CURRENT|,SEQNUM=num|,ALL] }
           { [,MODULE=mod|,USERS]
           {,JOBNAME=jobname
           {,ASID=asid
           }
           }
           [,L={a|name|name-a}]]
```

Note: This command requires a /* */ around comments. Refer to “System Command Formats” on page 4-13 for further information.

RTLS

Displays information about the current RTLS environment.

NAMES

Displays the RTLS common storage use and definitions, as well as the names of the physical and logical libraries.

LIBRARY=*Iname*

Displays the common storage use and definitions, as well as the names of the physical and logical libraries that match the specified name. For *Iname*, you must specify a valid 1 to 8 character library name. You can use wildcard characters when specifying the library name.

VERSION=*ver*

Displays information only for versions that match the specified version identifier. For *ver*, you must specify a valid 1 to 8 character version name. You can use wildcard characters when specifying the version.

CURRENT

Displays information only for the current level.

SEQNUM=*num*

Displays information only for the level that matches the specified sequence number.

ALL

Displays information for all levels.

DISPLAY RTLS Command

PHYSICAL,LIBRARY=pname

Displays, for the specified library, the common storage use and definitions, as well as the names of the data sets that make up the physical library. For *pname*, you must specify a valid 1 to 8 character physical library name. You can use wildcard characters when specifying the library name.

CURRENT

Displays information only for the current level.

SEQNUM=num

Displays information only for the level that matches the specified sequence number.

ALL

Displays information for all levels.

MODULE=mname

Displays information about each module in the physical library with a name that matches the specified name. For *mname*, you must specify a valid 1 to 8 character load module name. You can use wildcard characters when specifying the module name.

LOGICAL

Displays the logical libraries of which this physical library is a part.

LOGICAL

Displays, for each matching library, statistics about the use of the library and the physical libraries that are part of the logical library.

LIBRARY=lname

Identifies a logical library. For *lname*, you must specify a valid 1 to 8 character name of a logical library. You can use wildcard characters when specifying the library name.

VERSION=ver

Displays information only for versions that match the specified version identifier. For *ver*, you must specify a valid 1 to 8 character version name. You can use wildcard characters when specifying the version.

CURRENT

Displays information only for the current level.

SEQNUM=num

Displays information only for the level that matches the specified sequence number.

ALL

Displays information for all levels.

MODULE=mname

Displays information about each module in the logical library with a name that matches the specified name. For *mname*, you must specify a valid 1 to 8 character name of a load module. You can use wildcard characters when specifying the module name.

USERS

Displays users connected to the logical library.

JOBNAME=jobname

Displays the logical libraries to which the specified *jobname* is connected. You can use wildcard characters when specifying *jobname*.

ASID=asid

Displays the logical libraries to which the specified address space is connected.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

Example 1

To display the RTLS common storage use and definitions, as well as the names of all the physical and logical libraries, enter:

```
D RTLS,NAMES
```

Example 2

To display the RTLS common storage use and definitions, as well as the names of all the physical and logical libraries that start with the letters MONTH and are at sequence number 4 of version D1, enter:

```
D RTLS,NAMES,LIBRARY=MONTH*,VERSION=D1,SEQNUM=4
```

Example 3

To display, for physical library MYMODS, the common storage use and definitions, as well as the names of the data sets that make up the current level of the physical library, enter:

```
D RTLS,PHYSICAL,LIBRARY=Mymods
```

Example 4

To display, for physical library MYMODS, information about each module that is 6 characters long and has NEWPG as the first 5 characters, enter:

```
D RTLS,PHYSICAL,LIBRARY=Mymods,MODULE=NEWPG?
```

Example 5

To display, for physical library MYMODS, the logical libraries of which physical library MYMODS is a part, enter:

```
D RTLS,PHYSICAL,LIBRARY=Mymods,LOGICAL
```

Example 6

To display, for logical library MYMODS, statistics about the use of the library and the physical libraries of which the logical library is a part, enter:

```
D RTLS,LOGICAL,LIBRARY=Mymods
```

Example 7

To display information about module SALES in logical library MONTHJAN, enter:

```
D RTLS,LOGICAL,LIBRARY=MONTHJAN,MODULE=SALES
```

Example 8

To display the users connected to logical library MONTHJAN, enter:

```
D RTLS,LOGICAL,LIBRARY=MONTHJAN,USERS
```

DISPLAY RTLS Command

Example 9

To display the logical libraries to which job name REPORTS is connected, enter:
D RTLS,LOGICAL,JOBNAME=REPORTS

Example 10

To display the logical libraries to which the ASID 1234 is connected, enter:

D RTLS,LOGICAL,ASID=1234

Displaying SLIP Trap Information

Use the DISPLAY SLIP command to display information about SLIP traps.

```
D SLIP[=xxxx] [,L={a|name|name-a}]
```

SLIP

Indicates that the system is to display summary information about SLIP traps or detailed information about one SLIP trap (message IEE735I).

xxxx

The system is to display detailed information about the SLIP trap associated with the identifier xxxx. If you do not specify xxxx, the system lists all the SLIP traps in the system and tells whether each trap is enabled or disabled.

Where asterisks replace any or all of the four characters of xxxx, the system displays all SLIP traps whose identifiers match the non-asterisk characters in xxxx. If you specify fewer than four characters, the xxxx is padded on the right with blanks. A matching identifier must have blanks in those positions.

The asterisks allow you to group your SLIP traps by common characters and display them as a group.

L=a, name, or name-a

Specifies the display area (a), console name (name), or both (name-a) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

To display all SLIP traps and whether they are enabled or disabled, enter:
DISPLAY SLIP

Example 2

To display detailed information about trap W292, enter:
D SLIP=W292

Example 3

To display all SLIP traps with an identifier having 'A' as the first character and 'B' as the third character and identify whether they are disabled or enabled, enter:

```
DISPLAY SLIP=A*B*
```

Displaying SMF Data

Use the DISPLAY SMF command to display System Management Facilities (SMF) data.

```
D SMF[,S|,0][,L={a|name|name-a}]
```

SMF

Indicates that the status of SMF data sets or the SMF options in effect are to be displayed (message IEE967I).

- S** Directs the system to display the names and status of the SMF data sets.
- O** Directs the system to display the current SMF options.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Displaying Storage Management Subsystem Information

Use the DISPLAY SMS command to display the following kinds of information about the Storage Management Subsystem (SMS):

- Active SMS configuration
- 3990-3 or 3990-6 cache control unit statistics (if the 3990-3 or 3990-6 is installed and has at least one system-managed volume attached)
- Status of system-managed volumes, storage groups, drives, or libraries
- SMS trace options in effect
- SMSVSAM status of sharing control data sets, the SMSVSAM server, data set statistical monitoring, coupling facility cache and lock structures and cache structures that contain data for a specified volume
- CICSVR status of Recovery Control Data sets (RCDSS) and the status of the CICSVR address space.

Storage groups and volumes can be in one of the following states:

- *Not defined* means that the storage group or volume (specified on this command) is specified in the SMS configuration as not being connected to this MVS system. If the storage group or volume is not defined in the active configuration, a message indicates that the system rejects this command for that reason. If the command is issued for a storage group with no volume, another message indicates the lack of volumes in that group. The symbol for this state is ‘.’
- *Enabled* means that SMS permits allocation of data sets in this storage group or volume to this MVS system. The symbol for this state is ‘+’
- *Disabled* means that SMS does not allow allocation of data sets in this storage group or volume for this MVS system. The symbol for this state is ‘-’
- *Disabled (new only)* means that SMS does not allow allocation of new data sets in this storage group or volume for this MVS system. The symbol for this state is ‘D’

DISPLAY SMS Command

- *Quiesce* means that for a data set, SMS selects the specified volume or storage group only if it finds no other choices. The symbol for this state is “*”
- *Quiesce(new)* means that for a new data set, SMS selects the specified volume or storage group only if it has no other choices. The symbol for this state is ‘Q’

For a detailed discussion of the DISPLAY SMS command pertaining to optical and tape volumes, libraries, and drives, and the OAM address space, see *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*, and the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

For a detailed discussion of the DISPLAY SMS command pertaining to CICSVR, see *CICSVR V3R1 Implementation Guide*, *CICSVR V3R1 User’s Guide and Reference*, and *CICSVR V3R1 Messages and Problem Determination*.

```

D SMS[,{ACTIVE|A}]                                ]
      [,CACHE]                                     ]
      [,CFCACHE({structurename|*})]                ]
      [,CFLS]                                      ]
      [,CFVOL(volid)]                            ]
      [,CICSVR[,{ALL|LOGSTREAMS({LogstreamName|ALL})|RCDS} ]]
      [,,{DRIVE|DRI}({name|ALL})[ ,STATUS    ]
          |,DETAIL]                               ]
      [,DSNAME(dsn){,WTOR}]                      ]
      [,JOB(jobname){,WTOR}]                     ]
      [,,{LIBRARY|LIB}({name|ALL})[ ,STATUS[,LISTDRI]]
          |,LISTDRI
          |,DETAIL]                             ]
      [,LOG({logstreamid|ALL}{,WTOR}            ]
      [,MONDS({specmask|*})]                      ]
      [,OAM]                                       ]
      [,OPTIONS]                                    ]
      [,OSMC[,TASK(name)]]                        ]
      [,,{PDSE|PDSE1}{,LATCH(1addr)[ ,DETAILED ]
          |,SUMMARY
          {,MODULE(modname)}}]                   ]
      [,SEP]                                       ]
      [,SHCDS]                                     ]
      [,SHUNTED,{SPHERE(sphere)|UR({urid|ALL})}{,WTOR} ]
      [,SMSVSAM[,ALL]]                           ]
      [,SMSVSAM,QUIESCE]                         ]
      [,,{STORGRP|SG}{(storgrp|ALL)}[ ,LISTVOL ]
          |,DETAIL]                               ]
      [,,{TRACE|T}]                                ]
      [,TRANVSAM[,ALL][,ALLLOGS][,WTOR]        ]
      [,URID({urid|ALL}){,WTOR}                 ]
      [,,{VOLUME|VOL}(volume)]                  ]
      [,VOLSELMSG]                                ]
      [,L={a|name|name-a}]                       ]

```

DISPLAY SMS Command

SMS

Displays information about the Storage Management Subsystem. If SMS is the only operand specified, this command displays the active SMS configuration.

ACTIVE or A

The display includes the names of the three main SMS system data sets currently in use: the active control data set (ACDS), the communications data set (COMMDS), and the source control data set (SCDS). The display also includes the interval (DINTERVAL), in seconds, that SMS waits between reading device statistics for the 3990-3 control unit. In addition, the display shows the values of the REVERIFY and ACSDEFAULTS parameters in the IGDSMSxx parmlib member.

The display also includes a list of the MVS systems and system groups in the complex. For each system or system group, the display shows a date and time stamp that indicates the level of its SMS configuration, and the synchronizing interval value for its SMS subsystem (not the same as DINTERVAL). This synchronizing interval is the number of seconds that an SMS subsystem delays before synchronizing with the other SMS subsystems in the complex. A longer interval setting enables a slower system to avoid being locked out from accessing the communications data set. For more information on the SMS control data sets, the levels of SMS configuration, and the synchronizing interval, see *z/OS DFSMS Storage Administration Reference*.

CACHE

Displays the following information for each 3990-3 or 3990-6 control unit that has at least one system-managed volume attached to it:

SSID Four-character identifier for the subsystem

SMSCNT Number of SMS volumes attached to the cache

READ CONTROL

Percentage of reads and non-retentive writes for SMS-controlled data sets that will use the cache

FAST WRITE CONTROL

Percentage of writes with SMS-controlled data that will use the fast write feature

READ HIT RATIO

Percentage of I/O requests that make a hit in the cache

FAST WRITE RATE

Number of DASD fast write bypasses per minute due to non-volatile storage (NVS) overload.

CFCACHE(*structurename* or *)

Displays information about cache structures in the coupling facility. Specify *structurename* to display information for a given structure. Specify '*' to display information for all cache structures.

CFLS

Displays the following information about the coupling facilities lock structure:

- Size
- Status
- Contention rate
- False contention rate

CFVOL(*volid*)

Displays a list of coupling facilities cache structures that contain data for the specified volume (*volid*) and the status of the volume.

CICSVR[,{ALL|LOGSTREAMS(LogstreamName|ALL)|RCDS}]

Displays overall information concerning the CICSVR address space.

[ALL] is specified. The command returns the requested information from all of the active CICSVR address spaces within the sysplex.

[LOGSTREAMS(LogstreamName|ALL)] allows the operator to view all the logstreams that are currently connected to the CICSVR address space. If ALL is specified, the system displays information about all the logstreams in use and known to CICSVR on the system on which the command is issued. If a LogstreamName is specified, the system displays only the information regarding that specific logstream.

[RCDS] returns the information about the Recovery Control data sets in the CICSVR address spaces.

DETAIL

Displays detailed status information for tape and optical libraries, tape and optical storage groups and optical drives (in messages CBR1110I, CBR1120I, and CBR1130I).

If you specify a system-managed tape library name, then the system displays more detailed information about the named system-managed tape library. If you issue this command from a TSO/E terminal in OPERATOR mode, you cannot obtain detailed status for optical drives or libraries.

Note: When you specify the DETAIL keyword, you cannot specify the LISTDRI keyword.

DRIVE(name or ALL)

Displays system connectivity and the online/offline status of optical drives only. When the drive name is specified, the status for that drive is shown in a single line display. When ALL is specified the status for all the optical drives is shown.

To display the status of a drive named ALL, place the keyword in double parentheses, as DRIVE((ALL)).

Tip: To obtain the online or offline status of devices within a tape library, use the DISPLAY UNIT, DEVSERV or LIBRARY DISPDRV command.

DSNAME(dsn)

For a given fully qualified data set name, displays the jobs currently accessing the data set using DFSMS Transactional VSAM Services (DFSMStvs) access on the systems within the sysplex. If you specify WTOR, the system will issue a WTOR if the display output exceeds 255 lines.

JOB(jobname)

Displays information about a particular job that is using DFSMStvs services on one of the systems in the sysplex. The output includes:

- The name of the current step within the job
- The current URID for the job
- The status of the unit of recovery (in-reset, in-flight, in-prepare, in-commit, in-backout, indoubt)

If you specify WTOR, the system will issue a WTOR if the display output exceeds 255 lines.

LIBRARY(name or ALL)

Displays system connectivity and the online or offline status of the tape and optical libraries. Specify the library name to display the status on a single line for the named library. Specify ALL to display the status for all tape and optical libraries.

DISPLAY SMS Command

If both optical libraries and system-managed tape libraries are defined in the SMS configuration, then the system or system group displays the optical library information followed by the system-managed tape library information.

To display a library named ALL, place the name in double parentheses, as LIBRARY((ALL)).

LSTDRI

Displays the offline or online status for all the optical drives associated with the specified libraries.

Restriction: When you specify the LSTDRI keyword, you cannot specify the DETAIL keyword.

LOG(logstreamid or ALL)

Displays information about a log stream that DFSMStvs is currently using on one of the systems in the sysplex. If ALL is specified, information is displayed about all of the logs in use on the entire sysplex. The output includes the status of the log stream (failed or available), type of log (undo, shunt, forward recovery, or log of logs), the job name and URID of the oldest unit of recovery using the log, and a list of all DFSMStvs instances that are using the log. If information about a specific log stream is requested and the log stream is either a system log or a forward recovery log, the output includes the names of the jobs using the log stream.

This command might be issued to determine why a log stream is increasing in size. If a unit of recovery is long running, DFSMStvs would be unable to delete any log blocks that contain data associated with the unit of recovery, which in turn would make truncation of the log stream impossible.

If you specify WTOR, the system will issue a WTOR if the display output exceeds 255 lines.

MONDS(specmask or *)

Displays the data set specifications eligible for coupling facilities statistics monitoring. You can specify a full or partial data set name (*specmask*) to view a subset of the data set specifications. You must specify at least one high-level qualifier. A wildcard in the data set name cannot be followed by additional qualifiers.

Specify an asterisk (*) to display all the data set specifications eligible for coupling facilities statistics monitoring.

OAM

If both optical libraries and tape libraries are defined in the SMS configuration, then the system displays the optical library information followed by the tape library information.

Note: This operand is not valid when issued from a TSO/E terminal in OPERATOR mode.

OPTIONS

Displays all of the SMS parameters and their status and displays the setting of the FAST_VOLSEL parameter at the time this command is issued. The display indicates whether each option is on or off, what data sets are being used, the size of regions, the time interval for recording data, and all other parameter specifications.

For a description of the output, use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books to see message IGD002I.

When DFSMStvs is running on the system, the output of this command includes DFSMStvs-related information.

OSMC

Displays the status of the OAM storage management component (OSMC).

Note: This operand is not valid when issued from a TSO/E terminal in OPERATOR mode.

TASK(name)

Displays the OSMC status for a specific task.

STATUS

Displays online/offline status for tape or optical libraries or optical drives.

{PDSEIPDSE1} ,LATCH(*laddr*), {DETAILED|SUMMARY}

Display the status of PDSE latch at latch address (*laddr*) for the SMSPDSE or SMSPDSE1 address space. See message IGW045I for information provided by this command.

{PDSEIPDSE1} ,MODULE(*modname*)

Display the address and maintenance level of module name (*modname*) for the SMSPDSE or SMSPDSE1 address space. This command is provided to help users when a SLIP is needed for a particular module. See message IGW046I for information provided by this command.

SEP

Display the name of the active data separation profile.

SHCDS

Displays the following information about the sharing control data sets. (SHCDS):

- Name
- Size
- Amount of free space for the active and spare SHCDS
- Whether the data set is usable

SHUNTED{, SPHERE(*sphere*)|URID({urid|ALL})}}

Displays the entries currently contained in the shunt logs of the systems in the sysplex. Entries are moved to the shunt log when DFSMStvs is unable to finish processing a syncpoint, for example, due to an I/O error. As long as a shunted entry exists, the locks associated with that entry are retained.

Three types of information that can be displayed in response to this command:

- When neither the SPHERE nor URID keyword is specified, this command results in a list of systems in the sysplex and the number of units of recovery which that system has shunted
- When the SPHERE keyword is specified, this command results in a list of shunted work for the sphere specified for all of the systems in the sysplex
- When the URID keyword is specified, this command results in a list of shunted work for the unit of recovery specified for all of the systems in the sysplex. When ALL is specified, this command results in a list of shunted work for all shunted units of recovery for all the systems in the sysplex. To avoid flooding the console, DFSMStvs writes out 255 lines and then issues a WTOR to determine whether or not to continue.

If the error is correctable, the installation might choose to fix the problem and then request that DFSMStvs again attempt processing of the entry by issuing the SHCDS RETRY command. If the data set cannot be restored to a point where it is consistent with the log entry, so that it does not make sense to

DISPLAY SMS Command

attempt processing of the log entry again, the installation might choose to discard the log entry by issuing the SHCDS PURGE command.

If you specify WTOR, the system will issue a WTOR if the display output exceeds 255 lines.

SHCDS

Displays the following information about the sharing control data sets. (SHCDS):

- Name
- Size
- Amount of free space for the active and spare SHCDS
- Whether the data set is usable

SMSVSAM [,ALL]

Displays the status of the SMSVSAM server on this system or all the SMSVSAM servers and lock table connection status.

SMSVSAM,QUIESCE

Displays the status of all active VSAM record-level sharing (VSAM/RLS) sphere quiesce events **on the system that the command is entered**. (This is not a SYSPLEX-wide command.)

STATUS

Displays the online or offline status for tape or optical libraries or optical drives.

STORGRP (storgrp) or SG (storgrp) [,LISTVOL]

Displays the status of the storage group for each MVS system or system group in the SMS complex. If LISTVOL is specified, all the volumes in the storage group and their SMS status are displayed. The status of the storage group is displayed for each MVS system or system group in the SMS complex.

To display a drive named ALL, place the name in double parentheses, as STORGRP((ALL)) or SG((ALL)).

STORGRP(ALL) or SG(ALL) [,LISTVOL]

Displays a list of all storage groups in the SMS configuration, indicating by symbols (. + - * Q D) the status of each storage group for each MVS system or system group.

If you specify LISTVOL, the system displays the following information:

- A list of volumes in the storage group, giving the status of each volume for each MVS system or system group in the complex.
- The device number of the volume on the system or system group at which the command is issued.

If ALL is specified and no storage groups are defined in the active configuration, the system displays this message:

NO STORAGE GROUPS DEFINED IN THE ACTIVE CONFIGURATION

To display a drive named ALL, place the name in double parentheses, as STORGRP((ALL)) or SG((ALL)).

TRACE or T

Displays the SMS trace options in effect at the time the command was issued, followed by the setting of the parameters that are related to SMS volume selection analysis messages. The display indicates whether each trace option is now on or off. For a description of the output use LookAt (see "Using LookAt to look up message explanations" on page xviii) or use the *MVS System Messages* books to see message IGD002I.

TRANVSAM [,ALL]

Displays information about the instance of DFSMStvs on this system, or on all systems in the sysplex when the ALL keyword is specified. The output includes this information:

- The activity keypoint (AKP) trigger, which is the number of logging operations between the taking of keypoints
- The status of this instance of DFSMStvs (initializing, active, quiescing, quiesced, disabling, disabled)
- How DFSMStvs started:
 - Cold start
The log data was not read, and any old data was discarded.
 - Warm start
The log data was read and processed.
- DFSMStvs status with respect to resource recovery services (RRS)
- The quiesce timeout value
- All logs known to this instance of DFSMStvs, including the log of logs if one is in use
- The number of active units of recovery
- The status of all known logs associated with this TVS instance, if you specify ALLLOGS. Otherwise, only the UNDO and SHUNT logs are returned

If you specify WTOR, the system will issue a WTOR if the display output exceeds 255 lines.

URID({urid|ALL})

Displays information about a particular unit of recovery currently active within the sysplex or about all units of recovery currently active on the system on which the command was issued on whose behalf DFSMStvs has performed any work. This parameter does not include information about work that has been shunted because you can use the DISPLAY SMS,SHUNTED command to display that information. This parameter also does not include information about units of recovery that might be in restart processing as a result of an earlier failure. This work is not considered to be currently active because it is not associated with any batch job, and the units of recovery associated with the work will end as soon as commit or backout processing for them can be completed. The output includes this information:

- The age of the unit of recovery
- The name of the job with which the unit of recovery is associated
- The name of the current step within the job
- The status of the unit of recovery (in-reset, in-flight, in-prepare, in-commit, in-backout, indoubt)
- The user ID associated with the job

If you specify WTOR, the system will issue a WTOR if the display output exceeds 255 lines.

VOLUME or VOL(*volume*)

For a DASD volume, the system displays detailed status in the pool storage group with respect to the systems or system groups in the complex. The status indication can be NOT DEFINED TO THE SYSTEM, ENABLED, DISABLED, QUIESCED, DISABLED FOR NEW ALLOCATIONS ONLY, or QUIESCED FOR NEW ALLOCATIONS ONLY. The display also gives the device number of the device on which the volume is mounted. This occurs only on the system or system group at which the command is issued.

DISPLAY SMS Command

For an optical volume, the system displays (in message CBR1140I) detailed status from the optical configuration database.

For a system-managed tape volume, the system displays (in message CBR1180I) detailed status from the tape configuration database.

If the specified volume is not system managed, this message is issued:
COMMAND REJECTED; VOLUME *volume* IS NOT DEFINED

To display a volume named ALL, place the name in double parentheses, as VOLUME((ALL)) or VOL((ALL)).

VOLSELMMSG

Displays the setting of the parameters that are related to SMS volume selection analysis messages, followed by the setting of parameters that are related to SMS trace.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example 1

Either of the following commands displays the active SMS configuration:

d sms

d sms,a

In this example, three MVS systems are defined to SMS, but only the first two systems have an active Storage Management Subsystem. The third system is either not IPLed or was IPLed without starting SMS.

The response to this command is as follows:

```
IGD002I 10:13:25 DISPLAY SMS 050
SCDS = STAGE2.SCDSPLEX.DATA
ACDS = SYS1.PDSESHR.ACDS
COMMDS = SYS1.PDSESHR.COMMDS
DINTERVAL = 150
REVERIFY = NO
ACSDEFAULTS = YES
SYSTEM CONFIGURATION LEVEL INTERVAL SECONDS
SYSTEM1 2005/03/18 10:13:10 15
SYSTEM2 ----- ----- N/A
SYSTEM3 ----- ----- N/A
SYSTEM4 ----- ----- N/A
SYSTEM5 ----- ----- N/A
SYSTEM6 ----- ----- N/A
SYSTEM7 ----- ----- N/A
SYSTEM8 ----- ----- N/A
```

Example 2

To display the name of the active data separation file, enter:

d sms,sep

In response to this command, the system issues message IGD002I:

```
IGD002I 09:59:03 DISPLAY SMS 047
ACTIVE DATA SET SEPARATION PROFILE NAME: sep_dname
```

If there are no active separation profile, the system issues message IGD002I as:

```
IGD002I 09:59:03 DISPLAY SMS 047
NO DATA SET SEPARATION PROFILE IS ACTIVE
```

Example 3

To display the status of the storage group **sgmixed**, enter:

```
d sms,sg(sgmixed)
```

The response to this command is as follows:

```
IGD002I 11:50:52 DISPLAY SMS 448
STORGRP      TYPE          SYSTEM= 1   3   4   5   6   7   8
SGMIXED      TYPE          +   .   .   .   .   .   .
***** LEGEND *****
. THE STORAGE GROUP OR VOLUME IS NOT DEFINED TO THE SYSTEM
+ THE STORAGE GROUP OR VOLUME IS ENABLED
- THE STORAGE GROUP OR VOLUME IS DISABLED
* THE STORAGE GROUP OR VOLUME IS QUIESCED
D THE STORAGE GROUP OR VOLUME IS DISABLED FOR NEW ALLOCATIONS ONLY
Q THE STORAGE GROUP OR VOLUME IS QUIESCED FOR NEW ALLOCATIONS ONLY
SYSTEM 1 = SYSTEM1
IEF196I IEF285I   SYS2.LINKLIB           KEPT
IEF196I IEF285I   VOL SER NOS= MVSRES.
```

Example 4

To display storage group **group26**, showing the states of all its volumes, enter:

```
d sms,storgrp(group26),listvol
```

The output from this command is similar to that of the previous example except that the specific volumes defined to each system are listed.

```
IGD002I 11:51:34 DISPLAY SMS 453
STORGRP      TYPE          SYSTEM= 1   7   8
GROUP26      OBJECT        +   .
LISTVOL IS IGNORED FOR OBJECT, OBJECT BACKUP, AND TAPE STORAGE GROUPS STORGRPNAMES
***** LEGEND *****
. THE STORAGE GROUP OR VOLUME IS NOT DEFINED TO THE SYSTEM
+ THE STORAGE GROUP OR VOLUME IS ENABLED
- THE STORAGE GROUP OR VOLUME IS DISABLED
* THE STORAGE GROUP OR VOLUME IS QUIESCED
D THE STORAGE GROUP OR VOLUME IS DISABLED FOR NEW ALLOCATIONS ONLY
Q THE STORAGE GROUP OR VOLUME IS QUIESCED FOR NEW ALLOCATIONS ONLY
SYSTEM 1 = SYSTEM1
```

Example 5

To illustrate the display of trace options, assume that the following two commands have been issued for system MVS3:

```
SETSMS TRACE(ON),SIZE(128K),TYPE(ERROR),JOBNAME(*)
SETSMS DESELECT(ALL),SELECT(ACSINT,CONFC,MSG),ASID(*)
```

The first command turns on tracing and sets the trace table size to 128KB. The second command turns off all trace options except for the three options indicated. To display the SMS trace options now in effect, enter:

```
DISPLAY SMS,TRACE
```

DISPLAY SMS Command

The output from this command is similar to that shown below. Note that the trace is shown to be *on*, the trace table size is indicated, and all options are shown to be *off* except for the three turned *on* by the SETSMS...,SELECT command.

```
IGD002I 10:34:39 DISPLAY SMS 053
TRACE = ON SIZE = 128K TYPE = ERROR
JOBNAME = * ASID = *
TRACING EVENTS:
MODULE = ON SMSSJF = ON SMSSSI = ON ACSINT = ON
OPCMD = ON CONFC = ON CDSC = ON CONFS = ON
MSG = ON ERR = ON CONFR = ON CONFA = ON
ACSPRO = ON IDAX = ON DISP = ON CATG = ON
VOLREF = ON SCHEDP = ON SCHEDS = ON
VTOCL = ON VTOCD = ON VTOCR = ON VTOCC = ON VTOCA = ON
RCD = ON DCF = ON DPN = ON TVR = ON
DSTACK = ON UAFF = ON

VOLSELMMSG = (OFF,0) TYPE = ERROR JOBNAME = *
ASID = * STEPNAME = *
DSNAME = *
```

Example 6

To display the SMS options now in effect, enter:

```
12.16.35      d sms,options
```

The output from this command is similar to the following output.

```
12.16.35 SYSTEM1      IGD002I 13:01:52 DISPLAY SMS 918
ACDS = SYS1.PDSESHR.ACDS
COMMDS = SYS1.PDSESHR.COMMDS
INTERVAL = 15 DINTERVAL = 150
SMF TIME = YES CACHETIME = 3600
CF_TIME = 3600 PDSE_RESTARTABLE_AS = YES
PDSE_BMFTIME = 3600 PDSE1_BMFTIME = 3600
PDSE_LRUTIME = 60 PDSE1_LRUTIME = 60
PDSE_LRUCYCLES = 15 PDSE1_LRUCYCLES = 15
LOCAL_DEADLOCK = 15 GLOBAL_DEADLOCK = 4
REVERIFY = NO DSNTYPE = LIBRARY
ACSDEFAULTS = YES PDSESHARING = EXTENDED
OVRD_EXPDT = NO SYSTEMS = 8
PDSE_HSP_SIZE = 256MB PDSE1_HSP_SIZE = 256MB
USE_RESOWNER = YES RLS_MAX_POOL_SIZE = 100MB
RLSINIT = YES RLSTMOUT = 500
COMPRESS = GENERIC LOG_OF_LOGS = IGWTVS1.LOG.OF.LOGS
QTIMEOUT = 400 TVSNAME = 001
AKP = 200 TV_START_TYPE = WARM
MAXLOCKS = (100,50)
CICSVR_INIT = NO
CICSVR_DSNAME_PREFIX = DWW.
CICSVR_RCDS_PREFIX = DWW
CICSVR_GRPNAME_SUFFIX = PROD
CICSVR_ZZVALUE_PARM =
CICSVR_UNDOLOG_CONTROL =
CICSVR_UNDOLOG_PREFIX = DWW
CICSVR_BACKOUT_CONTROL =
CICSVR_GENERAL_CONTROL =
R1s_MaxCffFeatureLevel = A
R1sAboveThebarMaxPoolSize = 0
R1sFixedPoolSize = 0
PDSE_MONITOR = (YES,0,0) PDSE1_MONITOR = (YES,0,0)
PDSE_BUFFER_BEYOND_CLOSE = NO
PDSE1_BUFFER_BEYOND_CLOSE = NO
```

```

GDS_RECLAIM = YES DSSTIMEOUT = 0
BLOCKTOKENSIZE = REQUIRE

IGD002I 12:16:35 DISPLAY SMS 806
TRACE = ON      SIZE = 128K      TYPE = ERROR
JOBNAME = *     ASID = *
TRACING EVENTS:
  MODULE = ON   SMSSJF = ON   SMSSSI = ON   ACSINT = ON
  OPCMD = ON    CONFC = ON   CDSC = ON    CONFS = ON
  MSG = ON      ERR = ON    CONFR = ON   CONFA = ON
  ACSPRO = ON   IDAX = ON   DISP = ON    CATG = ON
  VOLREF = ON   SCHEDP = ON  SCHEDS = ON  VTOCL = ON
  VTOCD = ON    VTOCR = ON  VTOCC = ON  VTOCA = ON
  RCD = ON      DCF = ON   DPN = ON    TVR = ON
  DSTACK = ON   UAFF = ON
VOLSELMMSG = (OFF,0) TYPE = ERROR JOBNAME = *
ASID = * STEPNAME = *
DSNAME = *

```

Example 7

The DISPLAY SMS,TRANVSAM command displays information about the status of DFSMStvs.

If DFSMStvs is not active, this results in:

```

IGW800I DFSMS VSAMRLS REQUEST TO DISPLAY TRANSACTIONAL VSAM INFORMATION
IS REJECTED: TRANSACTIONAL VSAM IS NOT ACTIVE ON THIS SYSTEM

```

If DFSMStvs is active, this command results in the following output:

```

D SMS,TRANVSAM
IEE932I 774
IGW800I 13.33.01 DISPLAY SMS,TRANSACTIONAL VSAM
DISPLAY SMS,TRANSACTIONAL VSAM - SERVER STATUS
System    TSVNAME  State   Rrs #Urs Start    AKP   QtimeOut
-----
SYSTEM1  IGWTVO01 ACTIVE REG          1 WARM/WARM    200    400
DISPLAY SMS,TRANSACTIONAL VSAM - LOGSTREAM STATUS
LogStreamName       State   Type   Connect Status
-----
IGWTVO01.IGWLOG.SYSLOG Enabled  UndoLog  Connected
IGWTVO01.IGWSHUNT.SHUNTLG Enabled ShuntLog  Connected
IGWTS1.LOG.OF.LOGS    Enabled  LogOfLogs Connected
IGWTS.FR.LOG001      Enabled  FrLog    Connected

```

Example 8

To display information about DFSMStvs, enter this command:

```
D SMS,TRANVSAM,ALL
```

The output from this command

```

D SMS,TRANVSAM,ALL
IEE932I 777
IGW800I 13.33.12 DISPLAY SMS,TRANSACTIONAL VSAM,ALL
DISPLAY SMS,TRANSACTIONAL VSAM,ALL - SERVER STATUS
System    TSVNAME  State   Rrs #Urs Start    AKP   QtimeOut
-----
SYSTEM1  IGWTVO01 ACTIVE REG          1 WARM/WARM    200    400
SYSTEM2  IGWTVO02 ACTIVE REG          0 WARM/WARM    200    400
DISPLAY SMS,TRANSACTIONAL VSAM,ALL LOGSTREAM STATUS
LogStreamName: IGWTVO01.IGWLOG.SYSLOG

```

DISPLAY SMS Command

System	TVSNAME	State	Type	Connect Status
SYSTEM1	IGWTV001	Enabled	UnDoLog	Connected
LogStreamName: IGWTV001.IGWSHUNT.SHUNTLG				
System	TVSNAME	State	Type	Connect Status
SYSTEM1	IGWTV001	Enabled	ShuntLog	Connected
LogStreamName: IGWTVS1.LOG.OF.LOGS				
System	TVSNAME	State	Type	Connect Status
SYSTEM1	IGWTV001	Enabled	LogOfLogs	Connected
SYSTEM2	IGWTV002	Enabled	LogOfLogs	Connected
LogStreamName: IGWTV002.IGWLOG.SYSLOG				
System	TVSNAME	State	Type	Connect Status
SYSTEM2	IGWTV002	Enabled	UnDoLog	Connected
LogStreamName: IGWTV002.IGWSHUNT.SHUNTLG				
System	TVSNAME	State	Type	Connect Status
SYSTEM2	IGWTV002	Enabled	ShuntLog	Connected
LogStreamName: IGWTVS.FR.LOG001				
System	TVSNAME	State	Type	Connect Status
SYSTEM1	IGWTV001	Enabled	FrLog	Connected

Example 9

The DISPLAY SMS,JOB command displays information about the status of a job.

If the job is using DFMSMStvs services, this command results in the following output:

```
D SMS,JOB(TVS30601)
IEE932I 780
IGW801I 13.33.27 DISPLAY SMS,JOB
TRANSACTIONAL VSAM Job Status On System: SYSTEM1
JobName StepName Urid Ur Status # Locks
-----
TVS30601 STEP04 B4E070267EAFD0000000000301010000 IN_FLIGHT 20
```

Example 10

The DISPLAY SMS,UR command displays information about a particular unit of recovery on whose behalf DFMSMStvs has performed any work.

If the unit of recovery is not active, this command results in the following output:

```
IGW802I DFSMS REQUEST TO DISPLAY ACTIVE TRANSACTIONAL VSAM UR(s)
WAS REJECTED, SPECIFIED URID(s) ARE NOT ACTIVE
ON ANY TRANSACTIONAL VSAM INSTANCE IN THE SYSPLEX.
```

If the UR is currently active, this command results in the following output:

```
D SMS,URID(ALL)
IEE932I 783
IGW802I 13.33.39 DISPLAY SMS,URID
TRANSACTIONAL VSAM UrId Status For System: SYSTEM1
Urid Ur Status JobName StepName # Locks
-----
B4E070267EAFD0000000000301010000 InFlight TVS30601 STEP04 20
```

Example 11

Use the DISPLAY SMS,SHUNTED command to display information about URs currently in the DFSMStvs shunt log.

If there is currently no shunted work, the results are as follows:

```
IGW803I 09.50.47 DISPLAY SMS,UR (Summary Data)
SysName # Urid(s) SysName # Urid(s) SysName # Urid(s)
----- -----
SYSTEM1      0 ----- 0 ----- 0
```

If there are shunted URs, the results are as follows:

```
IGW803I 09.50.47 DISPLAY SMS,UR (Summary Data)
SysName # Urid(s) SysName # Urid(s) SysName # Urid(s)
----- -----
SYSTEM1 urid ----- 0 ----- 0
```

Example 12

Use the DISPLAY SMS,LOG command to display information about the log streams that DFSMStvs is using.

If the log stream is not currently in use by DFSMStvs, this command results in the following output:

```
IGW804I DFSMS REQUEST TO DISPLAY
TRANSACTIONAL VSAM LOG STREAM: Logstream WAS REJECTED.
```

LOG STREAM NOT KNOWN TO DFSMS.

If the log stream is currently in use, this command results in the following output:

```
D SMS,LOG(IGWTVS.FR.LOG001)
IEE932I 789
IGW804I 13.34.10 DISPLAY SMS,LOG
DISPLAY SMS,LOG - LOG STREAM STATUS
  Name: IGWTVS.FR.LOG001      State: Enabled      Type: FrdRecovr
  System   TVSNAME JobName Urid of Oldest Log Block
-----
SYSTEM1 IGWTV001 TVS30601 B4E070267EAFD000000000301010000*
DISPLAY SMS,LOG - LOG STREAM USAGE
  LogStreamName: IGWTVS.FR.LOG001
  System   TVSNAME JobName JobName JobName JobName JobName
-----
SYSTEM1 IGWTV001 TVS30601
*OLDEST URID ACROSS ALL SYSTEMS IN THE SYSPLEX
```

Example 13

The DISPLAY SMS,LOG(ALL) command displays information about the log streams that DFSMStvs is using.

D SMS,LOG(ALL)

This command results in the following output:

IEE932I 792

```
IGW804I 13.34.18 DISPLAY SMS,LOG
DISPLAY SMS,LOG - LOG STREAM STATUS
  Name: IGWTVS.FR.LOG001      State: Enabled      Type: FrdRecovr
  System   TVSNAME JobName Urid of Oldest Log Block
-----
SYSTEM1 IGWTV001 TVS30601 B4E070267EAFD000000000301010000*
```

DISPLAY SMS Command

```
DISPLAY SMS,LOG      - LOG STREAM USAGE
  LogStreamName: IGWTVS.FR.LOG001
  System    TVSNAME JobName JobName JobName JobName JobName
  -----
  SYSTEM1  IGWTV001 TVS30601
  *OLDEST URID ACROSS ALL SYSTEMS IN THE SYSPLEX
  DISPLAY SMS,LOG      - LOG STREAM STATUS
    Name: IGWTVS1.LOG.OF.LOGS      State: Enabled      Type: LogOfLogs
  System    TVSNAME JobName Urid of Oldest Log Block
  -----
  SYSTEM1  IGWTV001 TVS30601 B4E070267EAFD0000000000301010000*
  DISPLAY SMS,LOG      - LOG STREAM USAGE
  LogStreamName: IGWTVS1.LOG.OF.LOGS
  System    TVSNAME JobName JobName JobName JobName JobName
  -----
  SYSTEM1  IGWTV001 TVS30601
  *OLDEST URID ACROSS ALL SYSTEMS IN THE SYSPLEX
  DISPLAY SMS,LOG      - LOG STREAM STATUS
    Name: IGWTV001.IGWLOG.SYSLOG      State: Enabled      Type: UnDo
  System    TVSNAME JobName Urid of Oldest Log Block
  -----
  SYSTEM1  IGWTV001 TVS30601 B4E070267EAFD0000000000301010000*
  DISPLAY SMS,LOG      - LOG STREAM USAGE
  LogStreamName: IGWTV001.IGWLOG.SYSLOG
  System    TVSNAME JobName JobName JobName JobName JobName
  -----
  SYSTEM1  IGWTV001 TVS30601
  *OLDEST URID ACROSS ALL SYSTEMS IN THE SYSPLEX
  DISPLAY SMS,LOG      - LOG STREAM STATUS
    Name: IGWTV001.IGWSHUNT.SHUNTLOG State: Enabled      Type: Shunt
  System    TVSNAME JobName Urid of Oldest Log Block
  -----
  SYSTEM1  IGWTV001      ----- NO ACTIVE UR -----*
  DISPLAY SMS,LOG      - LOG STREAM USAGE
  LogStreamName: IGWTV001.IGWSHUNT.SHUNTLOG
  System    TVSNAME JobName JobName JobName JobName JobName
  -----
  SYSTEM1  IGWTV001 **NONE**
  *OLDEST URID ACROSS ALL SYSTEMS IN THE SYSPLEX
  DISPLAY SMS,LOG      - LOG STREAM STATUS
    Name: IGWTVS1.LOG.OF.LOGS      State: Enabled      Type: LogOfLogs
  System    TVSNAME JobName Urid of Oldest Log Block
  -----
  SYSTEM2  IGWTV002      ----- NO ACTIVE UR -----*
  DISPLAY SMS,LOG      - LOG STREAM USAGE
  LogStreamName: IGWTVS1.LOG.OF.LOGS
  System    TVSNAME JobName JobName JobName JobName JobName
  -----
  SYSTEM2  IGWTV002 **NONE**
  *OLDEST URID ACROSS ALL SYSTEMS IN THE SYSPLEX
  DISPLAY SMS,LOG      - LOG STREAM STATUS
    Name: IGWTV002.IGWLOG.SYSLOG      State: Enabled      Type: UnDo
  System    TVSNAME JobName Urid of Oldest Log Block
  -----
  SYSTEM2  IGWTV002      ----- NO ACTIVE UR -----*
  DISPLAY SMS,LOG      - LOG STREAM USAGE
  LogStreamName: IGWTV002.IGWLOG.SYSLOG
  System    TVSNAME JobName JobName JobName JobName JobName
  -----
  *OLDEST URID ACROSS ALL SYSTEMS IN THE SYSPLEX
  DISPLAY SMS,LOG      - LOG STREAM STATUS
    Name: IGWTV002.IGWSHUNT.SHUNTLOG State: Enabled      Type: Shunt
  System    TVSNAME JobName Urid of Oldest Log Block
```

```
-----  
SYSTEM2 IGWTV002 ----- NO ACTIVE UR -----*  
DISPLAY SMS,LOG - LOG STREAM USAGE  
LogStreamName: IGWTV002.IGWSHUNT.SHUNTLG  
System TVSNAME JobName JobName JobName JobName JobName  
-----
```

Example 14

The DISPLAY SMS,DSNAME command display information about the jobs that have a data set open for DFSMStvs access.

If the data set is not currently open for DFSMStvs access, this command results in the following output:

```
IGW805I DFSMS REQUEST TO DISPLAY TRANSACTIONAL VSAM USAGE OF  
DATASET: dsname WAS REJECTED.
```

```
DATASET NOT KNOWN TO TRANSACTIONAL VSAM.
```

If the data set is currently open for DFSMStvs access, this command results in the following output:

```
D SMS,DSNAME(SYSPLEX.SHCD.SUND01.KSDS01)  
IEE932I 795  
IGW805I 13.34.44 DISPLAY SMS,DSNAME  
DATASET: SYSPLEX.SHCD.SUND01.KSDS01  
IS CURRENTLY IN USE BY THE FOLLOWING JOBS:
```

```
System Name: SYSTEM1 TVSNAME: IGWTV001  
JobNames: TVS30601
```

Example 15

The DISPLAY SMS,OPTIONS command displays the values with which SMS, SMSVSAM, and DFSMStvs are currently operating.

This command results in the following output:

```
D SMS,OPTIONS  
IGD002I 13:01:52 DISPLAY SMS 918  
ACDS = SYS1.PDSESHR.ACDS  
COMMDS = SYS1.PDSESHR.COMMDS  
INTERVAL = 15 DINTERVAL = 150  
SMF_TIME = YES CACHETIME = 3600  
CF_TIME = 3600 PDSE_RESTARTABLE_AS = YES  
PDSE_BMFTIME = 3600 PDSE1_BMFTIME = 3600  
PDSE_LRUTIME = 60 PDSE1_LRUTIME = 60  
PDSE_LRUCYCLES = 15 PDSE1_LRUCYCLES = 15  
LOCAL_DEADLOCK = 15 GLOBAL_DEADLOCK = 4  
REVERIFY = NO DSNTYPE = LIBRARY  
ACSDEFAULTS = YES PDSESHARING = EXTENDED  
OVRD_EXPT = NO SYSTEMS = 8  
PDSE_HSP_SIZE = 256MB PDSE1_HSP_SIZE = 256MB  
USE_RESOWNER = YES RLS_MAX_POOL_SIZE = 100MB  
RLSINIT = YES RLSTMOUT = 500  
COMPRESS = GENERIC LOG_OF_LOGS = IGWTVS1.LOG.OF.LOGS  
QTIMEOUT = 400 TSVNAME = 001  
AKP = 200 TV_START_TYPE = WARM  
MAXLOCKS = (100,50)  
CICSVR_INIT = NO CICSVR_DSNAME_PREFIX = DWW.  
CICSVR_RCDs_PREFIX = DWW  
CICSVR_GRPNAME_SUFFIX = PROD
```

DISPLAY SMS Command

```
CICSVR_ZZVALUE_PARM =  
CICSVR_UNDOLOG_CONTROL =  
CICSVR_UNDOLOG_PREFIX = DWW  
CICSVR_BACKOUT_CONTROL =  
CICSVR_GENERAL_CONTROL =  
R1s_MaxCffFeatureLevel = A  
R1sAboveThebarMaxPoolSize = 0  
R1sFixedPoolSize = 0  
PDSE_MONITOR = (YES,0,0) PDSE1_MONITOR = (YES,0,0)  
PDSE_BUFFER_BEYOND_CLOSE = NO  
PDSE1_BUFFER_BEYOND_CLOSE = NO  
GDS_RECLAIM = YES DSSTIMEOUT = 0  
BLOCKTOKENSIZE = REQUIRE  
IGD002I 13:01:52 DISPLAY SMS  
TRACE = ON SIZE = 128K TYPE = ERROR  
JOBNAME = * ASID = *  
TRACING EVENTS:  
MODULE = ON SMSSJF = ON SMSSSI = ON ACSINT = ON  
OPCMD = ON CONFC = ON CDSC = ON CONFS = ON  
MSG = ON ERR = ON CONFR = ON CONFA = ON  
ACSPRO = ON IDAX = ON DISP = ON CATG = ON  
VOLREF = ON SCHEDP = ON SCHEDS = ON VTOCL = ON  
VTOCD = ON VTOCR = ON VTOCC = ON VTOCA = ON  
RCD = ON DCF = ON DPN = ON TVR = ON  
DSTACK = ON UAFF = ON  
VOLSELMSG = (OFF,0) TYPE = ERROR JOBNAME = *  
ASID = * STEPNAME = *  
DSNAME = *
```

Example 16

To display the setting of the parameters that are related to SMS volume selection analysis messages, enter:

```
D SMS,VOLSELMSG
```

The response to this command is as follows:

```
IGD002I 13:58:46 DISPLAY SMS 789  
VOLSELMSG = (OFF,0) TYPE = ERROR JOBNAME = *  
ASID = * STEPNAME = *  
DSNAME = *  
TRACE = ON SIZE = 128K TYPE = ERROR  
JOBNAME = * ASID = *
```

Displaying Information about All Subsystems

Use the DISPLAY SSI command to display the following information about all subsystems defined to MVS:

- Whether the subsystem is dynamic
- Whether the subsystem is active
- For a dynamic subsystem, whether it accepts or rejects dynamic SSI commands such as SETSSI.
- For an active subsystem, the function codes it supports.

An operator can use keyword filters to specify the information to be displayed from those subsystems that meet the specified criteria. For example, an operator may choose to display information about a particular subsystem by specifying its name.

The output from the DISPLAY SSI command is a multi-line message. It is written to the console from which the command was issued or to the specified console.

```
D SSI[,{LIST|L}|{ALL|A}][,{DYNAMIC|DYN|D}={YES|Y}|{NO|N}]
[,{FUNC|F}=funclist]
[,{STATUS|STAT|ST}={ACTIVE|ACT}|{INACTIVE|INACT|I}]
[,{SUBSYS|SUB}=subsysname]
[,L={a|name|name-a}]
```

SSI

Displays information about all subsystems defined to the SSI.

LIST or L

Displays the LIST output format, which includes the following information for each subsystem defined to the system:

- Whether the subsystem is dynamic
- Whether the subsystem is active
- For a dynamic subsystem, whether the subsystem accepts or rejects dynamic SSI commands such as SETSSI.

The LIST format is the default keyword.

ALL or A

Displays the ALL output format. This output is the same as the LIST format except that the system includes a sub-list after each list element. The sub-list contains a list of function codes to which the subsystem responds.

For the output messages of the DISPLAY SSI command, use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books to see message IEFJ100I.

DYNAMIC or DYN or D=YES or Y or No or N

Displays either dynamic or non-dynamic subsystems. If dynamic, the subsystem can use dynamic SSI services. See *z/OS MVS Using the Subsystem Interface* for more information on dynamic SSI requests.

FUNC or F=funclist

Displays those subsystems that respond to the function codes specified. The *funclist* value can be either a number no greater than three digits or a list of numbers no greater than three digits. The list of numbers must be separated by commas and enclosed in parentheses. For example, you can specify FUNC=3 or FUNC=(18,5,100).

You can use the FUNC parameter with either the LIST parameter or the ALL parameter. For either format, only those subsystems which respond to all the specified function codes appear in the display. If you use the ALL format, the list of function codes for each subsystem displayed is the complete list of all the function codes to which that subsystem responds. If you specify the FUNC parameter, inactive subsystems or subsystems without a vector table do not appear in the display.

STATUS or STAT or ST=ACTIVE or ACT or INACTIVE or INACT or I

Displays subsystems whose status is either active or inactive. Specifying ACTIVE or ACT means that displayed subsystems accept function requests directed to it by the SSI. Specifying INACTIVE or INACT or I means that displayed subsystems do not accept function requests directed to it by the SSI.

DISPLAY SSI Command

SUBSYS or SUB=subsysname

Displays information about the subsystem whose name matches the specified pattern. The pattern could be the name of the subsystem or it could contain wildcard characters.

Subsystem names that are not enclosed in apostrophes may contain any character that is valid for operator commands, with the following exceptions:

- , comma
- (left parenthesis
-) right parenthesis
- / slash
- = equals sign

Subsystem names containing these characters must be enclosed in apostrophes.

Subsystem names that contain any character that is not valid for operator commands must be enclosed in apostrophes. See Chapter 4, “MVS System Commands Reference,” on page 4-1 for a list of characters supported by commands.

You can specify an asterisk (*) or question mark (?) anywhere in the subsystem name. An asterisk (*) is a wildcard character used to replace 0 or more characters to obtain a matching name. A question mark (?) is a wildcard character used to replace one character to obtain a matching name. For example, if a system has subsystems JES2, JESA, A, SS2 and J specified: SUBSYS=JES* causes JES2 and JESA to appear in the display. SUBSYS=J* causes JES2, JESA and J to appear in the display. SUBSYS=*S2 causes JES2 and SS2 to appear in the display. SUBSYS=?S2 causes SS2 to appear in the display. SUBSYS=* causes all the subsystems to appear in the display. Note that specifying SUBSYS=* has the same effect as not specifying the SUBSYS parameter at all.

Also, you can specify the character string '!PRI' rather than a subsystem name, which causes the system to display only the primary subsystem.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Notes:

1. You cannot use the same parameter twice within a command.
2. A command line cannot be longer than 126 characters.

Example 1

To display information for all the subsystems defined to the system which:

- Are currently active
- Can use dynamic SSI services

without including the list of function codes to which the subsystems respond, enter:

D SSI,STAT=ACT,DYN=YES

Example 2

To display information for every subsystem whose name begins with 'JES' and include the list of function codes for each subsystem, enter:

D SSI,ALL,SUB=JES*

Example 3

To display information for every subsystem that responds to function codes 9 and 10 and include the list of function codes for each subsystem, enter:

D SSI,A,FUNC=(9,10)

Note: If a display in response to the command is greater than 65,533 lines, the system will truncate the output. If this happens, re-enter the DISPLAY SSI command using parameters to decrease the size of the display. For example, if D SSI,ALL yields a display that is too large, you can use D SSI,LIST to display subsystems without listing the function codes to which they respond. Then use D SSI,ALL,SUBSYS=subsysname to display the function codes for the particular subsystems of interest one subsystem at a time.

Displaying Static System Symbols

Use the DISPLAY SYMBOLS command to display the current static system symbols and their associated substitution texts.

```
D SYMBOLS[,L={a|name|name-a}]
```

SYMBOLS

The system is to display, in message IEA007I, the static system symbols defined to this system.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Example

To display the static system symbols defined to this system, enter:

D SYMBOLS

Displaying the Local and Coordinated Universal Time and Date

Use the DISPLAY T command to display the local time of day and date and the coordinated universal time (UTC) of day and date.

```
D T
```

T The local time of day and date and the coordinated universal time (UTC) of day and date are to be displayed (message IEE136I).

Displaying Component or Transaction Trace Status

Use the DISPLAY TRACE command to obtain status information about either components or applications defined to component trace, or about transaction trace.

```
D TRACE[,COMP=cname[,SUB=(subname)][,N=nnn][,SUBLEVEL]
      [,COMP={(cname[,cname]...)|ALL}                ]
      [,WTR={(name[,name]...)|ALL}                  ]
      [,TT                                         ]
      [,L={a|name|name-a}]
```

TRACE

Indicates that the system is to display status information, in short form, about the components defined to component trace.

COMP=ALL

Specifies that the system is to display status, in long form, for all components.

COMP=(cname[,cname]...)

Specifies that the system is to display component status, in long form, for the component names requested. See your system programmer for the components and applications active on the system that are defined to component trace.

SUB=(subname)

Specifies that the system is to display, in long form, the status of the specified sublevel trace. Obtain the specific names of sublevels from the system programmer. If the sub level trace name contains any national characters (@ # \$ _) then the name must be enclosed in quotes. Otherwise, quotes are not required. In either case, the alphabetic characters can be specified in upper or lower case.

The command displays information for only one sublevel trace. Enter a separate DISPLAY command for each sublevel trace.

N=nnn

Specifies that the system is to display the subordinate node status and, when the SUBLEVEL keyword is specified, the specific 'nnn' number of parallel or sublevel nodes of the requested subordinate node.

SUBLEVEL

Specifies that sublevel trace status is to be displayed.

WTR=(name[,name],...)

Displays information only about the component trace external writers you specify.

WTR=ALL

Displays information about all component trace external writers.

TT

Displays the status of the transaction trace currently in effect. In a parallel sysplex environment it displays the status of the transaction trace currently in effect in the sysplex. See Example 6.

L=a, name, or name-a

Specifies the display area (a), console name (name), or both (name-a) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Note: See *z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN* for information about application traces of application programs. See *z/OS MVS Diagnosis: Tools and Service Aids* for information about component traces of components.

Example 1

To display the status of component trace in short form, enter:

```
DISPLAY TRACE
```

The system responds with:

```
IEE843I 15.18.24 TRACE DISPLAY 447
        SYSTEM STATUS INFORMATION
ST=(ON,0016K,0016K) AS=ON BR=OFF EX=ON MT=ON,024K
COMPONENT MODE COMPONENT MODE COMPONENT MODE COMPONENT MODE
-----
SYSRSM    MIN   SYSGRS    ON    SYSVLF    PRE   SYSLLA    MIN
```

COMPONENT

The component name.

MODE

The current state of the trace.

- ON —trace is on.
- OFF —trace is off.
- MIN —this component has reduced tracing activity to the minimum required to provide serviceability data in a dump.
- PRE —this trace is PRESET. Trace options were established with a TRACE CT specifying a parmlib member containing the PRESET(DEFINE) option.

Example 2

To display the component trace status of all components.

```
DISPLAY TRACE,COMP=ALL
```

The system responds with:

```
IEE843I 15.18.24 TRACE DISPLAY 450
        SYSTEM STATUS INFORMATION
ST=(ON,0016K,0016K) AS=ON BR=OFF EX=ON MT=ON,024K
COMPONENT MODE BUFFER HEAD SUBS
-----
SYSVLF    PRE    4M
ASIDS      *NOT SUPPORTED*
JOBNAMES   *NOT SUPPORTED*
OPTIONS    COMPONENT,SPECIFIC,OPTIONS,EXAMPLE
WRITER     *NOT SUPPORTED*
-----
SYSLLA    MIN    200K HEAD 20
ASIDS      *NOT SUPPORTED*
JOBNAMES   *NOT SUPPORTED*
OPTIONS    COMPONENT,SPECIFIC,OPTIONS,EXAMPLE
WRITER     *NOT SUPPORTED*
-----
SYSRSM    MIN    4M
ASIDS      *NONE*
```

DISPLAY TRACE Command

```
JOBNAMES *NONE*
OPTIONS *NONE*
WRITER *NONE*
```

COMPONENT

The component name.

MODE

The current state of the trace.

- ON —trace is on.
- OFF —trace is off.
- MIN —this component has reduced tracing activity to the minimum required to provide serviceability data in a dump.
- PRE —this trace is PRESET. Through the PRESET(DEFINE) option in the parmlib member.

BUFFER

The buffer size, in decimal, established by PRESET(DEFINE) or by operator command when the component trace was turned on.

ASIDS

Any ASIDs, in hexadecimal, currently in use as a filter for tracing this component.

- *NOT SUPPORTED* indicates that ASIDs cannot be used as a filter for this component.
- *NONE* indicates that ASIDs can act as a tracing filter but none have been specified.

JOBNAMES

Any job names currently in use as a filter for tracing this component.

- *NOT SUPPORTED* indicates that job names cannot be used as a filter for this component.
- *NONE* indicates that job names can act as a tracing filter but none have been specified.

OPTIONS

List of options established when the component trace was turned on.

NONE indicates that options are permitted but none are currently in use.

WRITER

The component trace external writer established when the component trace was turned on or modified.

- *NOT SUPPORTED* indicates that writers cannot be used for this component.

Example 3

To display the status of all component trace external writers, enter:

```
DISPLAY TRACE,WTR=ALL
```

The system responds with:

```
IEE843I 15.18.24 TRACE DISPLAY 447
        SYSTEM STATUS INFORMATION
ST=(ON,0016K,0016K) AS=ON  BR=OFF  EX=ON  MT=ON,024K
WRITER    STATUS     HEAD   COMPONENT   SUBNAME
-----
```

WTRABC	ACTIVE	SYSGRS
		SYSXCF
WTRDEF	ACTIVE	
WTRXYZ	INACTIVE	HEAD SYSSMS

WRITER

The component trace external writer name.

STATUS

The current state of the component trace external writer.

- ACTIVE — writer is active and what components are connected
 - INACTIVE — writer is inactive and what components are connected to it.
- When the component trace external writer is started, tracing will begin.

HEAD

The specified trace is a HEAD trace, there may be SUB traces implicitly connected through the HEAD trace.

COMPONENT

The component(s) connected to the specified component trace external writer. Several components can be connected to the same component trace external writer.

SUBNAME

The specified trace is a SUB trace.

Example 4

To display the status of 2 sublevels of ‘SAMPLE STEP1.STEP2’, enter:

```
D TRACE,COMP=SAMPLE,SUB=(STEP1),N=2,SUBLEVEL
```

Example 5

To display the status of 2 parallel traces of ‘SAMPLE STEPB.STEPC’, enter:

```
D TRACE,COMP=SAMPLE,SUB=(STEPB.STEPC),N=2
```

Example 6

To display the status of the transaction trace in a sysplex, enter:

```
D TRACE,TT
IEE843I 14.47.19 TRACE DISPLAY 709
SYSTEM STATUS INFORMATION
ST=(ON,0064K,00064K) AS=ON BR=OFF EX=ON
MT=(ON,024K)
-----
TRANSACTION TRACE STATUS: ON
BUFSIZ=0002M WRITER=*NONE* LATENT=YES
01: TRAN=TRAN1 USER=TESTERP1
    LVL=0001
02: USER=DONNA* LVL=002
SYSTEMS PARTICIPATING IN TT: SYS1 SYS2 SYS3
```

Displaying Device Status and Allocation

Use the DISPLAY U command to help you keep track of the status (that is, the availability for allocation) of the devices attached to the system. This command lets you request the status of:

- All devices
- A specific device type
- Online devices
- Offline devices

DISPLAY U Command

- Allocation information (that is, jobname and ASID)
- A range of devices
- Automatically switchable devices.

The DISPLAY U command can also display the job names and ASIDs of device users. Knowing the jobs and ASIDs using a particular device can help you determine if you can or cannot take a particular device offline.

```
D {U[,devicetype][,ONLINE ][[,[/]devnum][,nnnnn]]      }
    [,OFFLINE]
    [,ALLOC   ]
    [,AUTOSWITCH|AS][[[[,[/]devnum][,nnnnn][SYS=sysname]]]] (See Note)
    {U,IPLVOL
    {U,VOL=volser
    [,L={a|name|name-a}] }
```

Note: Supply all commas between DISPLAY U and a specified operand. For example, *DISPLAY U,,OFFLINE*.

To compare this DISPLAY U command with the DEVSERV PATHS command, see “Displaying the Status of Devices and Availability of Paths” on page 1-10.

U

The system is to display unit status information about all devices or about specified devices, including non-supported devices (those devices defined in the IODF as DUMMY devices) and channel-to-channel (CTC) links.

devicetype

The device type for which the system is to display unit status information. You can enter any one of the following operands for device type:

CTC — channel-to-channel (CTC) adapters
DASD — direct access storage devices
GRAPHIC — graphic devices
TAPE — magnetic tape units
TP — communication equipment
UR — unit record and dynamic switches
ALL — all above options displayed (equivalent to specifying DISPLAY U)

ONLINE

The system is to display information (in message IEE457I) about only those devices of the specified device type that are online. If you do not specify a device type, the system displays information about all online devices.

OFFLINE

The system is to display information (in message IEE457I) about only those devices of the specified device type that are offline. If you do not specify a device type, the system displays information about all offline devices.

ALLOC

The system is to display allocation information (in message IEE106I) for any specified device that is allocated. Allocation information includes the jobname and address space identifier (ASID) of each job to which the device is currently allocated.

AUTOSWITCH or AS

The system is to display information the coupling facility has about one specific tape device or all tape devices that are defined as automatically switchable.

AUTOSWITCH is valid only for tape devices (that is, for a *devicetype* of TAPE.)

If you omit *devicetype*, the system displays information about all automatically switchable tape devices. If a device is offline to the issuing system, the information specifies that the device is offline ("OFFLINE" in the STATUS column) and does not provide any other information about the device.

SYS=sysname

The system is to display information about the status of automatically switchable devices on the system named *sysname*.

IPLVOL

The system is to display information (in message IEE457I) about the device from which the system was initially loaded (IPL'ed). For additional IPL information use the **DISPLAY IPLINFO** command.

VOL=volser

The volume serial of the device for which the system is to display unit status information in message IEE457I.

Note: If you do not specify ONLINE, OFFLINE, or ALLOC, the system displays status information, without allocation information, about both online and offline devices.

[/]devnum,nnnnn

The system is to display unit status information about devices starting with device number *devnum* for *nnnnn* number of devices. *devnum* is a 3-digit or 4-digit hexadecimal device number, optionally preceded by a slash (/).

If you omit the device number, the system assumes the starting device number is X'000'.

If you omit the number of devices and do not specify ALLOC, the system assumes the number of devices is 16. If you omit the number of devices and specify ALLOC, the system assumes the number of devices is 8.

The system displays status information for primary paths only.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

Notes:

1. If you specify a device number that was not specified in the IODF during system installation, the display starts with the next higher device number that was specified.
2. For multiple-exposure devices (supported pre-MVS/ESA SP 5.2), the value you specify for *devnum* must be the same as that specified in the IODF.
3. For parallel access volumes, if you specify an alias device number, the system ignores it and starts with the next device number that is not an alias device number.
4. When the system displays the device type for an MCS or SMCS console as a 3270 model X, HCD identifies it as a 3270 console.

DISPLAY U Command

5. For autoswitchable devices, a DISPLAY U command without the AUTOSWITCH parameter may display a volume serial number other than for the currently mounted volume if the device is assigned to a different system. Use the AUTOSWITCH parameter to display up-to-date information for such a device.

Example 1

To list the status of the first ten (if any) direct access devices with device numbers of 400 or higher, enter:

```
D U,DASD,,400,10
```

Example 2

To list the users (jobnames and ASIDs) of the first eight allocated devices with device numbers of A250 or higher, enter:

```
D U,,ALLOC,/A250,8
```

or

```
D U,,ALLOC,/A250
```

Example 3

To list the status of a device specified by the volume serial number D72665, enter:

```
D U,VOL=D72655
```

Example 4

To list the status of the automatically switchable tape device with the device number of 270, enter:

```
D U,,AUTOSWITCH,270,1
```

Example 5

To list the status of the first 16 automatically switchable tape devices with device numbers of 000 or higher on the system named GRS127, enter:

```
d U,,AUTOSWITCH,,,SYS=GRS127
```

Displaying Unicode Services

Use the DISPLAY UNI command to show the status of available conversions and whether the conversion services are already initialized (message CUN3000I).

The syntax for the DISPLAY UNI command is:

```
D UNI{,ENVIRONMENT|ENV
      {,SERVICE|SERV
      {,STORAGE|STOR
      {,CONVERSION|CONV
      {,NORMALIZATION|NORM
      {,COLLATION|COLL
      {,FROMID=ccsid
      {,TOID=ccsid
      {,FROMID=ccsid1,TOID=ccsid2}
      {,CASECONV|CASE
      {,STRPROFILE|STRP
      {,ALL|A}
```

The parameters are:

ENVIRONMENT or ENV

Displays three time stamps. The date and time are shown in the format of --/-/---- and --,--,- respectively.

The first time stamp shows when the Unicode environment was created; the second shows when the last change was made with the SET UNI or SETUNI command; the third shows when the image was created. The third time stamp shows a value only if there is an image loaded at IPL time or if an image was added dynamically to an empty Unicode environment. Any subsequent modification to the Unicode environment or dynamic additions of individual tables to an empty environment will result in clearing out of the time stamp field.

If all three time stamps are equal, no active image can be loaded. In that case, check your system log for Unicode error messages during IPL.

If you specify no parameters for the D UNI command, the default is ENVIRONMENT.

SERVICE or SERV

Lists the available conversion services.

STORAGE or STOR

Shows the active and inactive conversion table data space and its limit. The inactive data space pages only appear when an inactive data space exists.

CONVERSION or CONV

Displays the supported character conversion services.

NORMALIZATION or NORM

Displays if normalization service is enabled or disabled. And if the service is enabled, the output includes the normalization versions loaded in the environment.

COLLATION or COLL

Displays if collation service is enabled or disabled. And if the service is enabled, the output includes the collation versions loaded in the environment.

FROMID=ccsid

Displays only the conversions with the specified From-CCSID.

TOID=ccsid

Displays only the conversions with the specified To-CCSID.

DISPLAY UNI Command

FROMID=ccsid1,TOID=ccsid2

Displays only the conversions with the specified From-CCSID and To-CCSID.

CASECONV or CASE

Displays the supported case conversion services.

STRPROFILE or STRP

Displays the string preparation profiles loaded in the environment.

ALL or A

Displays the following information:

- Three time stamps for when the environment was created and last modified, and when the active image was modified
- The names of available services
- The data space pages currently in use and the page limit
- The status of available services

Example 1

To list the time stamps of the Unicode environment, enter:

```
D UNI,ENV
```

If the system has been IPLed with an image, the system responds:

```
SY1 CUN3000I 23.30.23 UNI DISPLAY 698
ENVIRONMENT: CREATED      02/28/2006 AT 19.21.14
              MODIFIED     02/28/2006 AT 19.36.22
              IMAGE CREATED 05/18/2004 AT 13.12.43
```

Otherwise, the system responds:

```
SY1 CUN3000I 23.30.23 UNI DISPLAY 698
ENVIRONMENT: CREATED      02/28/2006 AT 19.21.14
              MODIFIED     02/28/2006 AT 19.36.22
              IMAGE CREATED --/--- AT ---.--
```

Example 2

To list the conversion loaded in the environment, enter:

```
D UNI,CONV
```

The system shows the active conversions:

```
CUN3000I 00.09.01 UNI DISPLAY 584
CONVERSION: 00037-01200(13488)-R
```

If no conversion service is available, the system responds:

```
SY1 CUN3000I 23.51.10 UNI DISPLAY 459
CONVERSION: NO CONVERSIONS FOUND
```

Example 3

To show the environment, storage status and details of available services, enter:

```
D UNI,ALL
```

The system responds:

```
CUN3000I 00.10.40 UNI DISPLAY 587
ENVIRONMENT: CREATED      02/28/2006 AT 23.56.52
              MODIFIED     03/01/2006 AT 00.08.02
              IMAGE CREATED 02/28/2006 AT 23.56.52
```

```

SERVICE: CHARACTER CASE NORMALIZATION COLLATION STRINGPREP BIDI
STORAGE: ACTIVE          140 PAGES
          LIMIT      524287 PAGES
CASECONV: NORMAL SPECIAL
NORMALIZE: DISABLED
NORM VER: NONE
COLLATE: DISABLED
COLL RULES: NONE
STRPROFILES: NONE
CONVERSION: 00037-01200(13488)-R           00037-01200(13488)-L

```

Displaying Virtual Storage Information

Use the DISPLAY VIRTSTOR,HVSHARE command to identify the virtual storage configuration. The following information is displayed in message IAR019I:

- Source of HVSHARE parameter. Can be parmlib member, operator supplied, or default.
- The size of the high virtual shared area in gigabytes, in decimal.
- The range of the high virtual shared area in gigabytes, in decimal.
- The amount of shared storage allocated into memory objects in megabytes, in decimal.

```
D {VIRTSTOR|VS},(HVSHARE)[,L={a|name|name-a}]
```

Note: For the output of the DISPLAY VIRTSTOR,HVSHARE command, see the description of message IAR019I. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books.

Displaying Workload Manager Information

Use the DISPLAY WLM command to display the following information:

- The name of the active service policy for the sysplex, if there is one. If there is no active service policy, the response indicates that the system is running with the DEFAULT service policy.
- The date and time the service policy was activated.
- The date and time the service definition was installed.
- The status information for a specific application environment or for all application environments.
- The status information for a specific scheduling environment on a specific system or on all systems.
- The status information for a resource on a specific system or on all systems.
- The status of the Application Response Measurement (ARM) services as well as the EWLM platform support.

```
D WLM[,SYSTEM=sysname|,SYSTEMS]
      [,APPLENV=applenvname|*]
      [,DYNAPPL=applenvname|*[,SNODE=nodename]
          [,SNAME=subsystemname]
          [,STYPE=subsystemtype]]
      [,SCHENV=schenvname[,SYSTEM=sysname|,SYSTEMS]]
      [,RESOURCE=resource[,SYSTEM=sysname|,SYSTEMS]]
      [,L={a|name|name-a}]
      [,AM[,ALL]]
```

DISPLAY WLM Command

WLM

If no other parameters are specified, displays the name of the active service policy in effect on all systems in the sysplex, as well as the time and date that the service policy was activated. Also, for systems in the sysplex that are in an exception state, it shows summary information. If coupling facility structures are defined for either multisystem enclaves or for LPAR clustering, the status of those is shown.

SYSTEM=*sysname*

When used with D WLM, displays the name of the active service policy in effect on the *sysname* system only, including the time and date that the service policy was activated, when the service definition was installed, and the workload management version level. If coupling facility structures are defined for either multisystem enclaves or for LPAR clustering, the status of those is shown. It also displays the status of the named system in the sysplex (active or otherwise), including the service policy and the workload management mode in effect on the named system.

SYSTEMS

When used with D WLM, displays the name of the active service policy in effect on all systems in the sysplex, the time and date that the service policy was activated, when the service definition was installed, the workload management version level, functionality level, and couple data set format level. If coupling facility structures are defined for either multisystem enclaves or for LPAR clustering, the status of those is shown. Finally, it displays the status of each system in the sysplex (active or otherwise), including the service policy in effect.

If you see a system listed that is not running with the active service policy, either the system does not have connectivity to the WLM couple data set or an attempt to activate the policy on that system failed. If connectivity to the WLM couple data set has been lost (or does not exist), establish the connection. You can use the DISPLAY XCF,COUPLE,TYPE=WLM command to query the status of the WLM couple data set. If connectivity to the WLM couple data set is established and the system is still not running with the active service policy, contact the IBM support center.

When you partition a system out of a sysplex, the system may nevertheless remain listed for a period of time, even though it is no longer part of the sysplex. The reason is that workload management retains knowledge of the system for a certain amount of time in case your installation decides to reactivate the system. If the system is not activated within several days, the system is automatically removed from the list of systems.

APPLENV=*applenvname* or *

Displays status information for the specified application environment (*applenvname*). Specifying APPLENV=* | DYNAPPL=*, the command displays status information for all application environments. The following keywords are valid:

SNODE=*nodename*

When SNODE=*nodename* is used, DISPLAY WLM,DYNAPPL displays only information for dynamic application environments with the specified nodename.

SNAME=*subsystemname*

When SNAME=*subsystemname* is used, DISPLAY WLM,DYNAPPL displays only information for dynamic application environments with the specified subsystemname.

STYPE=subsystemtype

When **STYPE=subsystemtype** is used, DISPLAY WLM,DYNAPPL displays only information for dynamic application environments with the specified subsystemtype.

Note: The DISPLAY WLM,APPLENV command is processed on the coordinator system, which is determined dynamically and may not be the local system where the command was issued. The command output is routed to the console from which the command has been issued, but the command hardcopy is logged on the system on which the command has been processed. Therefore the message may not appear in the SYSLOG on the system where the command was entered.

SCHENV=schenvname

Displays status information for the specified scheduling environment (*schenvname*). You can display multiple scheduling environments by using wildcard characters. The multiple-character symbol (*) and the single-character symbol (?) can be used in any position.

SYSTEM=sysname

Displays the state of the scheduling environment and the availability of each resource referenced by the scheduling environment on the designated system.

SYSTEMS

Displays the state of the scheduling environment on all active systems in the sysplex.

If neither **SYSTEM=sysname** or **SYSTEMS** is specified, **SYSTEMS** is the default.

RESOURCE=resource

Displays status information for the specified resource (*resource*). You can display multiple resources by using wildcard characters. The multiple-character symbol (*) and the single-character symbol (?) can be used in any position.

SYSTEM=sysname

Displays the resource status information on the *sysname* system only.

SYSTEMS

Displays the resource status information on all active systems in the sysplex.

If neither **SYSTEM=sysname** or **SYSTEMS** is specified, the default is the system on which the command is entered.

AM

Displays the current state of the Application Response Measurement (ARM) services and Enterprise Workload Manager (EWLM) platform support together with the EWLM policy name and the number of active processes (address spaces) using ARM services.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

DISPLAY WLM Command

ALL

Displays a list of address spaces that are registered with ARM, together with the registered applications and the started application instances.

Example 1

To display the name of the active service policy, enter:

```
D WLM
```

The system responds with:

```
IWM025I 18.58.12 WLM DISPLAY 316
ACTIVE WORKLOAD MANAGEMENT SERVICE POLICY NAME: VICOM1
ACTIVATED: 1996/10/28 AT: 19:11:40 BY: *BYPASS* FROM: SYS2
DESCRIPTION: Weekday policy with ResGrp
RELATED SERVICE DEFINITION NAME: COEFS
INSTALLED: 1996/10/28 AT: 19:08:42 BY: IBMUSER FROM: SYS2
WLM VERSION LEVEL: LEVEL011
WLM FUNCTIONALITY LEVEL: LEVEL011
WLM CDS FORMAT LEVEL: FORMAT 3
STRUCTURE SYSZLWM_WORKUNIT STATUS: CONNECTED
STRUCTURE SYSZLWM_53052064 STATUS: CONNECTED
```

Example 2

To display status information associated with system WLTEAM9, enter:

```
D WLM,SYSTEM=wlteam9
```

The system responds with:

```
IWM025I 18.57.27 WLM DISPLAY 313
ACTIVE WORKLOAD MANAGEMENT SERVICE POLICY NAME: WEEKDAY
ACTIVATED: 1996/10/28 AT: 19:11:40 BY: USER01 FROM: SYS2
DESCRIPTION: Weekday policy with ResGrp
RELATED SERVICE DEFINITION NAME: COEFS
INSTALLED: 1996/10/28 AT: 19:08:42 BY: IBMUSER FROM: SYS2
WLM VERSION LEVEL: LEVEL011
WLM FUNCTIONALITY LEVEL: LEVEL011
WLM CDS FORMAT LEVEL: FORMAT 3
STRUCTURE SYSZLWM_WORKUNIT STATUS: CONNECTED
STRUCTURE SYSZLWM_53052064 STATUS: CONNECTED

*SYSNAME* *MODE* *POLICY* *WORKLOAD MANAGEMENT STATUS*
WLTEAM9 GOAL WEEKDAY ACTIVE
```

Example 3

To display status information associated with all systems in the sysplex, enter:

```
D WLM,SYSTEMS
```

The system responds with:

```
IWM025I 18.57.27 WLM DISPLAY 313
ACTIVE WORKLOAD MANAGEMENT SERVICE POLICY NAME: WEEKDAY
ACTIVATED: 1996/10/28 AT: 19:11:40 BY: USER01 FROM: SYS2
DESCRIPTION: Default policy with ResGrp
RELATED SERVICE DEFINITION NAME: COEFS
INSTALLED: 1996/10/28 AT: 19:08:42 BY: IBMUSER FROM: SYS2
WLM VERSION LEVEL: LEVEL011
WLM FUNCTIONALITY LEVEL: LEVEL011
WLM CDS FORMAT LEVEL: FORMAT 3
STRUCTURE SYSZLWM_WORKUNIT STATUS: CONNECTED
```

```
STRUCTURE SYSZLWM_53052064 STATUS: CONNECTED
*SYSNAME* *MODE* *POLICY* *WORKLOAD MANAGEMENT STATUS*
WLTEAM9 GOAL WEEKDAY ACTIVE
SYS2 GOAL WEEKDAY ACTIVE
```

Example 4

To display the status of the application environment named DB2PAY, enter:
 D WLM,APPLENV=db2pay

The system responds with:

```
IWM029I 18.59.22 WLM DISPLAY 319
APPLICATION ENVIRONMENT NAME      STATE      STATE DATA
DB2PAY                           AVAILABLE
ATTRIBUTES: PROC=BR14           SUBSYSTEM TYPE: DDF
```

Example 5

To display status of all application environments, enter:
 D WLM,APPLENV=*

The system responds with:

```
IWM029I 19.02.29 WLM DISPLAY 341
APPLICATION ENVIRONMENT NAME      STATE      STATE DATA
BR14                            AVAILABLE
FRESCA                          AVAILABLE
PAYAT2                          AVAILABLE
PAYAT5                          AVAILABLE
PAYROLL                         AVAILABLE
PAYTEST                         AVAILABLE
PAYTEST2                        AVAILABLE
```

Example 6

To display status of the scheduling environment DB2LATE, enter:
 D WLM,SCHENV=DB2LATE

The system responds with:

```
IWM036I 12.21.05 WLM DISPLAY 181
SCHEDULING ENVIRONMENT: DB2LATE
DESCRIPTION:          Offshift DB2 Processing
AVAILABLE ON SYSTEMS:  SYS1  SYS3
```

Example 7

To display status of all scheduling environments in a sysplex beginning with the string 'DB2', enter:
 D WLM,SCHENV=DB2*

The system responds with:

```
IWM036I 12.21.05
WLM DISPLAY 181
SCHEDULING ENVIRONMENT: DB2LATE
DESCRIPTION:          Offshift DB2 Processing
AVAILABLE ON SYSTEMS:  SYS1  SYS3
```

DISPLAY WLM Command

```
SCHEDULING ENVIRONMENT: DB2PRIME
DESCRIPTION:          Primetime DB2 Processing
AVAILABLE ON SYSTEMS: SYS2
```

Example 8

To display system-level status information about the DB2LATE scheduling environment on system SYS1, enter:

```
D WLM,SCHENV=DB2LATE,SYSTEM=SYS1
```

The system responds with:

```
IWM037I 12.21.05 WLM DISPLAY 181
SCHEDULING ENVIRONMENT: DB2LATE
DESCRIPTION:          Offshift DB2 Processing
SYSTEM:              SYS1
STATUS:              AVAILABLE
                     REQUIRED   CURRENT
RESOURCE NAME      STATE      STATE
DB2A               ON        ON
PRIMETIME          OFF       OFF
```

Example 9

To display status of all resources on all systems in a sysplex, enter:

```
D WLM,RESOURCE=*,SYSTEMS
```

The system responds with:

```
IWM038I 12.21.05 WLM DISPLAY 181
RESOURCE:          DB2A
DESCRIPTION:        DB2 Subsystem
SYSTEM  STATE    SYSTEM  STATE    SYSTEM  STATE
SYS1    ON        SYS2    ON        SYS3    ON

RESOURCE:          PRIMETIME
DESCRIPTION:        Peak Business Hours
SYSTEM  STATE    SYSTEM  STATE    SYSTEM  STATE
SYS1    OFF       SYS2    ON        SYS3    OFF
```

Example 10

To display whether the Application Response Measurement (ARM) services and Enterprise Workload Manager (EWLM) platform services are enabled or disabled, enter:

```
D WLM,AM
```

The system responds with:

```
IWM075I 17.54.21 WLM DISPLAY 512
EWLM ARM SERVICES ARE ENABLED
EWLM MANAGED SERVER JOBNAME=EWLMMS ASID=0032
EWLM POLICY NAME=BOSCH DAYTIME POLICY
NUMBER OF REGISTERED PROCESSES=3, APPLICATIONS=1
```

To further display the list of all ARM instrumented address spaces together with the registered applications and started application instances, enter:

```
D WLM,AM,ALL
```

The system responds with:

```
IWM075I 16.32.14 WLM DISPLAY 068
EWLM ARM SERVICES ARE ENABLED
EWLM MANAGED SERVER JOBNAME=EWLMMS ASID=0036
EWLM POLICY NAME=BOSCH POLICY 27JUN2006
NUMBER OF REGISTERED PROCESSES=2, APPLICATIONS=2
ADDRESS SPACES CURRENTLY REGISTERED WITH EWLM ARM:
  JOBNAME=CICSREG1 ASID=0027
    APPLICATION=CICS
      IDENTITY PROPERTIES=0 CONTEXT NAMES=0
      STARTED APPLICATION INSTANCES:
        EWLINST1
          TRAN=0 GROUP=<name omitted>
      REGISTERED TRANSACTIONS:
        SYS_DefaultZWLMTxName
  JOBNAME=BGBOMYT1 ASID=0035
    APPLICATION=My ARM Application
      IDENTITY PROPERTIES=7 CONTEXT NAMES=1
      STARTED APPLICATION INSTANCES:
        <name omitted>
          TRAN=2 GROUP=My-Group
      REGISTERED TRANSACTIONS:
        My-HttpTransaction
  ADDRESS SPACES WITH DISABLED EWLM ARM REGISTRATION:
  JOBNAME=BGBOMYT2 ASID=0034
```

Displaying Cross System Coupling Facility (XCF) Information

Use the DISPLAY XCF command to display cross system coupling information in the sysplex. The syntax of the DISPLAY XCF command is:

DISPLAY XCF Command

```
D XCF[,{PATHIN|PI}
      [, {DEVICE|DEV}={([/]indevnum[,/]indevnum...)|ALL}]
      [, {STRNAME|STRNM}={(strname[,strname]...)|ALL}]
      [, {SYSNAME|SYNSNM}=(sysname[,sysname]...)]
      [, {STATUS|STAT}={([STARTING][,RESTARTING][,WORKING]
                      [,STOPPING][,STOPFAILED][,INOPERATIVE]
                      [,LINKING][,QUIESCING])}]
      [, {PATHOUT|PO}
          [, {DEVICE|DEV}={([/]outdevnum[,/]outdevnum...)|ALL}]
          [, {STRNAME|STRNM}={(strname[,strname]...)|ALL}]
          [, {CLASS}={(classname[,classname]...)|ALL}]
          [, {SYSNAME|SYNSNM}=(sysname[,sysname]...)]
          [, {STATUS|STAT}={([STARTING][,RESTARTING][,WORKING]
                          [,STOPPING][,STOPFAILED][,INOPERATIVE]
                          [,LINKING][,QUIESCING])
                          [,REBUILDING][,QUIESCED])}]
          [, {LOCALMSG|LM}[,{CLASS}={(classname[,classname]...)|ALL}]]
          [, {GROUP|GRP},groupname[,membername|ALL]]
          [, {SYSPLEX|S}[,systemname|ALL]]
          [, {COUPLE|CPL}[,{TYPE}={(name[,name]...)|ALL}]]
          [, {CLASSDEF|CD}
              [, {CLASS}={classname|(classname[,classname]...)|ALL}]
              [, {GROUP|G}=groupname]
          [, {STRUCTURE|STR}
              [, {STRNAME|STRNM}={(strname[,strname]...)|ALL}]
              [, {CONNNAME|CONNM}={(connname[,connname]...)|ALL}]
              [, {STATUS|STAT}={([ALLOCATED][,NOTALLOCATED]
                              [,POLICYCHANGE][,DEALLOCPENDING]
                              [,LARGERCFRMD][,REBUILD][,STRDUMP]
                              [,ALTER][,FPCCONN][,NOCONN][,DUPREBUILD])}]
              [, {CF}[,{CFNAME|CFNM}={(cfname[,cfname]...)|ALL}]]
              [, {POLICY|POL}[,{TYPE}={(name[,name]...)|ALL}]]
              [, {PRSMPPOLICY|PRSMPPOL}]
              [, {ARMSTATUS|ARMS}
                  [, {RESTARTGRP|RG}=rgname]
                  [, {ELEMENT|EL}=elname|{JOBNAME|JOB}=jobname]
                  [, {INITSYS}=initsys]
                  [, {CURRSYS}=currssys]
                  [, {STATE}={([STARTING|START][,{AVAILABLE|AVAIL}][,FAILED]
                               [,RESTARTING|RESTART][,{RECOVERING|RECOVER}])}]
                  [, {DETAIL}]
              [, L={a|name|name-a}]]]
```

XCF

Displays a summary of the current sysplex.

PATHIN or PI

Displays in message IXC355I the device number of one or more inbound signalling paths that XCF can use and information about inbound XCF signalling paths to this system. The display provides information for only

those devices and structures that are defined to the system where this command is executed. The path summary response identifies each inbound path and, if known, the system name and device address of its associated outbound path. If specified without further qualification, summary information about all inbound XCF signalling paths is displayed. Use of the DEVICE or STRNAME keyword requests that more detailed information be displayed.

If there are no inbound paths to this system, the system displays message IXC355I.

Use, but do not repeat, the following keywords in any combination or order:

DEVICE= or DEV={([/]*indevnum*[,[/]*indevnum*]...) or ALL}

Displays in message IXC356I detailed information about the devices requested. If you specify DEVICE=ALL, the system provides information on all inbound signalling paths currently defined to XCF. If you specify only one device number, you do not need to enter the parentheses. A device number consists of 3 or 4 hexadecimal digits, optionally preceded by a slash (/).

If no paths match, the system displays message IXC355I. If there are no inbound paths to this system, the system displays message IXC355I.

STRNAME= or STRNM=(*strname*[,*strname*...]) or ALL

Requests that the system display (message IXC356I) detailed signalling path information for one or more named coupling facility structures. You may specify **ALL** to request information for all coupling facility structures. Wildcard (*) suffixes are allowed.

strname specifies the structure name of a coupling facility structure that is being used as a XCF signalling path for which information is to be displayed. The structure name can be up to 16 characters long. It may contain numeric characters, uppercase alphabetic characters, or the four special characters: \$, @, #, _. It must begin with an uppercase alphabetic character.

XCF signalling structures must begin with the letters IXC.

SYSNAME= or SYSNM=*sysname*(*s*)

Requests that signalling path information be displayed only for paths connected to one or more named systems.

sysname specifies the name of a system that is connected by a XCF signalling path for which information is to be displayed. The system name can be up to 8 characters long. It may contain numeric characters, uppercase alphabetic characters, or the three special characters: \$, @, #. It must begin with a numeric or uppercase alphabetic character.

STATUS= or STAT=*state*(*s*)

Requests that the system display signalling path information for paths or coupling facility structures having at least one of the specified states. *state* specifies the path status of a signalling path or coupling facility structure for which information is requested. You may use any of the following or the indicated abbreviation:

STARTING or START

Validating and initializing hardware.

RESTARTING or RESTART

Making ready (again) for use.

DISPLAY XCF Command

LINKING or LINK

Establishing communication.

WORKING or WORK

Capable of being used.

QUIESCING

Quiescing the use of.

QUIESCED

Use was quiesced.

REBUILDING

In the process of being rebuilt.

STOPPING or STOP

Stopping use, in the process of being removed from service.

STOPFAILED or STOPF

Stop failed, intervention required.

INOPERATIVE or INOP

Defined, but removed from service.

PATHOUT or PO

Displays in message IXC356I the device number of one or more outbound signalling paths that XCF can use and information about outbound XCF signalling paths to this system. The display provides information for only those devices and structures that are defined to the system where this command is executed. The path summary response identifies each outbound path and, if known, the system name and device address of its associated inbound path. If specified without further qualification, summary information about all outbound XCF signalling paths is displayed. Use of the DEVICE, STRNAME or CLASS keyword requests that detail information be displayed.

If there are no outbound paths to this system, the system displays message IXC356I.

DEVICE= or DEV={([/]outdevnum,[/]outdevnum)...} or ALL

Displays in message IXC356I detailed information about the devices requested. If you specify DEVICE=ALL, the system provides information on all outbound signalling paths currently defined to XCF. If you specify only one device number, you do not need to enter the parentheses. A device number consists of 3 or 4 hexadecimal digits, optionally preceded by a slash (/).

If no paths match, the system displays message IXC356I. If there are no outbound paths to this system, the system displays message IXC356I.

CLASS={classname[, classname]...} or ALL

Displays (message IXC356I) detailed information about all devices assigned to the requested transport class. If you specify CLASS=ALL, the system provides information on outbound signalling paths for all transport classes. When you specify a classname ending with an *, then all classes beginning with the specified name are displayed. Do not specify an asterisk for the first character of the classname. If you specify only one class, you do not need to enter the parentheses.

If no classes match, or if there are no outbound paths to this system, the system displays message IXC356I.

STRNAME= or STRNM=*strname(s)*

“See the STRNAME attribute of the PATHIN keyword” on page 4-253.

SYSNAME= or SYSNM=*sysname(s)*

“See the SYSNAME attribute of the PATHIN keyword” on page 4-253.

STATUS= or STAT=*state(s)*

“See the STATUS attribute of the PATHIN keyword” on page 4-253.

LOCALMSG or LM

Displays (message IXC341I) information about the signalling resources that service local message traffic.

If classes are missing, the system displays message IXC345I.

CLASS={(*classname*[, *classname*]...)} or ALL

Displays (message IXC341I) information for a specific transport class. If you do not specify this option, the system displays information for all transport classes. When you specify a classname ending with an *, then all classes beginning with the specified name are displayed. Do not specify an asterisk for the first character of the classname. If you specify only one class, you do not need to enter the parentheses.

GROUP or GRP

Displays information about multisystem groups. If you do not provide a qualifying operand, message IXC331I provides a list of all currently defined XCF groups.

If no groups are defined to the sysplex, the system displays message IXC339I.

groupname

Displays (message IXC332I) the members of the specified group.

If no groups match, the system displays message IXC340I.

membername or ALL

Displays (message IXC333I) detailed information -- the system name, MVS job name, or current status -- about the members of a particular group or all groups. When you specify a membername ending with an *, then all members beginning with the specified name are displayed. Do not specify an asterisk for the first character of the membername.

If no members of a group match, the system displays message IXC326I.

When ALL is specified, detailed information is displayed for all the members of the specified group.

SYSPLEX or S

Displays (message IXC334I) a list of all systems currently participating in the sysplex.

systemname or ALL

Displays (message IXC335I) system status and the last recorded system status monitor time stamp for a system. If ALL is specified for the system name, detailed information for all systems in the sysplex is displayed.

If the system is not defined to the sysplex, the system displays message IXC330I.

DISPLAY XCF Command

COUPLE or CPL

Displays (message IXC357I) information about the couple data set in use by the sysplex. If specified without further qualification, information will be displayed about all couple data sets.

If there is no primary data set defined, the system displays message IXC357I.

TYPE={(name[,name...]) or ALL}

Indicates that information about the couple data sets associated with the named (or ALL) data types is to be displayed.

name specifies the name of the service using the couple data set for which information is to be displayed. The name may be up to eight characters long. It may contain characters A-Z and 0-9 and the characters \$, @, and #. The name must start with a letter. The supported service names are:

- SYSPLEX for sysplex (XCF) types
- ARM for automatic restart management
- CFRM for coupling facility resource management
- SFM for sysplex failure management
- LOGR for the system logger
- WLM for workload management

CLASSDEF or CD

Displays (message IXC343I) the transport classes that are currently defined to XCF on the system upon which the DISPLAY command is executed. If you do not specify either CLASS or GROUP, an alphabetical summary of all transport classes is provided. The CLASS and GROUP operands are mutually exclusive: specify one or the other.

CLASS={(classname[, classname...]) or ALL}

Displays (message IXC344I) detailed information about the requested transport classes. When you specify a classname ending with an *, then all classes beginning with the specified name are displayed. Do not specify an asterisk for the first character of the classname. Specify CLASS=ALL to request detailed information for all transport classes defined to XCF. If you specify only one class, you do not need to enter the parentheses.

If specified transport classes are not defined to XCF, the system displays message IXC345I.

GROUP= or G=groupname

Displays (message IXC344I) detailed information about the transport classes to which the specific group is assigned. You can obtain information for any valid group name, even a group that is not active in the sysplex. Use UNDESIG to list information for those transport classes to which the undesignated groups have been assigned.

If the specified group is not assigned to any transport class, the system displays message IXC346I.

STRUCTURE or STR

Requests information about the coupling facility structures in the policy. If specified without further qualification, summary information (message IXC359I) will be displayed about all coupling facility structures that are in the policy. Using the STRNAME keyword requests the system to display detail information.

Use, but do not repeat, the following keywords in any combination or order:

STRNAME= or STRNM=*strname(s)*

Requests that the system display (message IXC360I) detailed information for one or more named coupling facility structures. You may specify **ALL** to request information for all coupling facility structures. Wildcard (*) suffixes are allowed.

strname specifies the structure name of a coupling facility structure for which information is to be displayed. The structure name can be up to 16 characters long. It may contain numeric characters, uppercase alphabetic characters, or the four special characters: \$, @, #, _. It must begin with an uppercase alphabetic character. Names provided by IBM begin with SYS, an IBM component prefix, or letters A-I.

CONNNAME= or CONNM=*connname(s)*

Requests that the system display detailed information about one or more connectors to a structure. You may specify **ALL** to request information for all connectors to the structure.

STATUS= or STAT=*state (s)*

Requests that the system display only structure information for coupling facility structures having at least one of the specified states. *state* specifies the status of a coupling facility structure for which information is requested and may be any of the following:

ALLOCATED or ALLOC

Coupling facility structure is allocated in a coupling facility.

NOTALLOCATED or NOTALLOC

Coupling facility structure is not allocated in any coupling facility.

POLICYCHANGE or POLCHG

The activation of a coupling facility resource management policy has caused pending policy changes to some coupling facility structures. The changes are pending the deallocation of the structure in a coupling facility.

DEALLOCPENDING or DEALLOC

A coupling facility structure is pending deallocation because of a loss of connectivity to the coupling facility where the structure is allocated or because of a structure dump table being associated with the structure.

LARGERCFRMDS or LARGER

A coupling facility structure has connections that cannot be represented in the coupling facility resource management couple data set.

REBUILD or RB

A coupling facility structure is in the process of being rebuilt or there is a pending structure rebuild for a rebuild to populate a coupling facility (POPULATECF).

STRDUMP or STRD

A structure dump table is associated with a coupling facility structure.

ALTER

A coupling facility structure is in the process of being altered.

FPCONN

A coupling facility structure has at least one failed-persistent

DISPLAY XCF Command

connector. The system displays all connectors to the structure, including those that are not failed-persistent.

NOCONN

A coupling facility structure has no connectors.

DUPREBUILD

A coupling facility structure is in the user-managed duplexing rebuild process.

CF

Requests information about the coupling facility in the policy. If specified without further qualification, the system displays (message IXC362I) summary information about all coupling facilities that are in the policy.

CFNAME= or **CFNM=** {*(cfname[, cfname...])* or **ALL**}

Requests that detailed usage information (message IXC362I) for the named coupling facility be displayed. **ALL** may be specified to request information for all coupling facilities. Generic, or "wildcard", (*) suffixes are allowed.

cfname specifies the name of a coupling facility for which detailed information is to be displayed.

The coupling facility name can be up to 8 characters long. It may contain numeric characters, uppercase alphabetic characters and the special characters \$, @, # and underscore (_). It must begin with an uppercase alphabetic character.

POLICY or POL

Requests information about the policies in use. If specified without further qualification, the system displays (message IXC364I) summary information about all policies that are active.

TYPE={{*(name[, name...])* or **ALL**}

Requests information (message IXC364I) about the policy associated with the named (or ALL) services is to be displayed. The named service is one that uses a couple data set to maintain policy data and supports usage of the SETXCF command to control the policy.

name specifies the name of a service for which policy information is to be displayed. The name may be up to eight characters long. The valid characters are A-Z and 0-9 and the characters \$, @, and #. The name must start with a letter. The supported service names are:

- ARM for automatic restart management
- CFRM for coupling facility resource management
- SFM for sysplex failure management

PRSMPOICY or PRSMPOL

Displays (message IXC349I) the name of the parmlib member that contains the current active XCF PR/SM policy.

If there is no XCF PR/SM parmlib member in use, the system displays message IXC348I.

ARMSTATUS or ARMS

Displays information (through message IXC392I) about active batch jobs and started tasks that are registered as elements of automatic restart management. The element information is grouped by the restart group in which they are defined (based on the current policy). If you specify **ARMSTATUS** without further qualification, summary information about all the active elements will be displayed. You may filter the information by

specifying RESTARTGRP, ELEMENT, JOBNAME, INITSYS, CURRSYS, or STATE. You may specify DETAIL to request more information about the elements. The JOBNAME and ELEMENT parameters are mutually exclusive.

RESTARTGRP= or RG=*rgname*

Information is displayed only for elements in the specified restart group. The *rgname* may contain a wildcard character (*) at the end, to request information for a set of restart groups. If you specify RG=* as the only filter for the display command, only summary information is shown for the restart groups.

ELEMENT= or EL=*elname*

Information is displayed only for the element specified. The *elname* may contain a generic character (*) at the end, to request information for a set of elements.

ELEMENT may not be specified with JOBNAME.

JOBNAME= or JOB=*jobname*

Information is displayed only for elements with the specified job name or started task name.

JOBNAME may not be specified with ELEMENT.

INITSYS= *initsys*

Information is displayed only for elements initially running on the system specified. The *initsys* may contain a generic character (*) at the end, to request information for a set of systems.

CURRSYS= *currsys*

Information is displayed only for elements currently running on the system specified. The *currsys* may contain a generic character (*) at the end, to request information for a set of systems.

STATE= *state*

Information is displayed only for elements in the states specified.

STARTING or START

The element has initially registered,(has issued the IXCARM macro with the REQUEST=REGISTER parameter) but has not yet indicated it is ready to accept work (has not issued the IXCARM macro with the REQUEST=READY parameter).

AVAILABLE or AVAIL

The element has indicated it is ready to accept work (has issued the IXCARM macro with the REQUEST=READY parameter) or the element exceeded the ready timeout threshold before it issued the IXCARM macro with the REQUEST=READY parameter.

FAILED

The element has ended and a restart has not been initiated by MVS, yet. This condition should apply only for a short amount of time if automatic restart management restarts have been enabled. (This state is not related to the failed state for an XCF member.)

RESTARTING or RESTART

MVS has initiated a restart of this element, but it has not re-registered with automatic restart management (has not issued the IXCARM macro with the REQUEST=REGISTER parameter), yet.

DISPLAY XCF Command

RECOVERING or RECOVER

The element has been restarted and has re-registered with automatic restart management (has issued the IXCARM macro with the REQUEST=REGISTER parameter), but has not indicated that it is ready to accept work (has not issued the IXCARM macro with the REQUEST=READY parameter), yet.

DETAIL

Detailed information is displayed. The scope of this information depends on the other parameters specified.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to appear.

If you omit this operand, the display is presented in the first available display area or the message area of the console through which you enter the command.

DUMP Command

The DUMP command requests a system dump (SVC dump) of virtual storage. The SVC dump is stored in a direct access data set. The data set may be either a pre-allocated dump data set named SYS1.DUMPxx, or an automatically allocated dump data set named according to an installation-specified pattern.

You should request only one dump at a time on one system. Otherwise, you might have trouble determining the dump request that causes a particular IEE094D message. Also, a system writes only one SVC dump at a time, so it does not save anything to make several requests at once.

Hiperspace information is not included in SVC dumps. For more information about hiperspace data in dumps, see *z/OS MVS Diagnosis: Tools and Service Aids*.

Wildcards

You can use wildcards to identify multiple names. On a reply for a DUMP command, you can specify wildcards in job names, data space names, user IDs, XCF group names, and XCF member names. The parameter descriptions tell you when you can use wildcards. The wildcards are:

Wildcard	Meaning
*	Zero or more characters, up to the maximum length of the string. An * can start the string, end it, appear in the middle, or appear in several places in the string. A single * for the name indicates that all job names, data space names, user IDs, XCF group names, or XCF member names will match.
?	One character. One or more ? can start the string, end it, appear in the middle, or appear in several places in the string. A single ? indicates all names consisting of one character.

Note: You can mix wildcards in any combination.

Examples are:

- *A* specifies all names that contain an A, including the name A.
- *A*B specifies all names that contain an A and ending with a B, with or without any intervening characters.
- ?A? specifies all 3-character names with an A as the second character.
- ?A?B specifies all 4-character names with A as the second character and B as the fourth character.
- ?A* specifies all names of 2 or more characters whose second character is A.

Syntax

The complete syntax for the DUMP command is:

```
DUMP {COMM={{(title)}} [,PARMLIB=xx]}           }
      {'title'} | [,PARMLIB=(xx[,xx]...)]
      {"title"} [SYMDEF=(symdef[,symdef]...)]
{TITLE={{(title)}}}                                }
      {'title'}
      {"title"}
```

DUMP Command

Parameters

COMM={(title)}
{'title'}
{"title"}
TITLE={(title)}
{'title'}
{"title"}

The title (1-100 characters) you want the dump to have. This title becomes the first record in the dump data set. COMM and TITLE are synonyms.

Note: The system determines the title of a dump as follows:

- A title specified in the DUMP command (for example, DUMP TITLE="DUMP Specified via WTOR") takes precedence over a title specified within a parmlib member.
- When you do not specify a title in the DUMP command, the title specified within a parmlib member takes precedence.
- If you specify titles in multiple parmlib members, the title in the first parmlib member takes precedence. For example, if all of the parmlib members in **PARMLIB = (RA,XC,CF)** are titled, the dump title is the one specified in the **RA** parmlib member.
- With no title specified in the DUMP command or parmlib members, the title becomes DUMP FOR PARMLIB=(xx,yy,zz), where xx,yy,zz are the parmlib members.

PARMLIB=xx | PARMLIB=(xx[,xx]...)]

If the installation has set up DUMP command parmlib members (IEADMCxx parmlib members), you can avoid having to reply with many dump options by instead specifying the suffixes of one or more IEADMCxx parmlib members. When you do this, the system concatenates the contents of the IEADMCxx parmlib members together. If each parmlib member contains a title, the first title in the sequence is used. If specified on the DUMP command, the TITLE= parameter supercedes the titles in the parmlib members.

For example, if parmlib member IEADMCTC represents dump options for a base TCP/IP configuration, and IEADMCTA contains additional dump options to append to the base settings, issuing the command DUMP PARMLIB=(TC,TA) will get you the base options plus the additional options.

If you do not specify the PARMLIB= parameter, the system prompts you with the following message for the dump options you want to specify:

* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

For information on setting up IEADMCxx parmlib members, and on using the IBM-supplied sample IEADMCxx parmlib members, see *z/OS MVS Diagnosis: Tools and Service Aids*.

SYMDEF=(symdef[,symdef]...)

You may specify one or more symbol definitions for use within dump command parmlib members by specifying the SYMDEF= keyword. *symdef* should take the form *&symbol.=value*. The value passed within the *&symbol.=value* parameter will be parsed as if specified within an IEASYMxx member; symbolic substringing, defining a symbol using a previously defined symbol, and character conventions are recognized, including upper and lower case. For example:

DUMP PARMLIB=xx,SYMDEF=(&PAGING1.='AQFT',&CICS.='CICS1')

Notes:

1. You can override existing symbols by specifying a double ampersand. If you do not use the double ampersand when specifying an existing symbol, it is ignored and a warning message is issued.
 2. When specifying the substitution text, you need to observe the same rules as you define it within the IEASYMxx members. For example, the length of the resolved substitution text cannot exceed the length of &symbol, including the ampersand on &symbol and excluding the single quotation marks on 'value'. In addition, all specified symbols must be unique. Specifying the same symbol for multiple times will result in unexpected behavior.
- For more information about the rules of defining symbols, see the IEASYMxx parmlib member in *z/OS MVS Initialization and Tuning Reference*.

The syntax of a DUMP command specified within the IEADMCxx members of Parmlib is identical to that specified on the DUMP command through writes to operator with reply (WTORs).

Specifying Dump Options

You must use the REPLY command to respond to message IEE094D.

The REPLY command syntax for specifying dump options is:

```
R id,U
      or
R id[,ASID=(n[,n]...)] [,JOBNAME=(name[,name]...)] [,TSONAME=(name[,name]...)]
      [,DSPNAME=(dspname-entry[,dspname-entry]...)]
      [,PROBDESC|PROB|PD=key-spec] [,REMOTE=(request[,request]...)]
      [,SDATA[=(option[,option]...)] [,STOR=(beg,end[,beg,end]...)]]
      [,STRLIST=(s-option[,s-option]...)]
      [,CONT],END]
```

Notes:

1. When you specify CONT or END, it must be the last parameter on the input line.
2. The CONT keyword does not work within a SYSP= list.
3. When you specify U, it must be the first parameter following the identification number.

Where **request** represents:

```
{GRPLIST={group(member)
           {(group(member[,member]...) [,group(member[,member]...)]...) }}
```

```
{SYSLIST={sysinfo|(sysinfo[,sysinfo]...)} }
           [,DSPNAME,DSPNAME=(dspname-entry[,dspname-entry]...)]
           [,SDATA|,SDATA=(option[,option]...)]
           [,STOR|,STOR=(beg,end[,beg,end]...)] }
```

DUMP Command

Where **s-option** represents:

```
STRNAME=strname
[,CONNAME=connname]
[,ACCESSTIME={ENFORCE|NOLIMIT|NOLIM}]
[,LOCKENTRIES]
[,USERCNTLS]
[,EVENTQS]
[,({EMCONTROLS={ALL|(list)})}
[,({COCLASS|STGCLASS|LISTNUM}={ALL|(list)}
{[,ADJUNCT={CAPTURE|DIRECTIO}] [,ENTRYDATA={UNSERIALIZE|SERIALIZE}]}
{[,SUMMARY]})]
```

- id** The identification number (0-99), as specified in system message IEE094D. The leading zero can be omitted.
- U** The dump is to be of the master scheduler address space and include the storage areas defined by the SDATA default options. No other parameters are allowed with this parameter.

ASID=(n[n]...)

n is the hexadecimal address space identifier of an address space you want to dump. If you specify only one identifier, you do not need to enter the parentheses. The maximum number of address space identifiers (ASIDs) that can be specified is 15. Please refer to the limitation section below.

Address space limitation applying to ASID, JOBNAMES, TSONAME, and DSPNAME parameters. A dump request can only handle a maximum of 15 unique ASIDs. The following items are used in the specified order to determine which ASIDs are added:

- MASTER ASID is added first when TYPE=XMEM or TYPE=XMEME is in force (in this case, the operator dump command can only accept 14 other ASIDs as input)
- The address spaces directly requested in the ASID parameter
- The address spaces associated with the jobs named in the JOBNAMES parameter
- The address spaces associated with the user IDs in the TSONAME parameter
- The address spaces for the data spaces in the DSPNAME parameter

Each ASID added due to an earlier item reduces the available slots for the next item. Also wildcards used in the JOBNAMES, TSONAME, and DSPNAME parameters can result in multiple address spaces being requested.

When the REPLY results in more ASIDs being requested than processing can handle, the system issues message ASA104I. If TYPE=XMEM or TYPE=XMEME is in force, then only the first 14 unique ASIDs will be allowed. Otherwise, 15 ASIDs are allowed. Regardless, the dump is taken with the truncated list of ASIDs as shown on the resultant IEA911E or IEA611I message.

JOBNAME=(*name[,name]...*)

name identifies a background job whose address space you want to dump. If you specify only one name, you do not need to enter the parentheses. You can specify a maximum of 15 job names. See the ASID parameter for the actual limit on address spaces that can be specified.

You can specify each name explicitly or with wildcards. See "Wildcards" on page 4-261.

TSONAME=(*name[,name]...*)

name is the name of any address space you want to dump, including the user identifier (ID) of a TSO user. If you specify only one name, you do not need to enter the parentheses. You can specify a maximum of 15 names. See the ASID parameter for the actual limit on address spaces that can be specified.

You can specify the name explicitly or with wildcards. See "Wildcards" on page 4-261.

Notes:

1. If you do not specify ASID, JOBNAME, or TSONAME, the master scheduler address space is dumped.
2. Dumping several large address spaces such as the VTAM, master scheduler, and job entry subsystem address spaces or large system areas such as the CSA and SQA degrades performance of the system significantly. Dump only the address spaces that are likely to be involved in a problem. For example, if a TSO terminal in OPERATOR mode is having a problem, dump only the TSO user's address space and the TCAM or VTAM address space.

DSPNAME=(*dspname-entry[,dspname-entry]...*)

Specifies the data spaces to be dumped. The form of a *dspname-entry* is:

asid.name
'jobname'.name.

You can mix the two forms.

asid

is the explicit hexadecimal address space identifier of the owner of the data space you want to dump.

Note: When a data space is owned by an address space not included in the dump, the ASID of that address space is added to the dump. While up to 15 unique ASIDs and/or JOBNAMEs may be specified for the limit of 256 address spaces, refer to the ASID parameter of the DUMP command for the actual limit on the number of unique ASIDs that can be specified for DSPNAME. The address space limitation may prevent specifying the maximum number of data spaces.

'jobname'

is the name of the job associated with the data space you want to dump. The *jobname* must be enclosed in single quotes; see the CONT parameter later in this chapter for special handling details. You can specify *jobname* explicitly or with wildcards. See "Wildcards" on page 4-261.

name

is the 1 to 8 character name associated with the data space at its creation. You can specify the data space name explicitly or with wildcards. See "Wildcards" on page 4-261.

DUMP Command

If you specify to dump only one data space, represented by *dspname-entry*, you do not need to enter the parentheses. You can dump a maximum of 256 data spaces. If you enter a larger number, the system will dump only 256 data spaces.

PROBDESC or PROB or PD = key-spec

Provides problem information that is passed to any SVC dump, but is intended for dumps requested by the REMOTE parameter. When a system requests a dump on another system in the sysplex, the system being dumped calls an IEASDUMP.QUERY routine. The routine uses the information to determine if its system should be dumped and, if so, what storage areas should be added to the dump; the IEASDUMP.QUERY routine suppresses the requested dump only if PROBDESC specifies SYSDCOND.

key-spec is either of the following:

key
(*key-value*[,*key-value*]...)

key-value is either of the following:

key
(*key*,*data*)

You can mix the two forms. You can specify a maximum of 16 *key-value* forms.

key

Is a 1- to 8-character value that corresponds to the SDPD_KLD_KEY field in the IHASDPD mapping macro. The key must not begin with A through I or SYS; these are reserved for IBM use. IBM-supplied values for *key* are:

- SYSDCOND: Suppresses a dump on another system in a sysplex if the other system does not have an IEASDUMP.QUERY routine or if no IEASDUMP.QUERY routine returns a code of 0.
- SYSDLOCL: Requests the following:
 - Dumps of other systems in a sysplex.
 - An immediate dump of the local system, on which you are entering the DUMP command.
 - A second, deferred dump of the local system, if a SYSLIST or GRPLIST option of the REMOTE parameter includes the local system. The deferred dump contains areas added by IEASDUMP.QUERY, IEASDUMP.GLOBAL, and IEASDUMP.LOCAL exit routines, if any routines had been associated with those exits.
 - SYSDNGRP: Causes the IEASDUMP.QUERY routine to receive control without the implicit address spaces specified with the SYSLIST or GRPLIST option of the REMOTE parameter. This allows the IEASDUMP.QUERY routine to control which address spaces should be dumped.

No data is specified with the IBM-supplied keys.

data

Is 1 to 16 characters of information to be used by the IEASDUMP.QUERY routine. If a *data* value is not provided, the system passes 16 blanks to the IEASDUMP.QUERY routine. If the value is shorter than 16 characters, the system pads it on the right with blanks to the length of 16.

For dumps on other systems in a sysplex initiated by the DUMP command, the other systems will not invoke IEASDUMP.QUERY routines unless the DUMP command contains a PROBDESC parameter.

See *z/OS MVS Data Areas, Vol 2 (DCCB-ITZYRETC)* for the IHASDPD data area. See *z/OS MVS Programming: Authorized Assembler Services Guide* for the IEASDUMP.QUERY routine.

REMOTE=(request[,request]...)

Specifies a dump on one or more systems in the sysplex. Each request consists of a GRPLIST or SYSLIST option to identify the system or systems, optionally followed by DSPNAME, SDATA, and STOR options to specify attributes for the requested dump or dumps. GRPLIST and SYSLIST options can appear more than once in the REMOTE parameter; each DSPNAME, SDATA, or STOR option applies to the preceding GRPLIST or SYSLIST option.

If the reply specifies a key of SYSDLOCL in the PROBDESC parameter and the GRPLIST or SYSLIST option includes the *local system* on which you are entering the DUMP command, the local system is dumped twice: the immediate dump is for the DUMP command and the deferred dump is for the REMOTE parameter. If the reply does not specify a key of SYSDLOCL, the REMOTE parameter does not apply to the local system; only the immediate dump is written.

Note: A dump requested through the REMOTE parameter may not be written. The reasons for dump suppression are listed in *z/OS MVS Diagnosis: Tools and Service Aids*.

If the GRPLIST and SYSLIST options specify the same system more than once, only one dump is written combining all of the options.

If the reply specifies REMOTE and the other parameters do not indicate the areas to be dumped on the local system, the immediate dump of the local system is of the master scheduler address space.

The *request* subparameters and values follow:

GRPLIST=group(member)

GRPLIST=(group (member[,member]...)[, group(member[,member]...)])...

Specifies one or more systems by the XCF group and member names.

You can specify the *group* and *member* explicitly or with wildcards. See "Wildcards" on page 4-261.

SYSLIST=sysinfo

SYSLIST=(sysinfo[,sysinfo]...)

Specifies one or more systems and, optionally, address spaces and jobnames to be dumped on those systems. The *sysinfo* consists of:

sysname

Specifies the name of the system to be dumped.

If SYSLIST does not specify any address spaces or job names, the dumping services address space (DUMPSRV) is dumped.

sysname(space-id[,space-id]...)

Specifies the name of the system and its address spaces or jobs to be dumped. *space-id* is of the form:

(asid)

Specifies the identifier of the address space.

('jobname')

Specifies the name of the job to be dumped.

DUMP Command

The asids and job names can be in any order; each job name must be enclosed in single quotes.

You can specify the *sysname* and *jobname* explicitly or with wildcards. See “Wildcards” on page 4-261.

See the ASID parameter for the limit on address spaces that can be specified for each of the dumps.

DSPNAME

Specifies for the dumps on other systems the same data spaces specified for the local SVC dump.

DSPNAME=(*dspace-entry*[,*dspace-entry*]...)

Specifies the data spaces to be dumped on the other systems. See the “DSPNAME parameter” on page 4-265 for the DSPNAME values.

Note: If DSPNAME is not specified for the other systems, data spaces are not dumped.

SDATA

Specifies for the dumps on other systems the same SDATA options specified or defaulted for the local SVC dump.

Note: The SDATA options for dumps on other systems are not affected by CHNGDUMP changes in effect for local SVC dumps.

SDATA=(*option*[,*option*]...)

Specifies the specific storage areas you want to dump on the other systems. See the “SDATA parameter” for the valid options and their definitions.

Note: If SDATA is not specified in the REMOTE parameter, each system uses the SDATA options that apply to any SVC dump on that system. The contents of a dump on each system are affected by CHNGDUMP commands previously entered on the system.

STOR

Specifies for the dumps on other systems the same ranges of virtual storage specified for the local SVC dump.

STOR=(*beg,end*[,*beg,end*]...)

Specifies the ranges of virtual storage you want to dump on the other systems. See the “STOR parameter” on page 4-269 for the STOR values.

Note: If STOR is not specified for the other systems, ranges of virtual storage are not dumped.

SDATA=(*option*[,*option*]...)

Specifies the specific storage areas you want to dump. The valid options and their definitions are:

ALLNUC All of the DAT-on nucleus, including page-protected areas, and all of the DAT-off nucleus.

COUPLE XCF related information in the sysplex.

CSA Common service area.

GRSQ Global resource serialization (ENQ/DEQ/RESERVE) queues.

Note that the GRS information collection is related to the GRSCNFxx GRSQ(xx) option setting.

LPA	Link pack area modules for the dumping task.
LSQA	Local system queue area.
NUC	Non-page-protected areas of the DAT-on nucleus.
PSA	Prefixed storage area for all processors. (Equivalent to the ALLPSA option on the SDUMPC macro.)
NOPSA	No prefixed storage area. (Equivalent to the NOALLPSA option on the SDUMPC macro.)
RGN	Private area of address space being dumped, including LSQA and SWA.
SERVERS	Requests that the registered IEASDUMP.SERVER dynamic exits receive control.
SQA	System queue area.
NOSQA	No system queue area.
SUM	Summary dump.
NOSUM	No summary dump.
SWA	Scheduler work area.
TRT	GTF, system trace, master trace, and NIP hard-copy buffer data.
WLM	Workload management related data areas and storage.
XESDATA	coupling facility-related information.

If you do not specify SDATA, or if you specify SDATA with no options, the system uses these SDUMPX macro options: ALLPSA, SQA, SUM, and IO. Also, SDATA=SERVERS is always used for operator dumps.

STOR=(beg,end[,beg,end]...)

Specifies the ranges of virtual storage you want to dump. You can specify the beginning and ending addresses of each range as 4-byte hexadecimal numbers, such as 010BA040, or 7-digit decimal numbers followed by a K, such as 0050860K.

STRLIST= or STL=(STRNAME=*strname*...)

Used to include in the dump a list of coupling facility structures. Cache and list structures can be dumped; lock structures cannot be dumped. Following are the structure-related keywords:

STRNAME= or STRNM=*strname*

Designates a particular coupling facility list or cache structure. *strname* is the name of the coupling facility structure to be included in the dump. Any dump options for this structure are replaced when you issue this command. If *strname* does not begin with a letter or is longer than 16 characters the system issues syntax error message IEE866I. You may include more than one STRNAME=*strname* within the parentheses, separated by commas.

CONNNAME= or CONNM=*connname*

When specified for a coupling facility cache structure, requests the user registry information for this user be included in the dump. *connname* is the name of a connected user. If the connected user represented by the *connname* does not exist, the dump will not contain user registry information.

DUMP Command

ACCESSTIME= or ACC={ENFORCE or ENF or NOLIMIT or NOLIM}

Indicates whether the dump time limit specified on the ACESSTIME parameter of the IXLCNN macro is in effect. When ACESSTIME=ENFORCE is specified, the system holds structure dump serialization no longer than the time interval specified on the IXLCNN macro. This is the default. If ACESSTIME=0 is specified on the IXLCNN macro and ACESSTIME=ENFORCE is specified on the dump request, the structure will not be included in the dump.

When ACESSTIME=NOLIMIT is specified, the dump time limit is not in effect and the system will hold structure dump serialization until processing is completed.

LOCKENTRIES or Locke

When specified for a coupling facility list structure, the system includes in the dump the lock table entries for the requested structure. Since lock table entries do not exist for coupling facility cache structures, this keyword is ignored when specified for a coupling facility cache structure.

USERCNTLS or UC

Requests that the user attach controls be included in the dump.

EVENTQTS or EQS

Requests that the event queues be included in the dump.

(list)

Represents a list of values, ranges of values, or values and ranges of values.

(start1-end1,value2,start3-end3, ...)

EMCONTROLS= or EMC=ALL or (list)

Specifies which event monitor controls are included in the dump.

EMCONTROLS is valid only for a coupling facility list structure. If specified for a coupling facility cache structure, the structure is not included in the dump.

When EMCONTROLS=ALL is specified, the event monitor controls associated with all lists in the structure are dumped.

When EMCONTROLS=(list) is specified, the event monitor controls associated with the specified list number are included in the dump. The values specified for (list) are the decimal list values, 0 – 4294967295. When event monitor controls for a requested list do not exist, they are not dumped and no error results.

COCLASS= or COC=ALL or (list)

Specifies which cast-out classes are included in the dump. For each cast-out class, the cast-out class controls are dumped and the directory information for each of the entries within the requested cast-out classes are dumped (if SUMMARY is not specified).

COCLASS is valid only for a coupling facility cache structure. If specified for a coupling facility list structure, the structure is not included in the dump.

When COCLASS=ALL is specified, the cast-out class controls for all cast-out classes are dumped along with the directory information for all entries within the classes (if SUMMARY is not specified).

When COCLASS=(list) is specified, the cast-out class controls for (list) are dumped along with the directory information for the entries in the requested cast-out classes (if SUMMARY is not specified). The values specified in a

range are the decimal cast-out class values in the range 0 – 65535. When a requested class does not exist, it is not dumped.

STGCLASS= or SC=ALL or (list)

Specifies which storage classes are included in the dump. For each storage class, the storage class controls are dumped and the directory information for each of the entries within the requested storage classes are dumped (if SUMMARY was not specified).

STGCLASS is valid only for a coupling facility cache structure. If specified for a coupling facility list structure, the structure will not be included in the dump.

When STGCLASS=ALL is specified, the storage class controls for all storage classes are dumped along with the directory information for all entries within the classes (if SUMMARY is not specified).

When STGCLASS=(list) is specified, the storage class controls for (list) are dumped along with the directory information for the entries in the requested storage classes (if SUMMARY is not specified). The values specified are the decimal storage class values, 0 – 255. When a requested class does not exist, it is not dumped.

LISTNUM= or LNUM=ALL or (list)

Specifies which lists are included in the dump. The list controls are dumped along with the entry controls for the entries on each requested list (if SUMMARY is not specified).

LISTNUM is only valid for a coupling facility list structure. If specified for a coupling facility cache structure, the structure is not included in the dump.

When LISTNUM=ALL is specified, the list controls for all lists in the coupling facility list structure are dumped along with the entry controls (if SUMMARY is not specified).

When LISTNUM=(list) is specified, the list controls for (list) are included in the dump along with the entry controls for those lists. The values specified are the decimal list values, 0 – 4294967295. The system ignores a zero in the case of LISTNUM. No error results. When a requested list does not exist, it is not dumped.

You may use the following keyword to further modify the STGCLASS, COCLASS and LISTNUM keywords:

ADJUNCT= or ADJ={CAPTURE or CAP or DIRECTIO or DIO}

Indicates that the adjunct data for each entry specified by the range is included in the dump. When this keyword is not specified or when adjunct data does not exist for this structure, the adjunct data is not included in the dump.

ADJUNCT may not be specified with SUMMARY. If they are both specified, a syntax error is issued.

When ADJUNCT=CAPTURE is specified, the adjunct data is captured in the facility dump space along with the directory information while dumping serialization is held.

When ADJUNCT=DIRECTIO is specified, the adjunct data is written directly to the dump data set after the directory information is captured. The adjunct data is not captured in the structure dump table. The adjunct data may be changing as dumping proceeds.

DUMP Command

ENTRYDATA= or EDATA={UNSERIALIZE or UNSER or SERIALIZE or SER}

Indicates that the entry data for each entry within the requested range is included in the dump. When this keyword is not specified or when entry data does not exist for the structure, entry data is not included in the dump.

ENTRYDATA may not be specified with SUMMARY. If they are both specified, a syntax error is issued.

When ENTRYDATA=UNSERIALIZE is specified, the entry data is dumped after structure dump serialization is released. The entry data may be changing relative to the entry controls that were captured while structure dump serialization was held.

When ENTRYDATA=SERIALIZE is specified, the entry data is dumped while serialization is held. If ACESSTIME=ENFORCE is specified and the dump time limit expires before the entry data is written to the dump data set, the system continues to write the entry data to the dump data set even though serialization is not held.

SUMMARY or SUM

Indicates that a summary of the range of classes or lists is dumped. The directory information for the entries is excluded from the dump.

SUMMARY may not be specified with ADJUNCT or ENTRYDATA. If it is specified with either of these keywords, a syntax error is issued.

Notes:

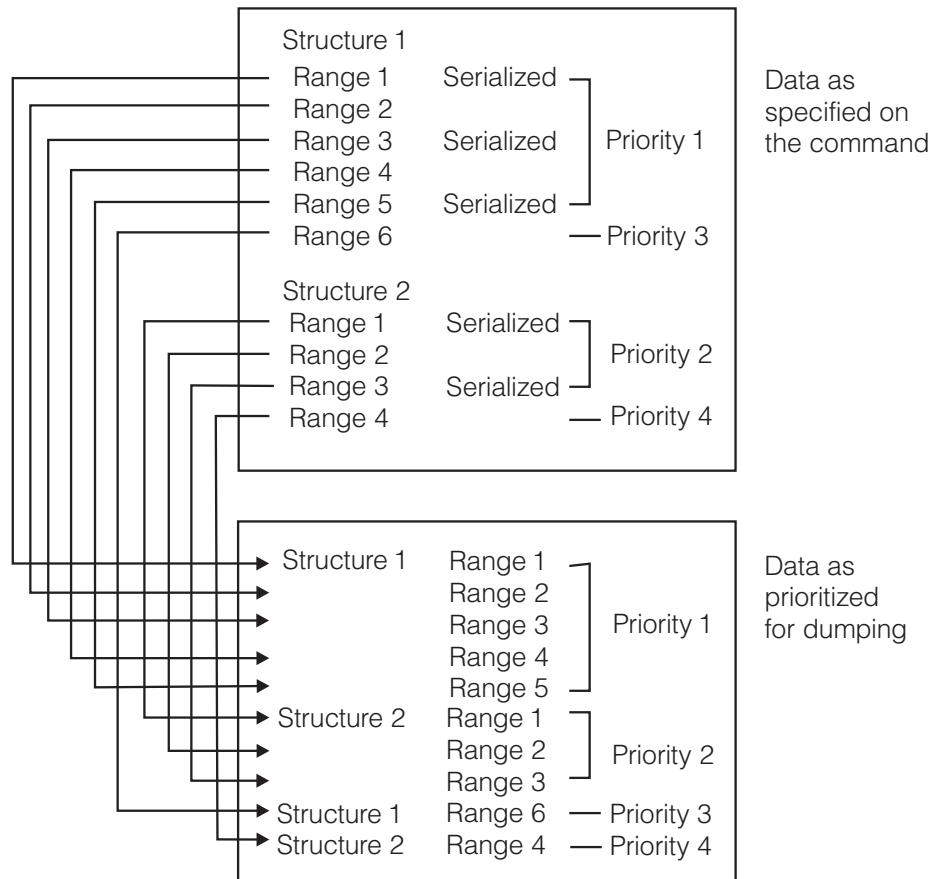
1. A syntax error is issued if STRNAME is not the first keyword.
2. If CONNAME and ACESSTIME are specified more than one time for a structure, the first CONNAME and the last ACESSTIME are used.
3. When a list number, event monitor controls, a storage class, a cast-out class, or an entry is specified in the STRLIST more than once, it will be dumped more than once. An example of this is when STGCLASS=ALL is specified with COCLASS=ALL. All entries in the coupling facility cache structure are dumped twice. Once grouped by storage class and again grouped by cast-out class.
4. When neither LISTNUM, STGCLASS, EMCONTROLS, or COCLASS is specified, no list or class controls are dumped and no entries are dumped.
5. If a large amount of data is requested to be dumped, the system may not be able to completely dump all the data. You can expect to successfully dump up to a maximum of 47 structures if you specify no more than a total of six ranges. If you must specify more than six ranges, you must specify fewer structures. For each structure less than 47 that you specify, you can specify another 10 ranges, as follows:

Number of Structures	Number of Ranges
47	6
46	16
45	26
44	36
:	:

If the system cannot dump all the data you requested, it prioritizes the data according to your specifications on the command in the following manner:

- a. The system will attempt to dump the first requested structure first.

- 1) Within that structure, the system processes the LOCKENTRIES, EVENTQS, EMCONTROLS, USERCNTLS, COCLASS, STGCLASS, and LISTNUM parameters in the order that they are specified. COCLASS, STGCLASS, EMCONTROLS, and LISTNUM may be specified more than once for a single structure.
- 2) The system dumps requested serialized data before requested unserialized data starting with the first requested data in the structure and proceeding through the last data that was requested as *serialized*.
 - b. The system then dumps the next-requested structure data starting with the first requested data in the structure and proceeding through the last data that was requested as *serialized*.
 - c. The system continues in this manner until all *serialized* data in all requested structures has been prioritized for dumping.
 - d. The system then dumps any remaining data that was requested as *unserialized* that may not have been dumped beginning with the first-requested structure.



6. The CONT parameter allows the operator to provide input to the CHNGDUMP command that spans more than one line of input. You can specify the CONT parameter after any comma within the STRLIST parameter list. If a line of input ends with a comma and any closing parentheses are missing, the system assumes the CONT parameter.

CONT

Specifies that you want to continue the reply on another line. The system

DUMP Command

reissues message IEE094D in response to this parameter, after which you can continue your reply. All parameters can appear on a continuation line.

On a continuation line, you can continue values for any parameter. In a parenthesized expression in the parameters, as you reach the end of a line, add the comma after a value and press ENTER, without typing CONT. The system will issue message IEE094D. In response, continue with the next value in the expression. For example:

```
R 17,JOBNAME=(PQRJOB07,QRSJOB08),REMOTE=(SYSLIST=($1)),JOBNAME=(PQRJOB07,  
* 18 IEE094D ...  
R 18,QRSJOB08),END
```

If you reach the end of a line with a keyword, a syntax error occurs; for example:

```
JOBNAME=
```

If you reach the end of a line with a value that is not in parentheses, the system considers the reply ended; for example:

```
JOBNAME=PQRJOB07
```

If a reply **begins** with a single quote, double all single quotes in the line and enclose the line with quotes. For example, a *jobname* in the DSPNAME parameter must be enclosed in quotes. If the first line is:

```
R 1,DSPNAME=('job1'.dsp1,
```

Then the second line is:

```
R 2,'''job2''.dsp2, ...)'
```

To avoid this problem, do **not** end the R 1 line with a comma, but instead begin the R 2 line with the comma:

```
R 1,DSPNAME=('job1'.dsp1  
R 2,, 'job2'.dsp2, ...)
```

END

Identifies the end of your reply. You need to specify END only when the reply line contains no other parameters. If you do not specify any parameters in the reply other than CONT and END, the system dumps the master address space with the SDATA default options.

Example 1

To dump the virtual address space for the job named PAYROLL, including the private area, the non-page-protected areas of the DAT-on nucleus, the GTF, system trace, and master trace data, the contents of storage locations 010CD450 to 010FF400, and the contents of storage locations 0000012K to 0000060K, you can enter:

```
DUMP COMM=(DUMP FOR PAYROLL)
```

In response to this command, the system issues:

```
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
```

Reply as follows:

```
R id,JOBNAME=PAYROLL,SDATA=(NUC,RGN,TRT),STOR=(010CD450,010FF400,0000012K,0000060K)
```

Because you did not specify CONT at the end of this reply, the system considers your dump request complete.

Example 2

To dump the private storage for ASIDs 6, 1, 2, 3, B, and C, so you can solve a problem loop in ASID 6, you can enter:

DUMP COMM=(LOOP IN ASID 6)

In response to this command, the system issues:

* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

If you are sure of the address space you want to dump, but are not yet sure of the storage areas you need to find the problem, you can enter:

R id,ASID=(6,1,2,3,B,C),CONT

Because you specified CONT, the system reissues:

* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

Now, if you decide you need to see only the private areas (including the LSQA and SWA) for the address spaces you are dumping, you can enter:

R id,SDATA=(RGN)

Because you did not specify CONT at the end of this reply, the system considers your dump request complete.

Example 3

To dump the default storage areas of the TSO address spaces TERMINAL and CONSOLE so you can help a user stuck in a loop, you can enter:

DUMP COMM=(TSO USER TERMINAL IN LOOP)

In response to this command, the system issues:

* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

If you think the default storage areas might not be enough to find the loop but you are sure that TERMINAL and CONSOLE are the address spaces you want, you can enter, for now:

R id,TSONAME=(TERMINAL,CONSOLE),CONT

Because you specified CONT, the system reissues:

* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

Now, if you decide that the default storage areas are really all you need to see, you can complete the dump request as follows:

R id,END

Example 4

Request a dump of two structures named CACHESTRUCTURE and LISTSTRUCTURE.

Enter: DUMP COMM=(Dump of CACHESTRUCTURE and LISTSTRUCTURE)

See: * id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

Enter: R id,STRLIST=(STRNAME=CACHESTRUCTURE,USERCNTLS,(STGCLASS=ALL),
STRNAME=LISTSTRUCTURE,LOCKENTRIES,(LISTNUM=ALL))

DUMP Command

This dump will include:

- Structure control data for CACHESTRUCTURE
- User attach control information for CACHESTRUCTURE
- Entry directory information for every entry, grouped by storage class.
- Structure control data for LISTSTRUCTURE
- All lock table entries for LISTSTRUCTURE
- List entry controls for every entry, grouped by list.

Example 5

Request a dump of LISTSTRUCTURE

Enter: DUMP COMM=(LISTSTRUCTURE, PRIORITIZE OPTIONS)

See: * id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

Enter: R id,STRLIST=(STRNAME=LISTSTRUCTURE,(LISTNUM=(5-8),SUMMARY),
LOCKENTRIES,CONT

See: * id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

Enter: R id,(LISTNUM=1,ADJUNCT=CAPTURE,ENTRYDATA=UNSERIALIZE))

This dump will include:

- Structure control data for LISTSTRUCTURE
- List control data for lists 5-8 (but no directory information for the entries)
- All lock table entries for LISTSTRUCTURE
- List entry controls, adjunct data and entry data for all entries in list 1. The adjunct data was captured with the list entry controls. The entry data may have changed relative to the adjunct data or entry controls that were captured while the system held structure dump serialization.

Example 6

Request a dump of CACHESTRUCTURE

Enter: DUMP COMM=(CACHESTRUCTURE, GROUP ENTRIES)

See: * id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

Enter: R x,STRLIST=(STRNAME=CACHESTRUCTURE,CONNNAME=USER,
ACCESSTIME= OVERRIDE,CONT

See: * id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

Enter: R x,(STGCLASS=(3-8,10),ENTRYDATA=SERIALIZE),(COCLASS=ALL))

This dump will include:

- Structure control data for CACHESTRUCTURE
- Directory information and entry data for all entries in storage classes 3-8 and 10. The entry data is written to the dump data set while structure dump serialization remains held. The entry data is unchanged relative to the captured adjunct or entry control information.
- Directory information for all entries grouped by cast-out class. The changed entries in storage classes 3-8 and 10 are dumped twice.
- The local cache index pertaining to USER is also dumped with each entry (and is dumped again with the changed entries in storage classes 3-8 and 10).

Example 7

The REMOTE parameter specifies a dump on another system in the sysplex, system S1; the SDATA parameter specifies the same SDATA options used for the local system's dump. The REMOTE parameter also specifies dumps on all the members of XCF groups G1 and G2; the DSPNAME parameter specifies the DSPNAME values specified for the local system's dump.

```
DUMP COMM=(REMOTE DUMP)
* 5,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 5,SDATA=(COUPLE),DSPNAME='XCFAS'.*,CONT
* 6,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 6,REMOTE=(SYSLIST=(S1),SDATA,
* 7,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 7,GRPLIST=(G1(*),G2(*)),DSPNAME)
```

You need CONT in R 5 because you are at the end of a parameter at the end of the line. You do not need CONT in R 6 because the end of the line is within the parentheses; the system prompts for a reply to complete the parenthetical value.

Example 8

The REMOTE parameter specifies dumps on S1 and all systems with names matching S2*. On these systems, the dumps are to include a data space with a name matching MYDS* for a job matching J* and a data space named THATDS for address space 01.

```
DUMP COMM=(REMOTE DUMP)
* 6,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 6,REMOTE=(SYSLIST=(S1,S2*),DSPNAME=('J*'.MYDS*,01.THATDS))
```

Example 9

The REMOTE parameter specifies dumps on members M1 and M2 of XCF group G1, with the SQA included in the dumps.

```
DUMP COMM=(REMOTE DUMP)
* 7,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 7,REMOTE=(GRPLIST=G1(M1,M2),SDATA=(SQA))
```

Example 10

Request a dump of jobname MYJOB on all systems in a sysplex, including the local system. JOBNAMESPACE=MYJOB requests the dump for the local system; the REMOTE parameter with SYSLIST=*(‘MYJOB’) requests the dump on all remote systems. Note that when specifying the jobname on the SYSLIST parameter, you must place the jobname in single quotes (‘MYJOB’), and there is no comma between the asterisk and (‘MYJOB’).

```
DUMP COMM=(MYJOB DUMP)
* 8,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 8,JOBNAMESPACE=MYJOB,REMOTE=(SYSLIST=*(‘MYJOB’))
```

Example 11

Request a dump of all jobs whose names begin with IRLM, on all systems in a sysplex, including the local system. You might have IRLMA, IRLMB, IRLMC, and so on.

```
DUMP COMM=(IRLM DUMP)
* 9,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 9,JOBNAMESPACE=IRLM*,REMOTE=(SYSLIST=*(‘IRLM*’))
```

Example 12

DUMP Command

Request a dump of all jobs whose names begin with IRLM, on all systems in a sysplex, including the local system. These jobs are the members of an XCF group called GRP1.

```
DUMP COMM=(IRLM DUMP)
* 10,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 10,JOBNAME=IRLM*,REMOTE=(GRPLIST=GRP1(*))
```

Example 13

Request a dump of all jobs that are members of an XCF group called GRP1, on all systems in a sysplex, including the local system. The jobnames begin with IRLM. Other XCF groups also exist, and contain members whose jobnames also begin with IRLM. You only want the members of GRP1 to be dumped. Note that the PROBDESC=SYSDLOCL parameter causes 2 dumps to be written for the local system: the master scheduler address space, and the address spaces for GRP1.

```
DUMP COMM=(IRLM GRP1 DUMP)
* 11,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 11,PROBDESC=SYSDLOCL,REMOTE=(GRPLIST=GRP1(*))
```

Example 14

Request a dump of *jobname1*, *jobname2*, and *jobname3* on all systems in a sysplex, including the local system.

```
DUMP COMM=(DUMP OF JOBNAMES 1, 2, AND 3)
* 12,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 12,JOBNAME=(jobname1,jobname2,jobname3),CONT
* 13,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 13,REMOTE=(SYSLIST=*(('jobname1','jobname2','jobname3')))
```

Example 15

Request a dump of jobname J on all systems in the sysplex except the local system.

```
DUMP COMM=(JOBNAME J DUMP)
* 14,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 14,REMOTE=(SYSLIST=*(J'))
```

Example 16

Request a dump of jobname J on systems named S1 and S2 in a sysplex.

```
DUMP COMM=(DUMP OF JOBNAME J ON S1 & S2)
* 15,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 15,REMOTE=(SYSLIST=(S1('J'),S2('J')))
```

Example 17

Request a dump of jobnames J and Q on all systems in a sysplex, including the local system.

```
DUMP COMM=(JOBNAMES J & Q DUMP)
* 16,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 16,JOBNAME=(J,Q),REMOTE=(SYSLIST=*(J,'Q'))
```

Example 18

Request a dump of jobname J on all systems in a sysplex, including the local system, and jobname Q only on system S2.

```
DUMP COMM=(JOBNAMES J ALL & Q S2 DUMP)
* 17,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 17,JOBNAM=J,REMOTE=(SYSLIST=*(‘J’),SYSLIST=S2(‘Q’))
```

Example 19

Request a dump of the following:

- XCF-related information from all systems in a sysplex including the local system
- All data spaces owned by jobs named MYJOB on all systems in a sysplex, including the local system.

Note that SDATA=(COUPLE) requests the XCF information for the local system. Specifying SDATA on the REMOTE parameter requests the same SDATA option for the remote systems as that used for the local system's dump.

DSPNAME=(‘MYJOB’.*) requests all data spaces belonging to jobs named MYJOB on the local system (note that while parentheses are included, they are optional, because you are specifying only one *dspname-entry*). Specifying DSPNAME on the REMOTE parameter requests the same DSPNAME option for the remote systems as that used for the local system's dump.

```
DUMP COMM=(XCF & MYJOB DUMP)
* 18,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 18,SDATA=(COUPLE),DSPNAME=(‘MYJOB’.* ),CONT
* 19,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 19,REMOTE=(SYSLIST=*,SDATA,DSPNAME)
```

Example 20

Request dumps of the following in a sysplex:

- On the local system, dump the XCF and common service area information.
- On the local system, dump data space DSP1 owned by MYJOB.
- On all remote systems, dump the XCF, common service area, and workload manager information.
- On all remote systems, dump all data spaces owned by MYJOB.

```
DUMP COMM=(VARIOUS DUMP`S)
* 20,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 20,SDATA=(COUPLE,CSA),DSPNAME=(‘MYJOB’.DSP1),CONT
* 21,IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
R 21,REMOTE=(SYSLIST=*,SDATA=(COUPLE,CSA,WLM),DSPNAME=(‘MYJOB’.* ))
```

DUMP Command Parmlib Examples

See *z/OS MVS Initialization and Tuning Reference* for examples of using the DUMP command with the IEADMCxx parmlib member.

DUMPDS Command

Use the DUMPDS command to:

- Change the system's list of dump data sets and resources
- Clear full SYS1.DUMP data sets and make them available for dumps
- Set up and alter the configuration of automatic dump data set allocation

Table 4-18. Summary of the DUMPDS Command

Command:	Topic:
DUMPDS ADD	"Adding System Dump Resources" on page 4-281
DUMPDS ALLOC	"Enabling and Disabling Automatic Dump Data Set Allocation" on page 4-284
DUMPDS CLEAR	"Making Dump Data Sets Ready To Receive Dumps" on page 4-285
DUMPDS DEL	"Deleting System Dump Resources" on page 4-285
DUMPDS NAME	"Setting the Name-Pattern for Dump Data Sets" on page 4-287

Notes:

1. You must issue the DUMPDS command from a console with SYSTEM authority.
2. SVC dump supports pre-allocated dump data sets and automatically allocated dump data sets.

Pre-allocated dump data sets are direct access data sets with names of the form SYS1.DUMPxx, where **xx** can be any decimal number from 00-99. You can allocate SYS1.DUMPxx data sets with both primary and secondary extents. When you allocate a dump data set, specify enough secondary extents to hold the entire dump. Also, specify RECFM=FB, LRECL=4160 and BLKSIZE=4160.

When automatic allocation is active, the dump is written to SMS-managed storage or to DASD volumes. The system allocates dump data sets of the correct size at the time a dump is requested. No pre-allocation is required for them.

3. A DUMPDS CLEAR or DUMPDS DEL command has no effect on any data set that is receiving an SVC dump when you issue the command.
4. Symbol substitution is supported on all sub-operands (e.g., ADD, ALLOC, ...) of the DUMPDS command except NAME=. (No substitution is done for the DD NAME=name-pattern command because the name-pattern may contain symbols that need to be passed through unchanged. See "Setting the Name-Pattern for Dump Data Sets" on page 4-287 for more information about the NAME= sub-operand of the DUMPDS command.)

Syntax

The syntax for each operand of the DUMPDS command is shown immediately preceding its respective parameter list.

DUMPDS or DD

Note: After using the DUMPDS command to make changes, you can use the DISPLAY DUMP command to verify, among other characteristics of dump data set automatic allocation:

- The status
- What resources are defined

- The naming convention
- The title and error-related data for pre-allocated and automatically allocated data sets

See “Displaying Dump Options or Dump Data Set Status” on page 4-119 for more information.

Adding System Dump Resources

Use the DUMPDS ADD command to add specific SYS1.DUMP data sets, SMS classes, or non-SMS-managed DASD volumes to the system’s list of dump data set resources.

```
DD ADD, {DSN={nn
              {(nn[,nn]...)
               {nn-nn
                {(nn-nn[,nn-nn]...)
                 {(nn[,nn]...,nn-nn[,nn-nn]...)}
                  {ALL
                   {(ALL)
                    {SMS={class
                      {(class[,class]...)
                       {VOL={volser
                         {(volser[,volser]...)}}
```

Where **class** represents:

```
{storclas
{([DATA|D=[dataclas]] [,MGMT|M=[mgmtclas]] [,STOR|S=[storclas]])}
```

ADD,DSN=nn or ALL

Identifies the direct access data sets you request the system to add to its list of SYS1.DUMP data sets. You should allocate, catalog, and protect the data sets you specify before using this command.

If any direct access data set you specify is empty, or does not contain a valid dump, the system marks it as available for a dump. If any data set you specify is full and you want to make it available for a dump, you must clear it. If you want to keep the contents of the data set, process the data set with the interactive problem control system, IPCS. For more information about IPCS see the *z/OS MVS IPCS User’s Guide*, GC28-1631.

If you do not want to keep the contents of the data set, issue a DUMPDS CLEAR,DSN= command for the data set. That will clear the data set and mark it as available for a dump, but not save the data set contents.

DUMPDS ADD,DSN= does not process any direct access SYS1.DUMP data set that is already on the system’s list of SYS1.DUMP data sets.

nn The two-digit decimal identifier (00-99) of a direct access SYS1.DUMP data set you want the system to add to its list of SYS1.DUMP data sets. You can specify one or more single identifiers and/or one or more ranges of identifiers. For any range of identifiers you specify, the first identifier must be less than the second. If you specify more than one range, use a comma between them and enclose the entire set in parentheses.

DUMPDS Command

ALL

Directs the system to add to its list of SYS1.DUMP data sets all the cataloged direct access SYS1.DUMP data sets not already on the list.

ADD,SMS=class

Designates SMS classes you request the system to add to its list of resources eligible for allocation as dump data sets. Installation-written automatic class selection (ACS) routines, if present, may use but can override a data, management, or storage class you specify. When a dump is taken, allocation of a dump data set to an SMS class occurs only if SMS is active. When a dump is taken, allocation of a dump data set to an SMS resource takes place only if automatic allocation is active.

class

The SMS class(es) you want the system to consider allocating as dump data sets. If you specify multiple SMS classes, enclose them in parentheses and separate them with commas.

storclas

The SMS class specifying the one- to eight-character SMS storage class you want added to the system's list of resources eligible for automatic allocation.

(**DATA**=*dataclas*, **MGMT**=*mgmtclas*, **STOR**=*storclas*)

The SMS class consisting of the specified combination of data, management, and storage class you want SMS to pass to the ACS routines.

dataclas

The 1-8 character name of the data class you want SMS to pass to the ACS routine.

mgmtclas

The 1-8 character name of the management class you want SMS to pass to the ACS routine.

storclas

The 1-8 character name of the storage class you want SMS to pass to the ACS routine.

The DATA, MGMT, and STOR keywords are optional.

ADD,VOL=*volser*

Identifies the non-SMS-managed direct access volume(s) you request the system to add to its list of resources for automatic allocation of dump data sets. Allocation will assign space from the first resource in the list until that resource is full, then use the next resource. If a dump is taken when automatic allocation is active, allocation of a dump data set to a non-SMS-managed DASD volume takes place if either of the following conditions occurs:

- No SMS classes are defined.
- SMS classes are defined but an attempt to allocate a dump data set using those classes failed (for example because of space problems).

volser

The 1-6 character volume serial identifier of the direct access volume you want added to the system's list of resources for automatic allocation. You can specify one or more direct access volume serial identifiers. (Enclose multiple volume serial identifiers in parentheses and separate them with commas.)

Notes:

1. Protect the data sets using your normal password or RACF procedures.
2. SMS cannot manage DASD volumes specified for automatic allocation of dump data sets.
3. If resources assigned for automatic allocation become full, the system attempts to write dumps to pre-allocated dump data sets.

Example 1

To put the allocated and cataloged direct access data set SYS1.DUMP02 on the system's list of SYS1.DUMP data sets, enter:

```
DUMPDS ADD,DSN=02
```

Example 2

To add the following allocated and cataloged direct access data sets to the system's list of SYS1.DUMP data sets,

```
SYS1.DUMP00-SYS1.DUMP05,  
SYS1.DUMP08, and  
SYS1.DUMP10-SYS1.DUMP12,
```

enter:

```
DD ADD,DSN=(00-05,08,10-12)
```

Example 3

To add to the system's list of SYS1.DUMP data sets all the allocated and cataloged direct access dump data sets not already on the list, enter:

```
DD ADD,DSN=ALL
```

Example 4

To specify the storage class DUMPC1, or if it is not available, storage class DMPADIT, for SMS to validate for use as dump data sets (which an ACS routine could override), enter:

```
DD ADD,SMS=(DUMPC1,DMPADIT)
```

Example 5

To use the data, management, and storage class defaults defined by the installation's SMS ACS routine, enter:

```
DD ADD,SMS=()
```

Example 6

To request that SMS consider allocating dump data sets from data class DUMP and management class KEEP, enter:

```
DD ADD,SMS=(DATA=DUMP,MGMT=KEEP,STOR=)  
or  
DD ADD,SMS=(D=DUMP,M=KEEP)
```

Example 7

DUMPDS Command

To specify that an installation's ACS routine consider allocating as dump data sets SMS storage class DUMP, and then when DUMP is filled, storage class DUMPTEMP, enter:

```
DD ADD,SMS=((STOR=DUMP),(S=DUMPTEMP))
```

The installation's SMS ACS routine will define the data and management classes for these storage classes. If you desire specific data or management classes, you must explicitly identify them. For example, to use management class KEEP with storage class DUMP, and management class SCRATCH with storage class DUMPTEMP, enter:

```
DD ADD,SMS=((STOR=DUMP,M=KEEP),(M=SCRATCH,S=DUMPTEMP))
```

Enabling and Disabling Automatic Dump Data Set Allocation

Use the DUMPDS ALLOC command to activate or deactivate the automatic allocation of dump data sets.

```
DD ALLOC={ACTIVE|INACTIVE}
```

ALLOC=ACTIVE

Dump data sets are automatically allocated when a dump is requested. Any of the resources that have been defined by the DUMPDS ADD command as available for automatically allocated dump data sets are used. If no automatic allocation resources are defined, the system issues message IEA799I and writes the dump to a pre-allocated dump data set on its list of SYS1.DUMP data sets. If no pre-allocated dump data sets are on the system's list of SYS1.DUMP data sets, then message IEA793A is issued requesting operator intervention. The requested dump is kept in virtual storage until an automatic allocation resource is defined, a pre-allocated dump data set is made available either by allocating a new one or clearing an existing one, or the dump is deleted either by operator request or expiration of the CHNGDUMP MSGTIME parameter.

ALLOC=INACTIVE

This is the initial state of the system after IPL. Dump data sets are not automatically allocated when a dump is requested. Resources defined by the DUMPDS ADD command as available for automatic allocation of dump sets are not used. Any requested dump will be written to a pre-allocated dump data set specified on the system's list of SYS1.DUMP data sets. If no pre-allocated dump data sets are specified on the system's list of SYS1.DUMP data sets, then message IEA793A is issued requesting operator intervention. The requested dump is kept in virtual storage until automatic allocation is enabled, a pre-allocated dump data set is made available either by allocating a new one or clearing an existing one, or the dump is deleted either by operator request or expiration of the CHNGDUMP MSGTIME parameter.

Example

To make the automatic dump data set allocation function inactive, without changing the automatic allocation resources or naming convention, enter:

```
DD ALLOC=INACTIVE
```

This is the system default.

Making Dump Data Sets Ready To Receive Dumps

Use the DUMPDS CLEAR command to empty the specified data set and mark it as available to receive a dump.

```
DD CLEAR,DSN={nn
  {(nn[,nn]...)}
  {nn-nn}
  {(nn-nn[,nn-nn]...)}
  {(nn[,nn]...,nn-nn[,nn-nn]...)}
  {ALL}
  {(ALL)}}
```

CLEAR,DSN=nn or ALL

Clear and mark as available for dumps the specified direct access dump data sets on the system's list of SYS1.DUMP data sets. The system clears each full direct access dump data set by writing an end-of-file mark at the beginning of the data set.

A DUMPDS CLEAR,DSN= command does not process any data set that is not in the system's list of SYS1.DUMP data sets.

nn The two-digit decimal identifier (00-99) of a direct access SYS1.DUMP data set you want to clear and mark as available for a dump. You can specify one or more single identifiers and/or one or more ranges of identifiers. If you specify a range of identifiers, the first identifier **must** be less than the second identifier.

ALL

Clears and marks as available for a dump all direct access dump data sets in the system's list of SYS1.DUMP data sets.

Example 1

To clear, and mark as available for dumps, the direct access data sets SYS1.DUMP00-SYS1.DUMP05 and SYS1.DUMP09, enter:

```
DD CLEAR,DSN=(00-05,09)
```

Example 2

To clear and mark as available for dumps all the full direct access data sets on the system's list of SYS1.DUMP data sets, enter:

```
DD CLEAR,DSN=(ALL)
```

Deleting System Dump Resources

Use the DUMPDS DEL command to remove from the system's list of dump data set resources specific SYS1.DUMP data sets, SMS classes or DASD volumes.

DUMPDS Command

```
DD DEL,{DSN={nn
             {(nn[,nn]...)
              {nn-nn
               {(nn-nn[,nn-nn]...)
                {(nn[,nn]...,nn-nn[,nn-nn]...)}
                {ALL
                 {(ALL)
                  }}}
```

```
{SMS={class
          {(class[,class]...)
           {ALL
            {(ALL)
             }}}
```

```
{VOL={volser
          {(volser[,volser]...)
           {ALL
            {(ALL)
             }}}
```

Where **class** represents:

```
{storclas
  {[DATA|D=[dataclas]] [,MGMT|M=[mgmtclas]] [,STOR|S=[storclas]])}
```

DEL,DSN=nn or ALL

The system is to remove the specified direct access dump data sets from its list of SYS1.DUMP data sets. In response to this command, the system does not uncatalog or change the space allocation for any data set you specify.

A DUMPDS DEL,DSN= command does not process any data set that is not on the system's list of SYS1.DUMP data sets.

nn The two-digit decimal identifier (00-99) of a cataloged direct access SYS1.DUMP data set you want to remove from its list of SYS1.DUMP data sets. You can specify one or more single identifiers or one or more ranges of identifiers. If you specify a range of identifiers, the first identifier **must** be less than the second identifier.

ALL

The system is to remove all pre-allocated direct access dump data sets from its list of SYS1.DUMP data sets.

DEL,SMS=class or ALL

The system is to remove the specified SMS resources from the system's list of resources for automatic allocation of dump data sets. Removing SMS resources does not deactivate automatic allocation of dump data sets.

class

The SMS resource you want to remove from the system's list of resources for automatic allocation. You can specify one or more SMS resources. Enclose multiple resources in parentheses, separated by commas.

storclas

The SMS resource consisting of the specified 1-8 character SMS storage class you want removed from the system's list of resources for automatic allocation.

(DATA=dataclas, MGMT=mgmtclas, STOR=storclas)

The SMS resource consisting of the specified combination of data, management, and storage class you want removed from the system's list of resources for automatic allocation.

dataclas

The 1-8 character SMS data class you want specified for this allocation resource.

mgmtclas

The 1-8 character SMS management class you want specified for this allocation resource.

storclas

The 1-8 character SMS storage class you want specified for this allocation resource.

The DATA, MGMT, and STOR keywords are optional and may be specified only once per resource.

ALL

Requests that all SMS resources classes be removed from the system's list of resources available for automatic dump data set allocation.

DEL,VOL=volser or ALL

Remove the specified direct access volume(s) from the system's list of resources for automatic allocation of dump data sets. Removing direct access volumes does not inactivate automatic allocation of dump data sets.

volser

The 1-6 character volume serial number of the direct access volume you want to remove from the system's list of resources for automatic allocation. You can specify one or more direct access volume serial numbers. Multiple volume serial numbers must be enclosed in parentheses and separated by commas.

ALL

Request that all DASD volumes be removed from the system's list of resources available for automatic allocation of dump data sets.

Example 1

To remove SYS1.DUMP02 from the system's list of SYS1.DUMP data sets, enter:

DD DEL,DSN=02

Example 2

To remove all direct access data sets from the system's list of SYS1.DUMPnn data sets, enter:

DD DEL,DSN=ALL

Example 3

To remove SMS resources consisting of the storage classes DUMPC1 and DMPADIT from the system's list of resources for automatic allocation of dump data sets, enter:

DD DEL,SMS=(DUMPC1,DMPADIT)
or
DD DEL,SMS=(DUMPC1,(STOR=DMPADIT))

Setting the Name-Pattern for Dump Data Sets

Use the DUMPDS NAME command to establish a name-pattern for automatically allocated dump data sets.

DUMPDS Command

```
DD NAME=name-pattern
```

The parameter is:

NAME=*name-pattern*

The system names automatically allocated dump data sets according to the naming convention specified by the name-pattern. The name-pattern can include both text and system symbols. The system substitutes text for the system symbols when it creates data set names.

The system default name-pattern is:

```
SYS1.DUMP.D&YYMMDD..T&HHMMSS..&SYSNAME..S&SEQ.
```

Note: If you want to use a name pattern other than the system default, place the DUMPDS NAME command before the DUMPDS ADD commands, so that the system uses the correct name pattern for the added resources. Using a different name pattern could cause the system to use different RACF profiles for the allocation.

Before you specify system symbols in the DUMPDS NAME command, read the rules and recommendations for using system symbols in commands in “Sharing System Commands” on page 1-14.

Notes:

1. All name-patterns must specify the &SEQ. sequence number system symbol to ensure uniqueness. The system rejects the name-pattern if you do not specify &SEQ..
2. When you change the installation naming convention for dump data sets, also change the procedures for protecting them (password or RACF protection).
3. If the installation has a SYSNAME that begins with a numeral, you must specify an alternate NAME. The default, which includes .&SYSNAME., will generate a data set name error when SDUMP tries to use it.
4. Names generated by a name-pattern must adhere to MVS data set naming conventions and limitations. Ensure that system symbols do not return a numeric character as the first character of any qualifier in the data set name. For example, in the default name-pattern:

```
SYS1.DUMP.D&YYMMDD..T&HHMMSS..&.SYSNAME..S&SEQ.
```

&HHMMSS, &SEQ, and &YYMMDD return numeric substitution texts. Each symbol is preceded by an alphabetic character to avoid placing a numeric character in the first character of each qualifier. If resolved substitution texts create a data set name that is not valid, the system rejects the name-pattern and issues message IEE855I. The previous name-pattern remains in effect.

5. The DUMPDS NAME function will not work when the command is issued from an MCS or SMCS console and the character ‘&’ is specified as a command delimiter. Command delimiters are defined using the CMDDELIM parameter on the INIT statement of the CONSOLxx parmlib member. See *z/OS MVS Initialization and Tuning Reference* for more information on defining command delimiters.

Example

To establish automatically allocated dump data sets with names such as

```
SYS1.USERIDX.T025930.S00001
```

DUMPDS Command

where the system name is *SYS1* and *USERIDX* is the name of the job requesting the dump, enter:

```
DD NAME=&SYSNAME..&JOBNAME..T&HHMMSS..S&SEQ.
```

FORCE Command

Consider using the FORCE command as a last resort when the CANCEL command still fails to perform its function after you have issued it several times.

The following table summarizes the tasks that the FORCE command can perform. Below the table are several considerations about using the FORCE command.

Table 4-19. FORCE Command Tasks

Task - Immediately Terminate:	Syntax:
<ul style="list-style-type: none">• A job in execution• A running Advanced Program-to-Program Communication/MVS (APPC/MVS) transaction program• A started task	FORCE <i>jobname</i>
<ul style="list-style-type: none">• A time-sharing user (<i>U=userid</i>)	FORCE <i>U=userid</i>
<ul style="list-style-type: none">• A started task• A MOUNT command• An external writer allocation• The output processing for a job• A z/OS UNIX process	FORCE <i>identifier</i>

Considerations

- FORCE is not a substitute for CANCEL. Unless you issue CANCEL first for a cancellable job, the system issues error message IEE838I. The steps to use in the process are:
 1. Issue the CANCEL *nnn* command, making several attempts if necessary.
 2. Use the DUMP command — if you want a dump produced. Respond to the prompt for parameters with the *jobname* or ASID of the "stuck" job, as well as ASID(1)=MASTER.
 3. Issue the FORCE *nnn,ARM* command for non-cancellable procedures.
 4. Issue the FORCE *nnn* command only when the previous steps fail.
- **WARNING:** Never use the FORCE command without understanding that:
 - After issuing FORCE, you might have to re-IPL.
 - If you issue FORCE for a job in execution or for a time-sharing user, the system deletes the affected address space and severely limits recovery unless you use the ARM parameter. (Arm is described below.)
 - If you need a dump, you must **issue a DUMP command before you issue FORCE**. Once you've issued a FORCE command it is usually NOT POSSIBLE to get a dump of the failing address space.
 - If your system was part of a global resource serialization ring (GRS=START, GRS=JOIN or GRS=TRYJOIN was specified at IPL) but has been quiesced (by entering the VARY GRS(system name),QUIESCE command), FORCE processing might not complete immediately. The system suspends termination of all address spaces holding global resources until the quiesced system rejoins the ring or is purged from the ring. Use a DISPLAY GRS command to determine GRS status.
- Do not FORCE a job that is in a loop; use the RESTART function. See "Using the System Restart Function" on page 1-18 for more information.
- The availability manager (AVM) cannot be ended by a STOP or CANCEL command. To end AVM, a FORCE AVM,ARM command is required.

- When you use the FORCE command to end the availability manager (AVM) address space, the operator must restart that address space by issuing the command START AVM,SUB=MSTR.
- You can enter FORCE only from a console with master authority.

Syntax

The complete syntax for the FORCE command is:

```
FORCE {jobname
       {U=userid
        {[jobname.]identifier}}
```

Parameters

jobname

The name of the batch job, started task, or APPC/MVS transaction program you want to end.

The name of a started task is based on whether the JOBNAME= keyword was specified on the START command.

If JOBNAME= was specified, *jobname* is the name assigned to the started task.

If JOBNAME= was not specified and the source JCL for the started task is

- A **job**, the system will use the job name from the JCL JOB statement.
- A **procedure**, the system will use the member name as the job name.

Notes:

- When you use the FORCE command to end a job in execution, you also terminate the address space for the job and any other tasks executing in that address space. If you use FORCE for a job running under an initiator, you terminate the initiator along with the job. With JES2 on your system, you must issue another START command to recover use of such an initiator. With JES3 on your system, this additional START command might not be necessary.
- When you force an APPC/MVS transaction program, you can find *jobname* (the transaction program's name as specified in the TP PROFILE in the address space) on the output by issuing a DISPLAY ASCH,A command.
- Entering FORCE for an external writer while the system is allocating the writer to a job terminates both the device allocation and the writer itself. Entering FORCE for an external writer while the writer is processing output for a job terminates both the output processing and the writer itself.

U=userid

The user ID of the time-sharing user to terminate.

If the user is just logging on and does not yet have a unique name, you must find out the address space identifier for the user (see the explanation under **A=asid**) and use the following version of the command:

FORCE U=LOGON*,A=asid

[jobname.]identifier

The identifier for the unit of work to terminate, optionally preceded by the job name. You can use the following types of identifiers:

- The identifier that was specified on the START command.

FORCE Command

- */devnum*, the device number specified when the START or MOUNT command was entered. The device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). You can precede the device number with a slash to prevent ambiguity between the device number and a device type or identifier.
- *devicetype*, the type of device specified when the START or MOUNT command was issued.

If no identifier was specified on the START command, the system assigns temporary identifier "STARTING" to the unit of work, until the system can assign an identifier according to the following order of precedence:

1. If an identifier was not specified on the START command, the identifier is the device type (for example, 3410) or device number (for example, X'0000') specified on the START or MOUNT command.
2. If an identifier, a device type, or a device number was not specified on the START or MOUNT command, the identifier is the device type specified on an IEFRDER DD statement (invoking a cataloged procedure) in the JCL.
3. If none of the above was specified, the identifier defaults to the job name.

Specifying both the job name and the entire identifier causes the command to take effect if *one and only one* work unit with that combination of job name and identifier is running. Where two or more work units with the same combination of job name and identifier are running, see "A=asid" below.

ARM

The system is to terminate the specified job, time-sharing user, or started procedure if it is non-cancellable. If the FORCE ARM command fails to terminate the address space within a reasonable time, reissue FORCE with the ARM parameter. The ARM parameter executes normal task termination routines without causing address space destruction. The system rejects this parameter if the address space for the specified job, time-sharing user, or started procedure cannot be terminated or should be terminated via the CANCEL command. If the command still fails after several attempts, try issuing FORCE without the ARM parameter.

Note: This keyword is not related to the ARMRESTART parameter and the functions of the automatic restart manager.

A=asid

The hexadecimal address space identifier of the work unit to terminate.

If two or more work units are running with the same job name, identifier, combination of job name and identifier, or user ID that you specified on the FORCE command, the system rejects the command because it does not know which work unit to terminate. To avoid this, you must add the parameter A=asid to your original FORCE command in order to specify the address space identifier of the work unit.

To find out the address space identifier for a unit of work, you can use the DISPLAY command as follows:

DISPLAY JOBS,ALL

Lists the address space identifiers for all batch jobs and started tasks.

DISPLAY ASCH,ALL

Lists the address space identifiers for all APPC/MVS transaction programs.

DISPLAY TS,ALL

Lists the address space identifiers for all logged-on time-sharing users.

DISPLAY OMVS,ASID=ALL or DISPLAY OMVS,A=ALL

Lists the address space identifiers for all z/OS UNIX processes.

ARMRESTART

Indicates that the batch job or started task should be automatically restarted after the force has completed, if it is registered as an element of the automatic restart manager. If the job or task is not registered, or if you do not specify this parameter, MVS will not automatically restart the job or task.

Example 1

To terminate an earlier MOUNT command for a 3380 device, enter:

FORCE 3380

Example 2

To terminate an earlier MOUNT command for the device number 3380, enter:

FORCE /3380

Example 3

To remove job JOBXYZ from the system, enter:

FORCE JOBXYZ

Example 4

To stop device allocation for writer 1AF and terminate the writer itself, enter, during device allocation for writer 1AF:

FORCE 1AF

Example 5

To stop the output processing on a writer to device number B1AF and terminate the writer itself, enter:

FORCE /B1AF

Example 6

To log user A237 off the system, enter:

FORCE U=A237

Example 7

To terminate the non-cancellable job BIGTASK, enter:

FORCE bigtask,ARM

Example 8

To terminate the non-cancellable job SERVICE with the address space identifier of 1A8, enter:

FORCE service,arm,a=1a8

Example 9

FORCE Command

To FORCE an APPC/MVS transaction program whose jobname is CALENDAR and whose address space identifier is 3B, enter:

```
FORCE CALENDAR,A=3B
```

HALT Command

Use the HALT command to record statistics before stopping the operating system. After you have stopped all subsystem processing (through the use of the appropriate subsystem command) and the system notifies you that all system activity has completed, you can issue the HALT EOD command to ensure that important job and system statistics and data records in storage are recorded.

Note: Do not use the HALT command if you intend to keep running, because this command:

- closes the system log
- allows SMF to continue writing records after switching to a new data set. During the next IPL, you might see message IEE949I, indicating the presence of old SMF data from a previous IPL. This SMF data was written after HALT was issued.

Syntax

The complete syntax for the HALT command is:

```
Z EOD
```

The HALT EOD command causes the system to take the following steps:

- Store the internal I/O device error counts in the logrec data set.
- Empty the SMF buffers onto the active SMF data set in SYS1.MANx.
- Copy the cached data in 3990 devices to DASD.
- Switch to another SMF data set in SYS1.MANx, allowing the previously active SMF data set to be dumped according to your installation's procedures.
- Close the system log and put it on the print queue.

When these actions are completed, the system sends you the message:

IEE334I HALT EOD SUCCESSFUL.

IOACTION Command

Use the IOACTION command to stop and resume I/O activity to direct access storage devices (DASDs) without varying the DASD offline, when the DASD is shared between systems AND is in recovery by the input/output system (IOS).

CAUTION:

Use this command only in response to the IOS recovery messages IOS427A and IOS062E and wait state X'062'.

After the IO STOP command is entered, the system allows several seconds for current I/O activity to end.

Do not leave devices stopped any longer than necessary to perform recovery. System storage is used by all initiated I/O operations and is only freed after the I/O operations complete.

Do not use this command for devices that contain system-owned data sets or the system residence volume, or page data sets. Also, EREP will not run while devices are stopped.

The system displays message IOS601I to remind you that I/O activity is stopped to the specified DASD. This message remains displayed until all I/O activity is resumed with the IOACTION RESUME command.

Syntax

The complete syntax for the IOACTION command is:

```
IO {STOP,DEV=([/]devnum[,/]devnum]...) }  
{STOP,DEV=([/]lowdevnum-[/]highdevnum[,/]lowdevnum-[/]highdevnum]...) }  
{RESUME,DEV=([/]devnum[,/]devnum]...) | ALL }  
{RESUME,DEV=([/]lowdevnum-[/]highdevnum[,/]lowdevnum-[/]highdevnum]...) }
```

Note: You can enter individual device numbers and ranges on the same command. For example:

```
IO RESUME,DEV=(/2233,/990-/1012,160)
```

Parameters

STOP,DEV

The system stops all I/O activity to the specified DASD. The system allows several seconds for current activity to complete. You do not have to enter the parentheses when specifying only one device. The system displays message IOS601I to remind you that I/O activity is stopped to specified DASD. To display stopped DASD, enter DISPLAY IOS,STOP.

Note: Before stopping a device, enter D U,DASD,ALLOC,*devnum* to determine what data sets will be affected. If any system-owned data sets, such as SYS1.LINKLIB, are stopped, the system will be affected.

RESUME,DEV

The system resumes normal I/O activity to the specified devices. When ALL is specified, I/O activity is resumed on any device that had been stopped by an

IOACTION STOP command on that system. You do not have to enter the parentheses when specifying only one device.

[/]devnum

The device number of a DASD for which the system is to stop or resume I/O activity. You do not have to enter the parentheses when specifying only one DASD.

[/]lowdevnum-[/]highdevnum

The lower device number *lowdevnum* and the upper device number *highdevnum* of a range of DASDs for which the system is to stop or resume I/O activity. You do not have to enter the parentheses when specifying only one range of DASDs.

A device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/).

Example 1

To stop I/O activity to device numbers 1A0 through 1AF, enter:

IO STOP,DEV=(1A0-1AF)

Example 2

To stop I/O activity to device number 1B0, enter:

IOACTION STOP,DEV=1B0

Example 3

To resume I/O activity to device number 1B0 and device numbers 1A0A through 1AFA, enter:

IO RESUME,DEV=(1B0,1A0A-1AFA)

Example 4

To resume I/O activity to device number 3480, enter:

IO RESUME,/3480

Example 5

To resume I/O activity to all devices previously stopped, enter:

IO RESUME,ALL

LIBRARY Command

Use the LIBRARY command to perform any of several tasks associated with tape drives and tape volumes.

For a complete description of the syntax and parameters of the LIBRARY command, including the tasks that command can perform, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*, and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

LOG Command

Use the LOG command to make an entry into the system log, the OPERLOG, or the system log and the OPERLOG.

Syntax

The complete syntax for the LOG command is:

```
L 'text'
```

Parameters

'text'

The entry (up to 122 characters) to be made in the system log, the OPERLOG or the system log and the OPERLOG.

Lowercase letters are permitted if the printer on which the log will be printed can print lowercase letters.

Note: Lowercase characters in quotes are not converted to uppercase.

Example

To include the following comment in the system log, the OPERLOG or the system log and the OPERLOG enter:

```
L 'DEVICE 235 OFFLINE FOR REPAIRS'
```

LOGOFF Command

LOGOFF Command

Use the LOGOFF command to log off from an MCS or SMCS console.

LOGOFF is the recommended method of ending an SMCS console session.

Syntax

The complete syntax for the LOGOFF command is:

```
LOGOFF
```

You must issue LOGOFF when you leave your console and your installation requires operators to log on before issuing commands.

When your installation requires LOGON, the LOGOFF command leaves the console in a secure state. This applies to MCS consoles only. The system does not accept commands from this console until another LOGON command is completed.

For SMCS consoles, use LOGOFF to deactivate the console. You can use LOGOFF whether the operator has logged on to the SMCS console or not.

For MCS consoles, if LOGON is automatic at your installation, the system issues another automatic MCS LOGON command for this console, after performing the LOGOFF.

Example

To leave your console secure, enter:

```
LOGOFF
```

LOGON Command

Use the LOGON command to identify yourself to the system when your installation requires operators to log on before issuing commands.

The LOGON command enables an operator to access the SMCS console. An installation can indicate that this command be mandatory. **It is suggested that if an SMCS console session can be established from outside a secure area, logging on should be mandatory.**

To remove the LOGON prompt from the screen for MCS consoles, use the CLEAR key or the PA2 key. For SMCS consoles, the prompt cannot be cleared from the screen until the console is successfully logged on. To restore the prompt enter:

`LOGON [userid]`

The LOGON command restores the logon prompt display.

The *userid* is an 8-character field where you enter your operator *userid*. The *userid* parameter is optional; the system will prompt you for it.

Scope in a Sysplex

The system does not substitute text for system symbols specified in the LOGON command.

Syntax

The complete syntax for the LOGON prompt follows the message:

```
IEE187I    ENTER LOGON PARAMETERS
LOGON {userid}      PASSWORD {password}
GROUP  [racfgroup]  SECLABEL [label]
```

Parameters

LOGON {*userid*}

The panel displays the LOGON prompt in a protected field. The *userid* is an 8-character field where you enter your operator *userid*. The *userid* parameter is required.

PASSWORD {*password*}

The panel displays the PASSWORD prompt in a protected field. The *password* is a 26-character field where you enter your *password* of up to 8 characters. The input to this field is not displayed. The *password* field allows you to change your *password* by using the old-password/new-password/new-password format. The *password* parameter is required.

GROUP [*racfgroup*]

The panel displays the GROUP prompt in a protected field. The *racfgroup* is an 8-character field where you enter your RACF group identifier. The *racfgroup* parameter is optional.

LOGON Command

SECLABEL [*label*]

The panel displays the SECLABEL prompt in a protected field. The *label* is an 8-character field where you enter your RACF security label identifier. The *label* parameter is optional.

Notes:

1. The syntax of the user id, password, group id, and security label is defined by RACF.
2. You can use the tab keys to tab from one input field to the next on the LOGON prompt.
3. The LOGON command for MCS and SMCS consoles is supported for full capability display consoles only.
4. Changes made to a user's access authority to a logged-on console may not take effect until the user logs off and then back on again to the console.

MODE Command

MODE VS command no longer supported

The MODE VS command is not supported on systems at z/OS V1R6 and higher.

Use the MODE command to control the actions of recovery management when certain types of machine check interruptions occur. The actions you can control are:

- The recording/monitoring status for each type of machine check interruption controlled by the MODE command — degradation machine check interruptions on the logrec data set. For the procedure to print the logrec data set, see the *EREP User's Guide* and the *EREP Reference*
- The monitoring of hard machine checks, including machine checks that indicate timing facility damage
- The suppressing of system recovery or degradation machine-check interruptions

Table 4-20 summarizes the information that the MODE command provides.

Table 4-20. Summary of the MODE Command

Command:	Topic:
MODE AD MODE CC MODE IV MODE PD MODE PS MODE PT MODE SC MODE SD MODE SL MODE TC	"Controlling the Recording of Hard Machine Check Interruptions" on page 4-304
MODE DG MODE SR	"Controlling the Recording of System Recovery and Degradation Machine Check Interruptions" on page 4-306
MODE STATUS	"Displaying Recording and Monitoring Status" on page 4-308

You can enter the MODE command any number of times for any processor. Issuing the MODE command for a particular type of machine check changes only the recording or monitoring mode for that type of machine check, and changes it only for the processor(s) specified (or for all processors if no particular processor is specified). To change the recording or monitoring mode for several types of machine checks, you must enter a series of MODE commands. Each such MODE command specifies one type of machine check and the desired recording or monitoring mode for that type of machine check. If you issue the MODE command more than once for the same type of machine check, the last command (most recent) supersedes the previous commands.

Syntax

The syntax for each variation of the MODE command is shown immediately preceding its respective parameter list.

MODE

MODE Command

Table 4-21 shows the machine check interruption types you can specify and the parameters allowed for each type:

Table 4-21. MODE Parameters Allowed for Machine Check Interruptions

Machine Check Interruption Type	Command Parameters				
	CPU	QUIET	RECORD	REPORT	INTERVAL
DG — degradation	X	X (note 1)	X (notes 1, 2)	X (note 2)	
SR — system recovery	X	X (note 1)	X (notes 1, 2)	X (note 2)	
PD — instruction processing damage	X		X (note 3)		X (note 3)
SD — system damage	X		X (note 3)		X (note 3)
IV — invalid PSW or registers	X		X (note 3)		X (note 3)
TC — TOD clock damage	X		X (note 3)		X (note 3)
PT — processor timer damage	X		X (note 3)		X (note 3)
CC — clock comparator damage	X		X (note 3)		X (note 3)
PS — primary synchronization damage	X		X (note 3)		X (note 3)
AD — ETR-attachment damage	X		X (note 3)		X (note 3)
SL — switch to local synchronization	X		X (note 3)		X (note 3)
SC — ETR synchronization check	X		X (note 3)		X (note 3)
Notes:					
<ol style="list-style-type: none"> 1. QUIET and RECORD= are mutually exclusive. 2. REPORT= can only be used with RECORD=ALL. 3. RECORD=ALL and INTERVAL are mutually exclusive. 					

When you specify more than one option, you can enter the parameters in any order but must separate them by commas.

Controlling the Recording of Hard Machine Check Interruptions

You can use the MODE command to control the recording or monitoring of hard machine-check interruptions.

MODE {PD} [, INTERVAL={nnnnn}] [, RECORD[=nnn] [, CPU={x }]			
{SD}	<u>{300 }</u>	=ALL	{ALL}
{IV}		=25	
{TC}		= <u>16</u>	
{PT}		= <u>5</u>	
{CC}			
{VS}			
{PS}			
{AD}			
{SL}			
{SC}			

The parameters are:

PD

Instruction-processing damage machine checks are to be monitored in the specified mode.

SD

System damage machine checks are to be monitored in the specified mode.

IV

Machine checks indicating invalid PSW or registers are to be monitored in the specified mode.

TC

Machine checks indicating TOD clock damage are to be monitored in the specified mode.

PT

Machine checks indicating processor timer damage are to be monitored in the specified mode.

CC

Machine checks indicating clock comparator damage are to be monitored in the specified mode.

PS

Machine checks indicating primary clock synchronization are to be monitored in the specified mode.

AD

Machine checks indicating ETR attachment are to be monitored in the specified mode.

SL

Machine checks indicating switch to local synchronization are to be monitored in the specified mode.

SC

Machine checks indicating ETR synchronization checks are to be monitored in the specified mode.

INTERVAL=nnnnn

This parameter is used together with the RECORD=nnn parameter. It defines the number of seconds used in counting hard machine check interrupts. If the specified number of seconds elapses before the specified number of interrupts of the specified type occur on the specified processor, the count of that type of interrupt is set to zero, and the counting is started again from zero. If the specified number of hard machine check interrupts does occur in the specified

MODE Command

interval, then the system invokes alternate CPU recovery (ACR) to take the failing processor offline. If the INTERVAL parameter is omitted, then INTERVAL=300 is assumed.

RECORD=nnn

After the specified number (1 to 999) of hard machine checks of the specified type occurs on the specified processor in the specified interval, the system invokes alternate CPU recovery (ACR) to take the failing processor offline. All interruptions of that type occurring on that processor are recorded on the logrec data set until ACR is invoked, including the interruption that caused the invocation of ACR. If no number is specified or if the RECORD parameter is omitted, the system uses RECORD=16 for PD, RECORD=25 for SL, and RECORD=5 for all others.

RECORD=ALL

All specified hard machine-check interruptions of the specified type occurring on the specified processor are to be recorded on the logrec data set. The system will no longer monitor the frequency of hard machine-check interruptions of that type occurring on that processor.

CPU=x

The address (0, 1, 2, 3...) of the processor to be monitored in the specified mode. If the parameter is omitted, ALL is assumed.

CPU=ALL

All processors in the system are to be monitored in the specified mode.

Example 1

Monitor instruction-processing-damage machine-check interruptions on processor 0. If seven of these interruptions occur in 600 seconds on processor 0, invoke ACR to take processor 0 offline.

```
mode pd,record=7,interval=600,cpu=0
```

Example 2

Record on the logrec data set all machine-check interruptions indicating invalid PSW or registers, but do not monitor them for any processor in the system.

```
MODE IV,CPU=ALL,RECORD=ALL
```

Example 3

Monitor the frequency of system damage machine-check interruptions on all processors, using the default values of five for the RECORD= parameter and 300 for the INTERVAL= parameter. After five system damage machine checks have occurred on a given processor within five minutes (300 seconds), invoke ACR to take that processor offline.

```
mode sd
```

Controlling the Recording of System Recovery and Degradation Machine Check Interruptions

You can use the MODE command to control the recording and reporting of system recovery and degradation machine check interruptions.

```

MODE {SR} [,QUIET
      {DG} |,RECORD[=nnn
                  ] [,CPU={x|ALL}]
                  |
                  |=ALL[,REPORT=nnn]
                  |=50
                  |=1

```

The parameters are:

SR

System recovery machine checks are to be placed in the specified recording mode.

DG

Degradation machine checks are to be placed in the specified recording mode.

QUIET

No machine check interruptions of the specified type (system recovery or degradation) are to occur or be recorded for the specified processor.

RECORD=nnn

After the specified number (1 to 999) of system recovery or degradation machine check interruptions occur on the specified processor, the system is to notify you and switch the recording mode to QUIET for that type of interruption on that processor. If you do not specify a number or omit the RECORD parameter, the system uses the following defaults:

- RECORD=1 for DG
- RECORD=50 for SR

RECORD=ALL

All system recovery or degradation machine check interruptions occurring on the specified processor are to be recorded on the logrec data set. The default number of interruptions is 1 for DG and 50 for SR unless you specify the REPORT parameter along with RECORD=ALL. You are notified each time the defined number of interruptions occur, (see the description of REPORT=nnn for more information) but the system does not switch to QUIET mode for that type of interruption.

REPORT=nnn

You are notified each time the specified number (1 to 999) of system recovery or degradation machine check interruptions occur on the specified processor. Use this parameter only with the RECORD=ALL parameter. If you omit the REPORT parameter, you will be notified each time the default number of interruptions occurs. The defaults are:

- REPORT=50 for SR
- REPORT=1 for DG

CPU=x

The address (0, 1, 2, 3,...) of the processor to be put in the specified mode. If the parameter is omitted, ALL is assumed.

CPU=ALL

All processors in the system are to be put in the specified mode.

Example 1

Degradation machine check interruptions are to be counted on processor 0. If the default number (1) occurs, the system notifies you and switches the recording mode

MODE Command

to QUIET for these interruptions. The other processor(s) in the system is not affected and no other types of machine check interruptions for processor 0 are affected.

```
MODE DG,CPU=0
```

Example 2

Degradation machine checks are to be put in QUIET mode on processor 2.

```
MODE DG,QUIET,CPU=2
```

Displaying Recording and Monitoring Status

You can use the following form of the MODE command to display the status of each type of machine check interruption.

```
MODE [STATUS]
```

STATUS

The event counters and recording/monitoring status associated with each type of machine check interruption are to be displayed for each processor. If the STATUS parameter is specified, it must be the only parameter specified.

MODIFY Command

Use the MODIFY command to pass information to a job or started task. Note that you can communicate with a currently running program only if it was designed to recognize input from the MODIFY command. If the program is not designed to accept input from the MODIFY command, MVS issues message IEE342I MODIFY REJECTED--TASK BUSY.

Note to Programmers: For more information, see the section on communicating with a program using EXTRACT and QEDIT in *z/OS MVS Programming: Authorized Assembler Services Guide*.

Summary of MODIFY

Table 4-22 shows examples of the tasks that the MODIFY command can perform. Use it to access the pages on which you can find details and examples of a particular task.

Table 4-22. Summary of the MODIFY Command

Topic:	Command:
“MODIFY Command”	F jobname.identifier,parameters
“Passing Information to a z/OS UNIX System Services Application” on page 4-313	F jobname.identifier,APPL=text
“Modifying TSO/VTAM Time Sharing” on page 4-314	F jobname.identifier,USERMAX=,USER=
“Controlling z/OS UNIX System Services (z/OS UNIX)” on page 4-314	F BPXOINIT,parameters
“Communicating with the Catalog Address Space” on page 4-321	F CATALOG,parameters
“Communicating with the Device Manager Address Space” on page 4-330	F DEVMAN,parameters
“Changing the DLF Processing Mode” on page 4-331	F DLF,MODE=
“Changing the DLF parameters” on page 4-331	F DLF,NN=
“Displaying DLF Status” on page 4-332	F DLF,STATUS
Syntax and parameters for the MODIFY hzsproc command in <i>IBM Health Checker for z/OS: User’s Guide</i> .	F hzsproc,parameters
“Building and Replacing Library Lookaside Directories” on page 4-332	F LLA,parameters
“Operating with the Network File System Server” on page 4-333	F MVSNFS
“Managing the Object Access Method (OAM)” on page 4-333	F OAM,parameters
“Recycling z/OS UNIX System Services (z/OS UNIX)” on page 4-333	F OMVS,parameters
“Dynamically activating maintenance for z/OS UNIX System Services (z/OS UNIX)” on page 4-335	F OMVS,parameters
“Stopping a physical file system (PFS) through a logical file system (LFS) interface” on page 4-336	F OMVS,STOPPFS=

MODIFY Command

Table 4-22. Summary of the MODIFY Command (continued)

Topic:	Command:
"Stopping a Temporary File System (TFS)" on page 4-336	F TFS,parameters
"Enabling and Disabling the Application Response Measurement (ARM) Agent and Enterprise Workload Manager (EWLM) platform services" on page 4-337	F WLM,AM=DISABLE ENABLE
"Changing Workload Manager Resource States" on page 4-337	F WLM,RESOURCE
"Specifying Data Set Selection Criteria for an External Writer" on page 4-338	F XWTR.identifier,devnum,parameters
"Causing an External Writer to Pause" on page 4-340	F XWTR.identifier,devnum,PAUSE

Using Asterisks in MODIFY Commands

Certain forms of the MODIFY command allow you to specify the following for a job or started task:

- An identifier
- An optional job name.

You can use the asterisk wildcard to direct the MODIFY command to more than one job or started task. The asterisk indicates that a MODIFY command applies to all jobs or started tasks that match a leading character string.

For example, when specifying an identifier without an optional job name, you can enter the following command to pass a two-digit value to all jobs with identifiers beginning with R1:

```
F R1*,00
```

You can also use the asterisk wildcard when you specify both a job name and identifier. For example you can enter the following command to pass a two-digit value to all jobs with names beginning with WX and identifiers beginning with R1:

```
F WX*.R1*,00
```

When you specify asterisks with device numbers, the system assumes that the device numbers are four digits long. For example, /13* would match on **1301**, **1302**, and so on, but would not match on **13C**.

Remember the following rules when using asterisk notation:

- If you specify only the *identifier* parameter (without the *jobname* parameter), you cannot specify a stand-alone asterisk on the *identifier* parameter.
- If you specify both the *jobname* and *identifier* parameters, you cannot specify a stand-alone asterisk for *both* parameters.

For example, to pass a two-digit value to all jobs with names beginning with WX, you can specify a single asterisk on the *identifier* to indicate a *wildcard*:

```
F WX*.*.,00
```

If you were to remove the WX characters from the above command, it would not be valid. You cannot specify *.* without a leading character string on the *jobname* parameter, *identifier* parameter, or both.

The following figures illustrate how asterisk notation works in MODIFY commands. Table 4-23 shows examples of START commands that are used to start jobs. Columns three and four show the associated jobnames and identifiers.

Table 4-23. Examples of START Commands to Start Jobs

Job Number	START Command	Jobname	Identifier
1	START YZ	YZ	YZ
2	START WX.YZ	WX	YZ
3	START WX.YZ1	WX	YZ1
4	START WX1.YZ1	WX1	YZ1
5	START WX, JOBNAME =WX1	WX1	WX1
6	START WX, JOBNAME =WX2	WX2	WX2
7	START WX, JOBNAME =YZ	YZ	YZ
8	START Q.YZ3	Q	YZ3
9	START WX.R1	WX	R1
10	START WX, JOBNAME =YZ4	YZ4	YZ4

Table 4-24 shows examples of MODIFY commands. The numbers in the second column indicate to which jobs in Table 4-23 each MODIFY command applies.

Table 4-24. Examples of MODIFY Commands

MODIFY Command	Affected Jobs
F WX.YZ,parameters	2
F WX.YZ*,parameters	2, 3
F YZ.*,parameters	1, 7
F WX*,parameters	5, 6
F YZ*,parameters	1, 2, 3, 4, 7, 8, 10
F WX*.YZ,parameters	2
F WX*.YZ*,parameters	2, 3, 4
F *.YZ*,parameters	1, 2, 3, 4, 7, 8, 10
F *.YZ,parameters	1, 2, 7
F WX*.*,parameters	2, 3, 4, 5, 6, 9
F WX.*,parameters	2, 3, 9

MODIFY Command Syntax

The syntax for each variation of the MODIFY command is shown immediately preceding its respective parameter list.

MODIFY or F

Syntax

F [jobname.]identifier,parameters

MODIFY Command

Parameters

The parameters for this command are:

jobname

The name of the batch job, started task, or APPC/MVS transaction program to be modified.

The job name for a given started task can be assigned based on a variety of inputs. These inputs are examined in the following order, so that if item #1 is not specified, item #2 is used. If neither #1 nor #2 is specified, then #3 is used, and so on.

1. The jobname specified in the JOBNAME= parameter of the START command
or
The identifier specified on the START command.
2. The jobname specified on the JOB JCL statement within the member.
3. The device number specified on the START command, or the device number associated with the device type specified on the START command
or
The device number associated with the device type specified on the START command.
4. The device number associated with the IEFRDER DD statement within the member.
5. The member name.

You can use asterisk notation to specify more than one job or started task on the MODIFY command. See “Using Asterisks in MODIFY Commands” on page 4-310 for more information.

identifier

The identifier assigned to the job or started task. (Refer to “Displaying Started Task Status” on page 4-157 for information about determining the jobname and identifier of currently active started tasks.)

Use one of the following types of identifiers:

- The identifier that was specified on the START command.
- */devnum*, the 3-digit or 4-digit hexadecimal device number specified when the START or MOUNT command was issued.
When you specify a device number that could be mistaken for the device name, precede the device number with a slash. The slash is optional with a 3-digit device number.
- *devicetype*, the type of device specified when the START or MOUNT command was issued.

If no identifier was specified, the identifier “STARTING” is temporarily assigned until the system can assign another according to the following order of precedence:

1. If an identifier was not specified on the START command, the identifier is the device type (for example, 3410) or device number (for example, X'0000') specified on the START or MOUNT command.
2. If an identifier, a device type, or a device number was not specified on the START or MOUNT command, the identifier is the device type specified on an IEFRDER DD statement (invoking a cataloged procedure) in the JCL.

3. If none of the above was specified, the identifier defaults to the job name.

You can use asterisk notation to specify more than one job or started task on the MODIFY command. See “Using Asterisks in MODIFY Commands” on page 4-310 for more information.

parameters

Program parameters passed to the started program.

Example 1

If the program currently running in job TMASGX02 was set up by the programmer to accept 2-digit values from the operator, enter the following (where nn is an appropriate 2-digit value):

```
f tmasgx02,nn
```

Example 2

If started tasks are running with the following jobnames and identifiers:

```
Jobname = ABC1, identifier = DEF1  
Jobname = ABC2, identifier = DEF2
```

And both tasks accept the value “INPUT1” from the operator, enter:

```
f ABC*.DEF*,INPUT1
```

Both tasks receive the value INPUT1.

Passing Information to a z/OS UNIX System Services Application

```
F [jobname.]identifier,APPL=text
```

The parameters are:

jobname

The name of the job.

You can use asterisk notation to specify more than one job or started task on the MODIFY command. See “Using Asterisks in MODIFY Commands” on page 4-310 for more information.

identifier

The identifier used on the START command to identify the application.

If an identifier was not specified on the START command, the system automatically uses the job name as the identifier. (Refer to “Displaying Started Task Status” on page 4-157 for information about determining the jobname and identifier of currently active started tasks.)

You can use asterisk notation to specify more than one job or started task on the MODIFY command. See “Using Asterisks in MODIFY Commands” on page 4-310 for more information.

APPL=text

Up to 110 characters. Quotes around the text are optional. If you do put the text in quotes, the quotes will be passed, with the text, back to the application.

Example

MODIFY Command

To modify the jobname DATASRVR, enter:

```
F DATASRVR,APPL=threadlimit=5
```

Jobnames can be a maximum of eight characters. No spaces are allowed.

Modifying TSO/VTAM Time Sharing

You can use the MODIFY command to control the number of users allowed to be logged on to TSO/VTAM and to terminate TSO user address spaces.

```
F [jobname.]identifier,{USERMAX=nnnnn}
    {USER={SIC   }  }
        {FSTOP} }
```

The parameters are:

jobname

The name of the job. Many installations use TCAS as the name.

You can use asterisk notation to specify more than one job or started task on the MODIFY command. See “Using Asterisks in MODIFY Commands” on page 4-310 for more information.

identifier

The identifier specified on the START command for TSO/VTAM time-sharing.

If an identifier was not specified on the START command, the system automatically assigns the job name as the identifier. (Refer to “Displaying Started Task Status” on page 4-157 for information about determining the jobname and identifier of currently active started tasks.)

You can use asterisk notation to specify more than one job or started task on the MODIFY command. See “Using Asterisks in MODIFY Commands” on page 4-310 for more information.

USERMAX=nnnnn

The maximum number (0 to 32,767) of users that can be logged on to TSO/VTAM time-sharing at one time. Note that specifying USERMAX=0 causes the terminal control address space (TCAS) to suppress all LOGONs.

USER=SIC

Causes the TCAS to cancel all TSO/VTAM terminal user address spaces normally. The terminal users receive any messages queued for them. The TCAS remains active.

USER=FSTOP

Forces the TCAS to cancel all TSO/VTAM terminal user address spaces immediately. The terminal users do not receive any messages queued for them. The TCAS remains active. Specify FSTOP only if a system problem causes SIC to be ineffective.

If you issue the MODIFY command with the USER=FSTOP parameter, the affected address space is deleted from the system and recovery is severely limited.

Controlling z/OS UNIX System Services (z/OS UNIX)

You can use the MODIFY command to control z/OS UNIX System Services and to terminate a z/OS UNIX process or thread. You can also use it to shut down z/OS

UNIX initiators and to request a SYSMDUMP for a process.

```
F BPXOINIT,{APPL=appl_data}
  {DUMP=pid}
    {FILESYS={DISPLAY[,FILESYSTEM=filesystemname] [, OVERRIDE]}
     |,ALL
     |,EXCEPTION
     |,GLOBAL
      {DUMP } }
      {FIX } }
      {REINIT } }
      {RESYNC } }
      {UNMOUNT,FILESYSTEM=filesystemname } }
      {UNMOUNTALL } }

  {FORCE=pid[.tid]}
  {RESTART=FORKS}
  {RECOVER=LATCHES}
  {SHUTDOWN={FILEOWNER | FILESYS | FORKINIT | FORKS}}
  {SUPERKILL=pid}
  {TERM=pid[.tid]}
```

The parameters are:

BPXOINIT

The name of the job.

APPL=app_data

Allows information to pass straight through to the application. *app_data* is a string that is passed back to the invoker in whatever format the application expects it.

Note: BPXOINIT does not accept any APPL= parm values. You will receive the error message BPXM029I APPL= KEYWORD WAS IGNORED BY BPXOINIT.

DUMP=pid

Requests a SYSMDUMP. A SIGDUMP signal is sent to the specified process. *pid* is the decimal form of the process id to be terminated.

FILESYS=

Indicates that a file system diagnostic or recovery operation is to be performed.

This function is applicable only to a sysplex environment where shared file system has been enabled by specifying SYSPLEX(YES) in the BPXPRMxx parmlib member named during system initialization. The command is intended to help diagnose and correct certain shared file system problems or errors that impact one or more systems in a sysplex environment.

Use this command with caution, and only under the direction of an IBM service representative.

To obtain the best results, issue this command at the system with the highest shared file system software service level. To determine which system is executing with the highest shared file system software service level, issue the command

```
F BPXOINIT,FILESYS=DISPLAY,GLOBAL
```

and select the system with the highest "LFS Version" value.

Specify one of the following functions:

MODIFY Command

DISPLAY or D

Display the type BPXMCDS couple data set information relating to the shared file system file system. **D** is an alias of **DISPLAY**.

Specify one of the following display options:

ALL

Displays all file systems in the shared file system hierarchy.

EXCEPTION

Displays all file systems that are in an exception state. A file system is in an exception state if one of the following criteria is met:

- State = Mount in progress
- State = Unmount in progress
- State = Quiesce in progress
- State = Quiesced
- State = Unowned
- State = In recovery
- State = Unusable
- The file system state in the couple data set representation is inconsistent with the local file system.

FILESYSTEM=*filesystemname*

Displays information for the specified file system.

GLOBAL

Displays the current sysplex state, consisting of the following items:

- The active systems in the sysplex (system name, logical file system (LFS) version, verification status, recommended recovery action).
- The type BPXMCDS couple data set version number.
- The minimum LFS version required to enter the BPXGRP sysplex group.
- The name of the system serving BRLM.
- The device number of the last mounted file system.
- The maximum and in-use mounts.
- The maximum and in-use AMTRULES.
- Whether or not DISTBRLM (distributed BRLM) is enabled.
- Whether or not DISTBRLM is active.
- The active "serialization categories," which systems are associated with each category, and the time that each "serialization category" was first started. The following serialization categories are defined:
 - SYSTEMS PERFORMING INITIALIZATION
 - SYSTEMS PERFORMING MOVE
 - SYSTEMS PERFORMING QUIESCE
 - SYSTEMS PERFORMING UNMOUNT
 - SYSTEMS PERFORMING MOUNT RESYNC
 - SYSTEMS PERFORMING LOCAL FILE SYSTEM RECOVERY
 - SYSTEMS PERFORMING FILE SYSTEM TAKEOVER RECOVERY
 - SYSTEMS RECOVERING UNOWNED FILE SYSTEMS
 - SYSTEMS PERFORMING REPAIR UNMOUNT

GLOBAL is the default display option.

DUMP

Initiate an SVC dump to capture all of the file system sub-records in the active type BPXMCDS couple data set.

FIX

Perform automatic file system and couple data set diagnosis and repair. As a part of the file system analysis, the system performs an analysis of possible file system latch contention on each system in the sysplex. An operator message identifies any possible problems. The system also analyzes file system serialization data that is maintained in the couple data set, and corrects it if an error is detected. It reports the status of the analysis in an operator message.

Note that the system initiates a dump of critical file system resources as a part of the FIX function. The dump is captured prior to the diagnosis and repair. If, however, a dump was captured due to a FIX or DUMP function that was initiated within the previous 15 minutes, the dump is suppressed.

Perform FIX prior to the UNMOUNTALL and REINIT functions.

REINIT

Re-initialize the file system hierarchy based on the ROOT and MOUNT statements in the BPXPRMxx parmlib member used by each system during its initialization. (Any changes to the BPXPRMxx parmlib member that are made after the system's initialization are not included in the REINIT processing. The system uses the version of the file system parmlib statements that is maintained in kernel storage. It does not re-process the parmlib member.)

Note that the system where the MODIFY command is issued will become the file system server to those file systems common to all systems in the sysplex (such as the ROOT file system) unless the SYSNAME() parameter is specified on the parmlib MOUNT statement.

The intended use of this function is to re-initialize the file system hierarchy after an **UNMOUNTALL** has been performed. However, you can issue **REINIT** at any time; those file systems that are already mounted will not be impacted when **REINIT** processes the parmlib mount statements.

REINIT is not applicable to MKDIR support in the BPXPRMxx parmlib member. The directory mountpoints are not available and thus cause successive mounts to fail.

Always issue the FIX function before performing the REINIT function.

RESYNC

Perform a file system hierarchy check on all systems. If a system has not mounted a file system that is active in the shared file system hierarchy, it is mounted locally and thus made available to local applications.

UNMOUNT

Unmount the file system specified by the *filesystem*= parameter. The file system cannot have any active mount points for other file systems. You must unmount those file systems first.

UNMOUNTALL

Unmount all file systems in the sysplex file system hierarchy, including the root file system. When processing is complete, mount SYSROOT on all systems.

MODIFY Command

Always issue the FIX function before performing the UNMOUNTALL function.

OVERRIDE

Normally only one MODIFY command for a FILESYS= function can be active on each system. Additionally, only one instance of the MODIFY command *in the sysplex* can be active for the **FIX**, **UNMOUNT**, **UNMOUNTALL**, and **REINIT** functions. If you specify the **OVERRIDE** parameter, the system accepts multiple invocations of this command on each system for the **DISPLAY**, **DUMP**, and **RESYNC** functions. Note, however, that the second invocation may be delayed.

The primary intent of the **OVERRIDE** parameter is to allow issuance of the **DISPLAY** functions while there is still a MODIFY in progress and the MODIFY appears to be delayed.

FORCE=

Indicates that the signal interface routine cannot receive control before the thread is terminated.

pid.tid

pid is the decimal form of the process id to be terminated. *tid* is the hexadecimal form of the thread id to be terminated.

RESTART=FORKS

Enables the system to resume normal processing. Suspended dub requests are resumed.

RECOVER=LATCHES

This command ends user tasks that are holding latches for an excessive amount of time. You can enter this command manually or provide an automation script allowing the system to automatically respond to message BPXM056E to aid in resolution of excessive latch contention. The command is primarily intended as an aid in resolving latch hangs that are caused by user task usage of UNIX System Services. However, it might not be able to resolve latch hangs caused by internal system tasks in the OMVS kernel address space, if the owning system task is in critical system code that cannot be interrupted.

Notes:

1. Only use this command if message BPXM056E is outstanding. The command causes one of the following:
 - If the contention can be resolved, the system DOMs the BPXM057E message.
 - If the contention cannot be resolved, the system issues message BPXM057 to indicate that condition.
2. MVS isolates the abnormal termination to individual tasks, but this command can result in the termination of an entire process. It is important to note that the abnormal termination will be caused by a non-retryable 422-1A5 abend that will cause the generation of a system dump, because of the likelihood of an internal system problem. Additionally, if more than one latch is in contention, multiple tasks might be abended and result in requests for multiple dumps.

SHUTDOWN=FILEOWNER

Unmounts the UNIX System Services file systems. Also prevents the system from becoming a filesystem owner through a move or recovery operation until z/OS UNIX System Services is recycled.

SHUTDOWN=FILESYS

Unmounts the UNIX System Services file systems.

SHUTDOWN=FORKINIT

Shuts down the z/OS UNIX initiators. Normally, these initiators shut themselves down in 30 minutes. Attempts to purge JES2 (command= P JES2) cannot complete until z/OS UNIX initiators have shut down.

SHUTDOWN=FORKS

Requests a shutdown of the fork() service by preventing future forks and non-local spawns. The kernel cannot obtain additional WLM fork initiators for fork and spawn. It attempts to terminate all WLM fork initiator address spaces that are running processes created by fork or non-local spawn. All other services remain "up", but any new dub requests are suspended until the fork() service is restarted.

SUPERKILL=pid

Indicates that a terminating signal is sent to the target process.

Guideline: SUPERKILL=pid ends the entire process and any subprocesses within the address space. Because SUPERKILL=pid is a stronger form of the TERM= and FORCE= parameters, only use this command if you are not able to end the process using F BPXOINIT TERM= and FORCE= commands.

pid

pid is the decimal form of the process ID to be ended.

TERM=

Indicates that the signal interface routine can receive control before the thread is terminated.

pid.tid

pid is the decimal form of the process id to be terminated. *tid* is the hexadecimal form of the thread id to be terminated.

Example 1

To display process information for a process id of '117440514' enter:

```
DISPLAY OMVS,pid=117440514
BPX0040I 14.16.58 DISPLAY OMVS 177
OMVS      000E ACTIVE
USER      JOBNAME   ASID      PID      PPID STATE    START     CT_SECS
MEGA      TC1        0021    117440514  117440515 HKI     14.16.14    .170
LATCHWAITPID=          0 CMD=ACEECACH
THREAD_ID   TCB@      PRI_JOB  USERNAME   ACC_TIME SC  STATE
0496146000000000  009E0438 OMVS           .050 PTJ  KU
04961D0800000001  009D5E88 OMVS           .002 SLP  JSN
049625B000000002  009D8798 OMVS           .003 SLP  JSN
04962E5800000003  009D5090 OMVS           .012 SLP  JSN
0496370000000004  009D5228 OMVS           .011 SLP  JSN
04963FA800000005  009D5A88 OMVS           .010 SLP  JSN
0496485000000006  009D8048 OMVS           .011 SLP  JSN
049650F800000007  009D81E0 OMVS           .011 SLP  JSN
049659A000000008  009D8378 OMVS           .011 SLP  JSN
0496624800000009  009D8510 OMVS           .011 SLP  JSN
04966AF00000000A  009D8930 OMVS           .030 SLP  JSN

f bpxoinit,force=117440514.04962E5800000003
BPXM027I COMMAND ACCEPTED.

f bpxoinit,term=117440514.0496624800000009
BPXM027I COMMAND ACCEPTED.
```

MODIFY Command

Example 2

To shut down the fork() service, enter:

```
F BPXOINIT,SHUTDOWN=FORKS  
BPXIxxxE FORK SERVICE HAS BEEN SHUTDOWN SUCCESSFULLY. ISSUE F  
BPXOINIT,RESTART=FORKS TO RESTART FORK SERVICE.
```

Example 3

To restart the fork() service, enter:

```
F BPXOINIT,RESTART=FORKS
```

Example 4

Sample outputs of the MODIFY BPXOINIT,FILESYS command:

- F BPXOINIT,FILESYS=DISPLAY,GLOBAL

```
SY1 BPXM027I COMMAND ACCEPTED.  
SY1 BPXF040I MODIFY BPXOINIT,FILESYS PROCESSING IS COMPLETE.  
SY1 BPXF041I 2004/08/10 14.10.09 MODIFY BPXOINIT,FILESYS=DISPLAY,GLOBAL  
SYSTEM   LFS VERSION ---STATUS----- RECOMMENDED ACTION  
Z0      1. 6. 0 VERIFIED          NONE  
JA0     1. 6. 0 VERIFIED          NONE  
TPN     1. 6. 0 VERIFIED          NONE  
Z1      1. 6. 0 VERIFIED          NONE  
J90     1. 6. 0 VERIFIED          NONE  
JF0     1. 6. 0 VERIFIED          NONE  
JB0     1. 6. 0 VERIFIED          NONE  
JC0     1. 6. 0 VERIFIED          NONE  
JE0     1. 6. 0 VERIFIED          NONE  
Z2      1. 6. 0 VERIFIED          NONE  
Z3      1. 6. 0 VERIFIED          NONE  
J80     1. 6. 0 VERIFIED          NONE  
JG0     1. 6. 0 VERIFIED          NONE  
JH0     1. 6. 0 VERIFIED          NONE  
CDS VERSION= 2      MIN LFS VERSION= 1. 6. 0  
BRLM SERVER=N/A      DEVICE NUMBER OF LAST MOUNT= 9266  
MAXIMUM MOUNT ENTRIES= 800  MOUNT ENTRIES IN USE= 699  
MAXIMUM AMTRULES= 51  AMTRULES IN USE= 9  
DISTBRLM ENABLED=YES      DISTBRLM ACTIVE=YES
```

- F BPXOINIT,FILESYS=DISPLAY,FILESYSTEM=POSIX.SY4.HFS

```
SY1 BPXM027I COMMAND ACCEPTED.  
SY1 BPXF035I 2000/05/12 11.55.34 MODIFY BPXOINIT,FILESYS=DISPLAY  
-----NAME----- DEVICE MODE  
POSIX.SY4.HFS           23  RDWR  
  PATH=/SY4  
  PARM=SYNC(04)  
  STATUS=ACTIVE          LOCAL STATUS=ACTIVE  
  OWNER=SY1             RECOVERY OWNER=SY1    AUTOMOVE=Y PFSMOVE=Y  
  TYPENAME=HFS           MOUNTPOINT DEVICE= 12  
  MOUNTPOINT FILESYSTEM=POSIX.SYSPLEX9.HFS1  
  ENTRY FLAGS=90000000  FLAGS=40000000  LFSFLAGS=08000000  
  LOCAL FLAGS=40000000  LOCAL LFSFLAGS=2A000000  
SY1 BPXF040I MODIFY BPXOINIT,FILESYS PROCESSING IS COMPLETE.
```

- F BPXOINIT,FILESYS=DISPLAY,FILESYSTEM=POSIX.ZFS.ETC

```
SY1 BPXM027I COMMAND ACCEPTED.  
SY1 BPXF035I 2000/05/12 11.55.34 MODIFY BPXOINIT,FILESYS=DISPLAY  
-----NAME----- DEVICE MODE  
POSIX.ZFS.ETC           23  RDWR
```

```

AGGREGATE NAME=POSIX.ZFS.ETC
PATH=/SY1/etc
PARM=SYNC(04)
STATUS=ACTIVE LOCAL STATUS=ACTIVE
OWNER=SY1 RECOVERY OWNER=SY1 AUTOMOVE=Y PFSMOVE=Y
TYPENAME=ZFS MOUNTPOINT DEVICE= 12
MOUNTPOINT FILESYSTEM=POSIX.SYSPLEX9.ZFS1
AGGREGATE=POSIX.ZFS.ETC
ENTRY FLAGS=90000000 FLAGS=40000000 LFSFLAGS=08000000
LOCAL FLAGS=40000000 LOCAL LFSFLAGS=2A000000
SY1 BPXF040I MODIFY BPXINIT,FILESYS PROCESSING IS COMPLETE.

```

For zFS file systems, the display includes an aggregate file system name, indicating membership in a data set containing multiple file systems. Aggregates provide member file systems with a common pool of disk space.

Example 5

In response to message BPXM056E UNIX SYSTEM SERVICES LATCH CONTENTION DETECTED, enter the MODIFY or F BPXINIT,RECOVER=LATCHES command to resolve the contention. You will receive one of the followings in response to the command:

- f bpxoinit,recover=latches
BPXM067I UNIX SYSTEM SERVICES LATCH CONTENTION RESOLVED
- f bpxoinit,recover=latches
BPXM057E UNIX SYSTEM SERVICES LATCH CONTENTION NOT RESOLVING

Communicating with the Catalog Address Space

Use the MODIFY CATALOG command to communicate with the catalog address space to display information or to request a specified service. **Use this command only at the direction of the system programmer.**

When an operator issues any MODIFY CATALOG command, messages return to that console exclusively with one exception: for a MODIFY CATALOG,RESTART command, messages return to the console issuing the command and any consoles receiving routing code 2.

MODIFY Command

```
F CATALOG,{ABEND{(id)} [,FORCE]
          {(yyyyyyyy)}
          {(ALLOCATE)}
          {(ANALYSIS)}
          {(MODIFY) }
          {ALIASLEVEL(n)}
          {ALLOCATE(catname)[,{NOISC | NOVLF}]}
          {ALLOCATED[(vvvvv)]}
          {CATMAX(nnnn)}
          {CLOSE(catname)}
          {DISABLE(feature) }
          {{DUMPON | DUMPOFF}}
          {ECSHR(value)}
          {ENABLE(feature)}
          {END(id)[,{REDRIVE|NOREDRIVE}]
            [,FORCE]
          }
          {ENTRY[(cname) | (mmmmmmmm)]}
          {{ISC | NOISC}(catname)}
          {LIST[(id) | (yyyyyyyy)]}
          {LISTJ[jobname]}
          {MLA(value)}
          {NOTIFYEXTENT(xxx)}
          {REPORT}
          {REPORT,CACHE[(catname)]}
          {REPORT,CATSTATS[(catname)]}
          {REPORT,DUMP}
          {REPORT,PERFORMANCE[(RESET)]}
          {RESET,CATSTATS[(catname)]}
          {RESTART}
          {{ROTATE | NORotate}}
          {{SYS%ON | SYS%OFF}}
          {TAKEDUMP}
          {TASKMAX(nnn)}
          {UNALLOCATE[(catname)]}
          {VCLOSE(vvvvvv)}
          {{VLF | NOVLF}(catname)}
          {{VDUMPON | VDUMPOFF}}
          {{VUNALLOCATE | NOVUNALLOCATE}}
          {VVDSSPACE(primary,secondary)}
```

The following are brief descriptions of the parameters. **For complete descriptions, including under which circumstances to use and when not to use this command, see z/OS DFSMS Managing Catalogs.**

ABEND

Terminates a catalog address space (CAS) task.

(*id*) [,FORCE]

The task ID of the catalog address space service task. If *id* is specified, you can specify the optional FORCE parameter:

FORCE

To end an active service task, even if the service task is in recall.

Restriction: Only use FORCE when the address space or task that the service task is operating on behalf of has ended abnormally.

(yyyyyyyy)

The full four-byte address of the TCB for the catalog address space service task, in hexadecimal.

Note: ABEND(*id*) or ABEND(yyyyyyyy) terminates the catalog address space with an abnormal termination of the service task with an abend code of 91A and requests an SVC dump. Any catalog request in process at the time of the abend is redriven one time. (See the MODIFY CATALOG,END command format for unlimited redrive capability.) When the task id is known, the END command format is the method IBM recommends for terminating catalog address space service task processing.

(ALLOCATE)

Causes the catalog address space allocation task to terminate and a new catalog address space allocation task to be attached.

(ANALYSIS)

Causes the catalog address space analysis task to terminate and a new catalog address space analysis task to be attached.

(MODIFY)

Causes the catalog address space modify task to terminate and a new catalog address space modify task to be attached.

ALIASLEVEL(*n*)

Changes the number of alias levels of qualification used by multilevel alias logic in a catalog search. The value of '*n*' can be 1 through 4. Alias levels can also be set automatically at IPL via the SYSCATnn member of SYS1.NUCLEUS or via a SYSCAT statement in IPLPARM.

ALLOCATE(*catname*)[,{NOISC | NOVLF}]

Allocates an integrated catalog facility catalog to the catalog address space.

catname specifies the integrated catalog facility catalog name.

NOISC

Deactivates the in-storage catalog option. (See ISC.)

NOVLF

Deactivates the catalog data space cache for a currently allocated and integrated catalog facility catalog.

ALLOCATED[(*vvvvv*)]

Lists the name, volume serial number, current allocation count, and various status flags for every catalog currently allocated on the system. Message IEC348I displays the information.

vvvvv specifies the volser that can be used to limit the list to allocated catalogs on a specific volume. Use *vvvvv* to specify the volume serial number.

Note: This form is synonymous with the OPEN(*vvvvv*) request, but should be used instead of that form.

CATMAX(*nnnn*)

Specifies the maximum number of catalogs that can be opened concurrently in CAS. When the limit is exceeded, the least recently accessed catalog is closed, freeing the CAS storage it had occupied. Closed catalogs are not unallocated. They remain allocated, but in restart status with no CAS storage. If the new limit is less than the previous limit, all currently open catalogs are closed.

The minimum value is 1 and the maximum value is 9999.

The number specified for *nnnn* is in decimal.

CLOSE(*catname*)

Closes an integrated catalog facility catalog dynamically.

MODIFY Command

All of the catalog address-space private storage associated with the catalog is freed. The catalog is reopened with a new set of control blocks the next time a request is processed for that catalog.

catname is the name of the integrated catalog facility catalog.

DISABLE(*feature*)

Disables a particular optional feature, where *feature* is one of the following:

AUTOTUNING

Disables automatic tuning by the catalog address space of the number of data buffers, index buffers and VSAM strings on the current system. See **ENABLE(AUTOTUNING)**.

DELFORCEWNG

Disables the warning message IDC1997I or IDC1998I when attempting to use the **DELETE VVDS RECOVERY** or **DELETE USERCATALOG FORCE** command.

DSNCHECK

Disables syntax checking on names being added to a catalog.

SYMREC

Specifies that SYMREC records are not to be created. Use this option to temporarily disable the creation of SYMREC records. For example, if a problem is causing repeated creation of SYMREC records and this is disrupting how well you are able to manage the SYMREC target data set, you can disable the SYMREC records.

UPDTFAIL

Disables the message IEC390I when a VSAM update request against a catalog has been abnormally terminated. This message is intended to alert the installation that potential catalog damage might result from the incomplete request.

Use this option if you identify that an excessive number of requests are abending. The default for this option is enabled. Disabling this option will prevent message IEC390I from being written if the condition is detected. IEC390I messages that have been written to the console remain unchanged. The state of the UPDTFAIL setting persists across a catalog restart, but not an IPL.

VVRCHECK

Disables enhanced VVR checking on VVDS I/O.

DUMPPON

DUMPOFF

Activates or deactivates dynamic dumping by the catalog address space. The default, DUMPOFF, is set at IPL.

ECSHR(*value*)

Specifies changes for enhanced catalog sharing (ECS) mode, where *value* is one of the following:

AUTOADD

Indicates whether or not the system is to add ECS-eligible catalogs automatically to the ECS structure on the next reference to the catalog. (See the ADD subparameter description earlier for the conditions that make a catalog ECS-eligible.)

Note: Catalogs that are ECS-eligible will not automatically be added to the ECS structure if (1) they have been explicitly removed from the ECS

structure with the ECSHR(REMOVE,catname) parameter of the MODIFY CATALOG command, or (2) their last accessor was a non-ECS system. You must add those catalogs explicitly using the ECSHR(ADD,catname) parameter of the MODIFY CATALOG command.

CONNECT

Connects the system to the enhanced catalog sharing (ECS) structure in the coupling facility (CF). The catalog address space (CAS) issues an IXLCONN request to allocate and connect to the cache structure in the CF or to connect to the already allocated structure.

DISCONNECT

The system disconnects from the ECS structure in the CF. The CAS issues an IXLDISC request. The names of any ECS catalogs in the CF are removed and the non-ECS sharing method will subsequently be used for the catalogs.

ENABLE,*catname*

ENABLE causes the named catalog (*catname*) that is temporarily ineligible to be enabled to the ECS structure if all the following conditions are true:

- The catalog has the ECSHARING attribute
- The catalog has shareoptions(3 4)
- The catalog resides on a volume that is defined as shared
- The system is connected to the ECS structure

If any of the above conditions are not true, the request is rejected. The issuer of this command should ensure that all current or potential sharers of the catalog are capable of ECS. Otherwise, manual intervention might be required to return to the VVDS sharing protocol.

ENABLEALL

Causes all temporarily ineligible catalogs known to CAS to be enabled for ECS.

REMOVE,*catname*

Removes the named catalog from the ECS structure in the coupling facility. Once you use REMOVE to remove the named catalog explicitly, you must use the MODIFY CATALOG command with the ECSHR(ADD,catname) parameter to put this catalog back into the ECS structure; the catalog will not be added automatically (even if AUTOADD is enabled).

STATUS

Displays (in message IEC380I) the status of the CF connection and the status of each catalog referenced since the last IPL.

STATUS,*catname*

Displays (in message IEC380I) the status of the coupling facility connection and the ECS status of the named catalog.

ENABLE(*feature*)

Enables a particular optional feature, where *feature* is one of the following:

AUTOTUNING

Enables the catalog address space to automatically tune the number of data buffers, index buffers, and VSAM strings on the current system to improve the performance of catalogs.

DELFORCEWNG

Enables issuance of messages IDC1997I and IDC1998I when a DELETE VVDS RECOVERY or DELETE USERCATALOG FORCE are performed.

MODIFY Command

DSNCHECK

Enables syntax checking of data set names being added to a catalog.

SYMREC

Specifies that SYMREC records are to be created. Use this option to reset the default value if you have disabled the creation of SYMREC records using the MODIFY CATALOG, DISABLE(SYMREC) command.

UPDTFAIL

Enables the warning messages, which is the default. The setting may be changed at any time, and the current setting is retained across Catalog Address Space restarts. It is not preserved across IPLs.

VVRCHECK

Enables enhanced VVDS record validation during VVDS I/O.

END(*id*)

This is the option IBM recommends to terminate a CAS service task.

id Specifies the unique task identification.

REDRIVE

The catalog request currently being processed will continue to be processed by a different task. REDRIVE will free up an ENQ lockout.

NOREDRIVE

Any catalog requests in process at the time will fail with a catalog return code of RC246.

FORCE

Specifies the abnormal ending of an active service task, even if it is in recall.

Restriction: Do not use FORCE unless the address space or task that the service task is operating on behalf of has ended abnormally.

ENTRY [(*cname*)]

[(*mmmmmmmm*)]

The catalog address space is to display the entry point of the CSECT with the name *cname* and its maintenance level. *cname* can be any CSECT name in load module IGG0CLX0 or IGG0CLHA. If you omit the name, the system displays the entry points of all CSECTs in the two load modules and their maintenance levels.

Provides the starting addresses, the FMIDs, and the PTF/APAR levels of all the modules resident in the catalog address space and resident in the link pack area. Message IEC349I displays the information.

mmmmmmmm is the csect name. If this name is omitted, *all* csects are listed.

The output of this command is probably best viewed on the system log, because of the large output size when all entry points are requested.

ISC(*catname*)

Activates the in-storage catalog for a currently allocated integrated catalog facility catalog. This action means that accessed catalog records are retained in storage, which can improve performance. Integrated catalog facility catalogs normally run with ISC active. You would issue this command only when the in-storage catalog for the named catalog has previously been deactivated with NOISC.

catname is the integrated catalog facility name.

LIST (id) or LIST (yyyyyyyy) or LISTJ (jobname)

Lists currently active catalog address space service tasks, their related jobnames, their elapsed times, and their unique IDs. Message IEC347I displays the information.

Specify a task id [LIST(*id*)] or TCB address [LIST(yyyyyyyy)] to list the information for a single service task.

Specify the jobname [LISTJ(*jobname*)] to list the information for all service tasks currently active for the specified jobname.

MLA(*value*)

Allows the operator to selectively enable, disable, or rebuild the multi-level alias facility control blocks. This can be necessary if the MLA has disabled itself, which is usually indicated by one of the messages IEC369I, IEC370I, IEC374I, or IEC375I.

DISABLE

Disable MLA processing. When the MLA is disabled, generic searches will fail with return code 194 and reason code 8.

ENABLE

Re-enable and rebuild the MLA control blocks. This may be used to counteract a previous DISABLE command or to activate the MLA after a previous failure during its initialization.

REBUILD

Forces a complete rebuild of the MLA control blocks. This can be used when an error is suspected in the MLA structure, but the MLA logic detects no condition that it considers a trigger for a rebuild.

NOROTATE

Deactivates rotation through the CRT table when processing a catalog request. Rotation is intended to delay as long as possible the reuse of catalog address space control blocks so they might be available if needed for diagnosing a problem. However, using a different set of control blocks for each successive catalog request might cause an excessive number of page faults. NOROTATE indicates that the catalog is to assign the first available control blocks to a request. Reusing the same control blocks more often might reduce page faults in some environments.

The default, NOROTATE, is set at IPL.

NOTIFYEXTENT(*xxx*)

Indicates the percentage of the maximum extents possible for a catalog that are currently allocated. *xxx* is a percentage number from 0-99. (You can omit leading zeros.) A percentage value of zero indicates that normal monitoring is suppressed, and is the default. The setting specified is retained across catalog restarts, but not IPLs. If the allocated extent threshold exceeds the given threshold for any catalog, the system will issue message IEC361I for that catalog. If a catalog exceeds 90% utilization of the maximum extents, the system will issue message IEC361I even if the threshold has been set to zero (that is, no normal monitoring).

REPORT

Provides basic information about some of the current limits and installation-specified defaults that are selected for the catalog address space.

REPORT,CACHE[(*catname*)]

This option lists general information on catalog cache status for all catalogs currently active in the catalog address space. The report generated shows

MODIFY Command

information useful in evaluating the catalog cache performance for the listed catalogs. If you specify a catalog name (*catname*), the report will show performance information only for the specified catalog.

REPORT,CATSTATS[(*catname*)]

This option lists I/O access statistics, BUFNI, BUFND, and STRNO information for all catalogs currently active in the catalog address space. If you specify a catalog name (*catname*), the statistics are listed only for the specified catalog.

REPORT,DUMP

This option calls for a display of the current dump status of the catalog address space. (Dump status can be enabled or disabled by the DUMPON and DUMPOFF forms of the MODIFY command.)

REPORT,PERFORMANCE[(RESET)]

This option produces a report of the performance of events in the catalog address space that invoke code outside of the catalog component. The display is a multi-line WTO showing the total number of occurrences of each event and the average time spent completing that event.

RESET, if specified, sets the performance reporting information to zero.

RESET,CATSTATS[*catname*]

Resets the I/O statistics for all catalogs currently active in the catalog address space. If you specify a catalog name (*catname*), the statistics are reset only for the specified catalog.

You can use this command to reset the I/O statistics and determine how much activity the specified catalog gets during a given time period. To see the current I/O statistics report, use the F CATALOG,REPORT,CATSTATS option.

RESTART

Terminates the catalog address space with an abend code of 81A. The termination is transparent to the user; the system restarts the catalog address space in a new address space and honors outstanding requests to the catalog address space.

ROTATE

Reactivates rotation through the CRT table when processing a catalog request. Rotation is intended to delay as long as possible the reuse of catalog address space control blocks so that the control blocks might be available if needed to diagnose a problem, at the potential cost of increased page faults.

NOROTATE deactivates rotation.

SYS%ON

SYS%OFF

Activates or deactivates the conversion of SYS% to SYS1. Conversion can also be set in the SYSCATnn member of SYS1.NUCLEUS or in a SYSCAT statement in IPLPARM.

TAKEDUMP

Causes the Catalog Address Space to issue an SVCDUMP using the proper dump options to ensure all of the data needed to diagnose catalog problems is available. This eliminates the need for a user to issue an MVS DUMP command and potentially omit vital dump parameters needed for the problem.

TASKMAX(*nnn*)

Sets the high limit of catalog address space service tasks that can be attached to process catalog requests. When the number of currently active catalog address space service tasks equals the limit, catalog management defers additional requests until a catalog address space service task is available.

For maximum performance/response, do not use this command. However, when storage is the primary concern, this command can be helpful.

nnn specifies the limit in decimal. You can specify a value from the lower limit to a maximum of 180. The default, set at IPL, is 60. You cannot specify a value lower than the highest number of tasks shown by message IEC359I, which appears in response to the MODIFY CATALOG,REPORT command. You can specify the lower limit in the SYSCAT*nn* member of SYS1.NUCLEUS.

UNALLOCATE[(*catname*)]

Specifies that all catalogs allocated to catalog address space (CAS) are to be removed from CAS. The CAS storage used by the catalogs is freed and the devices on which these catalogs reside are also unallocated from CAS. If you specify a catalog (*catname*), only the specified catalog is unallocated. If this is the only catalog that resides on the device, the device on which the catalog resides is unallocated from CAS.

VCLOSE(*vvvvv*)

Closes the VSAM volume data set (VVDS) whose volume serial is *vvvvv* without affecting any existing allocations. The next time a request is processed for that VVDS, the VVDS is reopened with a new set of control blocks.

{VLF | NOVLF}(*catname*)

Specifies that the catalog data space cache is to be activated (VLF) or deactivated (NOVLF) for the catalog named *catname*. To activate the catalog data space cache (CDSC) for a catalog, the catalog must already be defined as eligible for CDSC. To deactivate CDSC, the catalog must already be allocated to CDSC.

VDUMPPON | VDUMPOFF

Specifies whether or not VSAM dynamic dumping is to occur. Dynamic dumping does not occur unless you specify VDUMPPON. You can use these VDUMPPON commands for problem solving:

-
- MODIFY CATALOG,VDUMPPON(*rc,rsn*)
- MODIFY CATALOG,VDUMPPON(*rc,rsn,pdf*)
- MODIFY CATALOG,VDUMPPON(*rc,rsn,pdf,comp*)

where:

rc Specifies the VSAM return code in decimal format (one to three characters with a maximum of 255), or ** asterisks for wild card searches.

rsn

Specifies the VSAM reason code in decimal format (one to three characters with a maximum of 255), or ** asterisks for wild card searches.

pdf

Specifies the VSAM Problem Determination Function code (one to three characters with a maximum of 255), or ** asterisks for wild card searches.

comp

Specifies the component code (0 - 5) or ** asterisks for wild card searches.

VUNALLOCATE | NOVUNALLOCATE

Specifies that all dynamically allocated VVDSs are to be unallocated from CAS (VUNALLOCATE) when a request is completed, or left allocated to CAS (NOVUNALLOCATE) after a request has completed. By default, VVDSs are unallocated after processing a request. Performance can be improved if you

MODIFY Command

specify NOVUNALLOCATE, because repeated requests for a volume do not result in repeated dynamic allocations for the VVDS.

If you specify NOVUNALLOCATE, VVDSs remain allocated until VUNALLOCATE is issued, CAS is restarted, or the system is IPLed.

VUNALLOCATE can be used to unallocate a VVDS from CAS to allow a volume to be varied offline.

VVDSSPACE(*primary,secondary*)

Indicates that the Catalog Address Space should use the values specified as the primary and secondary allocation amount in tracks for an implicitly defined VVDS. The default value is ten tracks for both the primary and secondary values. The specified values are preserved across a Catalog Address Space restart, but are not preserved across an IPL.

For examples of the MODIFY CATALOG command see *z/OS DFSMS Managing Catalogs*.

Communicating with the Device Manager Address Space

Use the MODIFY DEVMAN command to communicate with the device manager address space to display information or to request a specified service. **Use this command only at the direction of the system programmer.**

```
F DEVMAN,{DUMP}
          {REPORT}
          {RESTART}
```

The following are brief descriptions of the parameters.

DUMP

Captures a diagnostic dump of the device manager address space, including the dataspace that contains device manager CTRACE records.

Note: The device manager CTRACE component name is SYSDMO. To connect device manager to an output writer, use the command TRACE CT,ON,COMP=SYSDMO.

REPORT

Provides basic information about the current activity and module levels for the device manager address space.

RESTART

Terminates the device manager address space and restarts the device manager in a new address space. The system allows any subtasks that are active in the device manager address space at the time of the restart to finish processing. The time allowed for subtask completion is determined by using the average time taken by previous subtasks. The system abnormally ends any subtasks that do not complete in time before it restarts the address space.

Notes:

1. Use RESTART to avoid IPL when you install software. You can install most device manager APARs by refreshing LLA (F LLA,REFRESH), and then restarting device manager (F DEVMAN,RESTART).

2. You can end and not restart the device manager address space by using the CANCEL DEVMAN command. When you end the address space in this way, you must restart the device manager with the DEVMAN cataloged procedure.

Changing the DLF Processing Mode

Use the MODIFY DLF,MODE command to change the processing mode for the data lookaside facility (DLF).

```
F DLF,MODE={DRAIN|D }
           {QUIESCE|Q}
           {NORMAL|N }
```

The parameters are:

DLF

The name of the job.

MODE

Changes the processing for DLF.

Note: This is the beginning of the shutdown process for DLF.

DRAIN or D

Sets drain mode. In drain mode, DLF connects the user to existing DLF objects or disconnects the user from DLF objects. No new DLF objects are created while in DRAIN mode.

Setting drain mode is part of the shutdown procedure for DLF. DLF will operate in normal mode until the STOP DLF command is issued.

QUIESCE or Q

Sets quiesce mode. In quiesce mode, DLF only disconnects the user from DLF objects. No new DLF objects are created while in QUIESCE mode. Specify QUIESCE only at the direction of the system programmer.

Setting quiesce mode is part of the shutdown procedure for DLF. DLF will operate in normal mode until the STOP DLF command is issued.

NORMAL or N

Sets normal mode. In normal mode DLF creates and connects the user to new DLF objects, connects the user to existing DLF objects, and disconnects the user from DLF objects.

Normal mode is in effect when DLF is active and not stopping. You need to specify MODE=NORMAL only when you have started to shutdown DLF but then decide to resume normal processing and cancel the shutdown.

Changing the DLF parameters

Use the MODIFY DLF,NN command to cause the data lookaside facility (DLF) to use the specified COFDLFxx member of the logical parmlib. The COFDLFxx parmlib member must contain a valid CLASS statement. The only values that change are limits on DLF use of storage through the keywords: MAXEXPB and PCTRETB. The new limit values affect only new connections; any existing connections or DLF objects that exceed the new limits are not affected. The

MODIFY Command

CONEXIT parameter from the initial COFDLFxx member remains valid. See *z/OS MVS Initialization and Tuning Reference* for further information about the use of the COFDLFxx member.

```
F DLF,NN=xx
```

The parameters are:

DLF

The name of the job.

NN=xx

Causes DLF to use the COFDLFxx member of the logical parmlib where xx identifies the COFDLFxx member of the logical parmlib.

Displaying DLF Status

Use the MODIFY DLF,STATUS command to display the limits from the COFDLFxx parmlib member currently in effect.

```
F DLF[,{STATUS|ST|S}]\n      ,SM\n      ,SB
```

The parameters are:

DLF

The name of the job.

STATUS or ST or S

Displays the DLF limits set in the COFDLFxx parmlib member that is currently in effect.

SM

The status is displayed in megabytes.

SB

The status is displayed in 4K blocks.

Note: When STATUS is specified the display is in the most recently requested unit, 4K blocks or megabytes. The default is megabytes.

Building and Replacing Library Lookaside Directories

Use the MODIFY LLA command to cause the library lookaside (LLA) program to build a new copy of all or part of the library directory indexes and then replace the old copy with this new copy.

```
F LLA,{REFRESH }\n      {UPDATE=xx}
```

The parameters are:

LLA

The name of the job.

REFRESH

Causes LLA to build a new copy of all the library directory indexes for the complete set of data sets currently managed by LLA. This procedure is necessary when an installation makes changes to the LLA directories.

UPDATE=xx

Causes LLA to rebuild a specified part of the directory. xx identifies the CSVLLAxx parmlib member or the data set pointed to by the IEFPARM DD statement in the START LLA procedure. CSVLLAxx contains libraries LLA is to manage.

Notes:

1. Use the MODIFY LLA command rather than stopping and restarting LLA to change the library directory indexes; system performance is slowed anytime LLA is stopped.
2. The MODIFY LLA command does not reload (or refresh) modules that are already loaded, for example, modules in long-running or never-ending tasks. The refreshed version does not get picked up unless the module is loaded after the MODIFY LLA completes. To refresh such a module, the system programmer has two options:
 - If the module has no co-requisite requirement in LPALIB, you can use the subsystem's command to replace the module, or stop and then restart the long-running or never-ending task.
 - Re-IPL the system with the CLPA option.

Operating with the Network File System Server

The network file system server provides transparent access from different client workstations to MVS data sets. It allows access to most file formats available under MVS. A server provides resources to the network service, such as disk storage and file transfer. For more information about the network file system server, see *z/OS Network File System Guide and Reference*. For information about network file system server messages (GFSAXXXI), use that book or LookAt (see “Using LookAt to look up message explanations” on page xviii).

Managing the Object Access Method (OAM)

Use the MODIFY OAM command to display information about the object access method (OAM) or to request that OAM perform a specified service: object management, space management, or recovery functions.

F OAM,parameters

For a detailed discussion of the MODIFY OAM command parameters, refer to the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*, and the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Recycling z/OS UNIX System Services (z/OS UNIX)

Use the MODIFY OMVS command to recycle z/OS UNIX System Services. This is an alternative to re-IPLing the system in order to reinitialize the z/OS UNIX System Services environment. This command should be used only on a **limited**

MODIFY Command

basis when complete reinitialization and reconfiguration are required. Prior to issuing MODIFY OMVS to initiate a shutdown, you should review the information about shutdown in *z/OS UNIX System Services Planning*.

The MODIFY OMVS,SHUTDOWN command completely disables z/OS UNIX services. IBM strongly recommends that an installation perform the following steps prior to issuing the command:

1. Attempt to dry up batch and interactive workloads. Once a shutdown request is accepted, jobs that subsequently attempt to connect to z/OS UNIX services for the first time will be delayed until restart occurs; jobs that are already connected to z/OS UNIX services (dubbed) will be sent terminating signals and will end abruptly.
2. Follow recommended procedures to quiesce applications and subsystem workloads using z/OS UNIX services. This will allow subsystems such as DB2, CICS, and IMS, and applications such as SAP, LOTUS DOMINO, NETVIEW, and WEBSPHERE to be quiesced in a more controlled manner than would occur if simply using this command. Use the D OMVS,A=ALL command to determine which applications, if any, require quiescing.
3. Shut down TCP/IP and all TCP/IP applications in the manner that TCP/IP recommends.
4. Shut down any colony PFS address spaces. This could potentially include NFS and DFS.

```
F OMVS{,SHUTDOWN}
  {,RESTART[,OMVS=(xx,yy)]}
```

The parameters are:

OMVS

The name of the job.

SHUTDOWN

Request that the system shut down the UNIX services environment, which includes quiescing all running UNIX services work on a given system.

RESTART

Request that the system restart the UNIX services environment which has shut down successfully.

OMVS

Specifies the parmlib member or members (BPXPRMxx) to use to locate the parmlib statements to configure the z/OS UNIX kernel.

If the OMVS parameter is not specified, the BPXPRMxx member used on the previous start of OMVS is processed again, with any updated contents taking effect.

Example 1

To shut down the UNIX services environment, enter:

```
F OMVS,SHUTDOWN
```

The following highlighted console message will be displayed when the shutdown request is accepted:

```
BPXI055I OMVS SHUTDOWN REQUEST ACCEPTED
```

If a shutdown request cannot be accepted, the following console message will be displayed:

BPXI057I OMVS SHUTDOWN REQUEST REJECTED

If a shutdown request is delayed because of blocking jobs or processes, the following console message will be displayed when the delay exceeds a finite time interval:

BPXI064E OMVS SHUTDOWN REQUEST DELAYED

Dynamically activating maintenance for z/OS UNIX System Services (z/OS UNIX)

Use the MODIFY OMVS command to dynamically activate maintenance for z/OS UNIX System Services.

```
F OMVS {,ACTIVATE=SERVICE}
{,DEACTIVATE=SERVICE}
```

The parameters are:

ACTIVATE=SERVICE

Requests that the system dynamically activate SMP/E installable service for the z/OS UNIX kernel and logical file system (LFS) components without requiring a component restart or system IPL. The system activates the service from the target libraries specified in BPXPRMxx parmlib statements, SERV_LPALIB and SERV_LINKLIB. When the F OMVS,ACTIVATE=SERVICE is issued, the system allocates and opens load libraries specified on SERV_LPALIB and SERV_LINKLIB to locate and load the new services items. Therefore, do not update these libraries during the activation process.

Restriction: The ACTIVATE=SERVICE parameter will **only** activate those service items in the target libraries that are appropriate for the installation service level. Your installation's service level might be too low to allow dynamic activation of some service items in the target library.

Result: The system issues message BPXM061I to display the service items to be activated. It then issues BPXM061D and prompts for the activation to continue based on the information in BPXM061I. The operator can reply 'Y' to let activation continue. Any other reply cancels the command.

You can also use the DISPLAY command (D OMVS,ACTIVATE=SERVICE) to display the dynamically activated service. See "Displaying z/OS UNIX System Services Status" on page 4-174.

DEACTIVATE=SERVICE

Requests that the system dynamically back off a set of dynamically activated service items. You might need to use this command if you encounter a problem with a service item, or if you do not need it any longer. The DEACTIVATE=SERVICE parameter dynamically deactivates **only** those service items that were activated with the previous ACTIVATE=SERVICE command.

Result: The system issues message BPXM063I to display the service items to be deactivated. It then issues message BPXM063D and prompts for the deactivation to continue based on the information in message BPXM063I. The operator can reply 'Y' to let deactivation continue. Any other reply cancels the command.

MODIFY Command

You can also use the DISPLAY command (D OMVS, ACTIVATE=SERVICE) to display the dynamically activated service. See “Displaying z/OS UNIX System Services Status” on page 4-174.

Stopping a physical file system (PFS) through a logical file system (LFS) interface

Use the MODIFY OMVS command to stop a PFS through an LFS interface.

```
F OMVS,STOPPFS=psfname
```

The parameters are:

pfstype

Indicates the name of the PFS to be stopped. This is the name used on the FILESYSTYPE statement from the BPXPRMxx parmlib member that defined the PFS.

Stopping a Temporary File System (TFS)

Use the MODIFY TFS command to stop a TFS running in a colony address space. The MODIFY TFS command can be used to force a TFS to stop or terminate even if TFS file systems are mounted.

Note: The MODIFY TFS command is not supported if TFS runs in the z/OS Unix kernel address space.

The complete syntax for the MODIFY TFS command is:

```
F TFS,{STOP}  
      {TERM}  
      {FORCESTOP}  
      {FORCETERM}
```

TFS

The name of the TFS to be stopped.

STOP

This is the same function as the STOP command. If no TFS file systems are mounted, this command causes TFS to exit. A WTOR is issued allowing TFS to be restarted.

TERM

If no TFS file systems are mounted, this command causes TFS to exit without prompting to restart the TFS. You can issue the SETOMVS RESET=(xx) command to start another TFS.

FORCESTOP

Similar to STOP, issuing this command will cause TFS to terminate even if there are mounted TFS file systems.

FORCETERM

Similar to TERM, issuing this command will cause TFS to terminate even if there are mounted TFS file systems.

Enabling and Disabling the Application Response Measurement (ARM) Agent and Enterprise Workload Manager (EWLM) platform services

Use the MODIFY WLM,AM command to enable or disable Application Response Measurement (ARM) services and Enterprise Workload Manager (EWLM) platform services. For more information on ARM services and EWLM platform services, see the [@server](http://publib.boulder.ibm.com/eserver/v1r1/en_US/index.htm?info/icmain.htm) Information Center on the Internet at publib.boulder.ibm.com/eserver/v1r1/en_US/index.htm?info/icmain.htm.

The complete syntax for the MODIFY WLM,AM command is:

```
F WLM, AM={DISABLE|ENABLE}
```

WLM

The name of the job.

AM

Indicates that the state of ARM services is to be changed.

DISABLE

Disables ARM services. Applications calling any of the ARM services will receive a return code indicating that this function is not operational from that point on; however, the applications continue to run. Also, if the EWLM managed server is connected, it will be disconnected from WLM. The EWLM managed server might not have to terminate; it might continue to run and wait for further instructions (for example to restart), but subsequent attempts to connect the EWLM managed server to the platform are rejected, before the ARM services get re-enabled.

ENABLE

Enables previously disabled ARM services. ARM-instrumented applications that are already active at the time this command is invoked must be restarted if the ARM calls should be processed. In addition, when the ARM services are enabled, you can restart the EWLM managed server.

Changing Workload Manager Resource States

Use the MODIFY WLM command to change the state of a resource. This command changes the resource state only on the system where you issue the command.

Note: The MODIFY WLM command cannot be specified in the COMMNDxx parmlib member.

Resource states are used by workload management in conjunction with scheduling environments to ensure that work is scheduled only on a system with the appropriate resources to handle that work. See *z/OS MVS Planning: Workload Management* for more information about resources and scheduling environments.

The complete syntax for the MODIFY WLM command is:

```
F WLM, [RESOURCE=resourceName,{ON|OFF|RESET}]
```

WLM

The name of the job.

MODIFY Command

RESOURCE=resourceName

Changes the state of *resourceName*.

ON

Specifies that if the required resource state in a scheduling environment is ON, that requirement will be satisfied on the target system.

OFF

Specifies that if the required resource state in a scheduling environment is OFF, that requirement will be satisfied on the target system.

RESET

Specifies that this resource setting will satisfy neither an ON nor an OFF resource requirement. Therefore if a scheduling environment includes *resourceName* in its list of resources (whether ON or OFF), then that scheduling environment will not be available on the target system.

Example

To change the setting of the DB2A resource to ON, enter:

```
F WLM,RESOURCE=DB2A,ON
```

The system will respond:

```
IWM039I RESOURCE DB2A IS NOW IN THE ON STATE
```

Specifying Data Set Selection Criteria for an External Writer

You can use the MODIFY command to specify the criteria that an external writer is to use in selecting data sets for processing.

```
F [XWTR.|jobname.]identifier,  
  {{CLASS|C}=[classes] }  
  {{DEST|D}=[LOCAL  
           |remote-workstation-name] }  
  {{FORMS|F}=[forms-name] }  
  {{JOBID|J}=[JOBnnnnn|Jnnnnn ]  
   |[STCnnnnn|Snnnnn]  
   |[TSUnnnnn|Tnnnnn] }  
  {{WRITER|W}=[STDWTR  
           |user-writer-name] }
```

The parameters are:

XWTR

The name of the IBM-supplied cataloged procedure for the external writer.

jobname

The job name assigned to the external writer.

The job name for a started task depends on whether the JOBNAME parameter was specified on the START command for the task:

- If JOBNAME was specified, *jobname* is the name specified on the JOBNAME parameter.

- If JOBNAME was *not* specified and the source JCL for the started task is a *job*, *jobname* is the name specified on the JCL JOB statement.
- If JOBNAME was *not* specified and the source JCL for the started task is a *procedure*, *jobname* is the member name.

You can use asterisk notation to specify more than one job or started task on the MODIFY command. See “Using Asterisks in MODIFY Commands” on page 4-310 for more information.

identifier

The identifier, from the START command, of the writer to be modified. (Refer to “Displaying Started Task Status” on page 4-157 for information about determining the jobname and identifier of currently active started tasks.)

The following types of identifiers can be used:

- The identifier that was specified on the START command.
- */Jdevnum*, the device number specified on the START or MOUNT command. A device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). You can precede the device number with a slash to prevent ambiguity between the device number and a device type or identifier.
- *devicetype*, the type of device specified on the START or MOUNT command.

If no identifier was specified, the identifier “STARTING” is temporarily assigned until the system can assign another according to the following order of precedence:

1. If an identifier was not specified on the START command, the identifier is the device type (for example, 3410) or device number (for example, X'0000') specified on the START or MOUNT command.
2. If an identifier, a device type, or a device number was not specified on the START or MOUNT command, the identifier is the device type specified on an IEFRDER DD statement (invoking a cataloged procedure) in the JCL.
3. If none of the above was specified, the identifier defaults to the job name.

You can use asterisk notation to specify more than one job or started task on the MODIFY command. See “Using Asterisks in MODIFY Commands” on page 4-310 for more information.

CLASS= or C=[classes]

Select only data sets enqueued in the specified classes. You can specify up to eight output classes, in priority order.

The output classes are named without separating commas. If no default class was specified in the cataloged procedure to start the external writer, and no class list is provided in the START or MODIFY command, the external writer selects any ready data set on the hard-copy queue.

JOBID= or J= [JOB or J]nnnnn, [STC or S]nnnnn, or [TSU or T]nnnnn

Select only data sets from the job with this subsystem-assigned JOBID, where *nnnnn* is the JOB id number, the STC id number, or the TSU id number. If JOBID is omitted, the external writer does not select data sets by job.

Notes:

1. JES2 ignores the prefix (JOB or J, STC or S, or TSU or T) and uses only the id number, *nnnnn*.
2. With JES2 operating on z/OS version 1.2 or higher, you may specify up to a six-digit id number, or *nnnnnn*.

MODIFY Command

WRITER or W=

STDWTR

Select only data sets that are to be processed by the standard (IBM-supplied) writer.

user-writer-name

Select only data sets that are to be processed by the specified user writer.

If WRITER= is specified without STDWTR or user-writer-name, the external writer does not use the writer program as a data set selection criterion and automatically invokes the correct writer programs.

FORMS= or F=[forms-name]

Select only data sets that specify this forms name.

If *forms-name* is omitted, the external writer does not use the forms name as a data set selection criterion, and notifies you whenever a forms change is needed.

DEST= or D=

LOCAL

Select only data sets destined for the central processor complex.

remote-workstation-name

Select only data sets destined for the specified remote workstations attached to this local complex.

If DEST= is specified without LOCAL or remote-workstation-name, the external writer does not use the destination as a data set selection criterion.

Previously-specified options remain in effect until respecified. Before the first MODIFY command is issued, the default options are:

CLASS=(see note), JOBID=, WRITER=, FORMS=, DEST=LOCAL

Note: If no default class list is specified in the cataloged procedure to start the external writer, and you do not provide a class list in the START command, the external writer does not begin processing until you enter a MODIFY command.

The MODIFY command passes the entire command buffer, including comments, to the external writer that is to be modified. Therefore, all modifiable external writers should be sensitive to embedded blanks in their parameter fields.

Causing an External Writer to Pause

You can use the MODIFY command to cause an external writer to pause for operator intervention.

```
F [XWTR.|jobname.] identifier,{PAUSE|P}={FORMS    }  
                           {DATASET}
```

The parameters are:

XWTR

The name of the IBM-supplied cataloged procedure for the external writer.

jobname

The job name assigned to the external writer.

The job name for a started task depends on whether the JOBNAME parameter was specified on the START command for the task:

- If JOBNAME was specified, *jobname* is the name specified on the JOBNAME parameter.
- If JOBNAME was *not* specified and the source JCL for the started task is a *job*, *jobname* is the name specified on the JCL JOB statement.
- If JOBNAME was *not* specified and the source JCL for the started task is a *procedure*, *jobname* is the member name.

You can use asterisk notation to specify more than one job or started task on the MODIFY command. See “Using Asterisks in MODIFY Commands” on page 4-310 for more information.

identifier

The identifier of the writer to be modified. (Refer to “Displaying Started Task Status” on page 4-157 for information about determining the job name and identifier of currently active started tasks.)

The following types of identifiers can be used:

- The identifier that was specified on the START command.
- */devnum*, the device number specified on the START or MOUNT command. A device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). You can precede the device number with a slash to prevent ambiguity between the device number and a device type or identifier.
- *devicetype*, the type of device specified on the START or MOUNT command.

If no identifier was specified, the identifier “STARTING” is temporarily assigned until the system can assign another according to the following order of precedence:

1. If an identifier was not specified on the START command, the identifier is the device type (for example, 3410) or device number (for example, X'0000') specified on the START or MOUNT command.
2. If an identifier, a device type, or a device number was not specified on the START or MOUNT command, the identifier is the device type specified on an IEFRDER DD statement (invoking a cataloged procedure) in the JCL.
3. If none of the above was specified, the identifier defaults to the job name.

You can use asterisk notation to specify more than one job or started task on the MODIFY command. See “Using Asterisks in MODIFY Commands” on page 4-310 for more information.

PAUSE= or P=FORMS

For unit record devices, the writer is to pause when a change of forms is necessary. The writer stops when it encounters a form name other than the installation default form name on a SYSOUT DD statement.

PAUSE= or P=DATASET

For unit record devices, the writer is to pause before starting to process each data set. When you are ready to continue processing, you can restart the writer by entering a single character response to message IEF382A.

Example 1

MODIFY Command

To stop writer 00E before it processes each new data set, enter:
F 00E,PAUSE=DATASET

Example 2

To stop writer ABCD before it processes each new data set, enter:
F /ABCD,PAUSE=DATASET

MONITOR Command

Use the MONITOR command to display jobnames, data set status, and time-sharing user sessions continuously, and to add certain information to mount and demount messages.

The MONITOR command does not display Advanced Program-to-Program Communication/MVS (APPC/MVS) transaction programs.

Notes:

1. To stop the MONITOR display, use the STOPMN command.
2. The monitor display ends when the specified MCS or SMCS console is varied offline. The operator must re-issue the MONITOR command when the MCS or SMCS console is restarted.
3. To receive MONITOR displays after the MCS or SMCS console is switched, re-issue the MONITOR command on the new MCS or SMCS console.
4. To route MONITOR displays to a specific console, use the MSGRT command. Otherwise, the display appears at the console at which you entered the command.

Note: The MSGRT command is not supported on systems at z/OS V1R8 and higher.

5. The DSNAME and SPACE operands are not routable.

Use the SETCON command if you want the system to generate monitor messages, but you do not want the messages to appear on the console.

Scope in a Sysplex

The MONITOR command has sysplex scope only when you specify L=. See “Using Commands That Have Sysplex Scope” on page 1-11 for an explanation of sysplex scope.

Syntax

The complete syntax for the MONITOR command is:

```
MN {JOBNAMES[,T] [,L={a|name|name-a}]}
{DSNAME}
{SPACE}
{STATUS}
{SESS[,T]}
```

Parameters

The parameters are:

JOBNAMES

The system is to display the name of each job when the job starts and terminates, and display unit record allocation when the step starts. If a job terminates abnormally, the job name appears in a diagnostic message.

DSNAME

The system is to display, in mount messages, the name of the first

MONITOR Command

non-temporary data set allocated on the volume to which the messages refer. No data set name appears in messages for data sets with a disposition of DELETE.

SPACE

The system is to display, in demount messages, the available space on the direct access volume.

STATUS

The system is to display the data set names and volume serial numbers of data sets with dispositions of KEEP, CATLG, or UNCATLG whenever they are freed.

SESS

The system is to display the user identifier for each TSO terminal when the session is initiated and when it is terminated. If the session terminates abnormally, the user identifier appears in the diagnostic message.

T The system is to display the time (in hours, minutes, and seconds) along with the user identifier or job name information. When specified, **T** is activated for all consoles that have MONITOR turned on.

L=a, name, or name-a

The display area (*a*), console name (*name*), or both (*name-a*) where the system is to present the display. If you omit this operand, the console on which the MONITOR command is entered is assumed.

Example

To display the job name information and the time when each job starts and terminates, enter:

```
MN jobnames,T
```

MOUNT Command

Use the MOUNT command to allow allocation of an I/O device to all job steps that require a particular volume without intervening demountings and remountings of the volume. Because the system must schedule MOUNT commands, there is a short delay between when you issue the command and when the volume is mounted.

In a JES3 complex, use the MOUNT command with great care. When a volume is to be mounted on a JES3-managed direct access device, you must enter a MOUNT command on each system in the complex that has access to the device. In addition, if you are requesting that a volume be mounted on a JES3-managed unit, you must also enter a JES3 mount command (*MODIFY,S,M=...). See *z/OS JES3 Commands*.

Once the system has executed a MOUNT command, the specified device becomes reserved and remains mounted and reserved until an UNLOAD or VARY OFFLINE command is issued. A reserved direct access volume can be assigned the USE attribute of PUBLIC, PRIVATE, or STORAGE. A reserved tape volume can be assigned the USE attribute of PRIVATE or PUBLIC.

If you need to nullify a scheduled mount before the system executes the MOUNT command, use the CANCEL command.

Scope in a Sysplex

The MOUNT command has sysplex scope only when you issue the command against an automatically switchable tape device. See “Using Commands That Have Sysplex Scope” on page 1-11 for an explanation of sysplex scope.

Syntax

The complete syntax for the MOUNT command is:

M	{ [/]devnum }	,VOL=({NL},serial) [,USE={STORAGE}]
{devicetype}	{SL}	{PUBLIC}
	{AL}	{PRIVATE}

Parameters

The parameters are:

[/]devnum

The device number for the input/output device to be mounted. A device number is 3 or 4 hexadecimal digits. A slash (/) must precede a 4-digit number and is optional before a 3-digit number.

devicetype

The type of device to be mounted. It can be any IBM-supplied name (for example, 3380).

VOL=(NL,serial)

The volume specified does not have a standard label. This parameter must not be used for direct access volumes. The serial number, up to six characters long, is used for allocation references.

Do not try to mount volumes with a label type of NL in a system-managed tape library these volumes are not supported.

MOUNT Command

VOL=(SL,serial)

The volume specified has a standard label (SL). The serial number, up to six characters long, is used for label verification and allocation references. Tape label verification is not performed until the tape is opened.

VOL=(AL,serial)

The volume has an American National Standard label (AL). The serial number, up to six characters long, is used for label verification and allocation references. AL can be specified only if it was selected as an option at system installation. Tape label verification is not performed until the tape is opened.

USE=STORAGE, PUBLIC, or PRIVATE

The USE attribute, defined by your installation procedures, to be assigned to the specified volume. Refer to *z/OS MVS Using the Subsystem Interface* for more information on USE attributes.

Example 1

To request that a volume with a standard label of 222222 be mounted on device 282, enter:

```
m 282,vol=(s1,222222),use=private
```

Example 2

To request that a volume with a standard label of 222222 be mounted on device 3333, enter:

```
m /3333,vol=(s1,222222),use=private
```

Tape Library Dataserver Considerations

The MOUNT command is the only type of library request that can be used to request a specific system-managed tape library device. (Requesting a specific device is also called demand allocation.) The request must specify a system-managed tape library volume.

Table 4-25 shows the possible volume and device combinations you can specify on the MOUNT command, and how the system will handle those inputs.

Table 4-25. Possible Volume and Device Combinations on MOUNT Command

Volume location:	Device location:	System action:
Non-library	Non-library	Processing continues
Non-library	Library	Issues error message IEF120I
Library	Non-library	Issues error message IEF113I
Library A	Library B	If volume and device are in different libraries — issues error message IEF111I
Library A	Library A	If volume and device are in the same library but the volume is not eligible for allocation on the device — issues error message IEF150I
Library A	Library A	If volume and device are in the same library and the volume is eligible for allocation on the device — processing continues

MSGRT Command

MSGRT COMMAND NO LONGER SUPPORTED

The MSGRT command is not supported on systems at z/OS V1R8 and higher.

The TR=A option of the MSGRT command is not supported on systems at z/OS V1R7 and higher.

Use the MSGRT command to establish or change message routing instructions for displays from the DISPLAY or CONFIG commands to a specified message area, console, or both. MSGRT also controls the action of the CONTROL, MONITOR, and STOPMN commands. The instructions you specify remain in effect until another MSGRT command is issued for that message area or console or you IPL the system. The command can route display information or stop the routing of display information.

The MSGRT parameter on the CONSOLE statement in the CONSOLxx parmlib member controls the display routing instructions at IPL.

The MSGRT command has no effect on extended MCS consoles or on system consoles, and is not valid for managing these consoles. The MSGRT command also has no effect on commands issued via the ROUTE *ALL or ROUTE systemgroupname commands.

Syntax

The complete syntax for the MSGRT command is:

```

MR { [D=(operand[,operand]...)] [,L={a      }] }
    | TR=A           {name      }
    | K              {name-a   }
    | CF
    | MN

    {REF
    {NONE
    {CONTn

```

Parameters

The parameters are:

D=operand[,operand]...

The system is to route DISPLAY command output to the specified MCS and SMCS consoles. With a few exceptions, any DISPLAY command is valid with this command. If you only specify one operand, the parentheses are not necessary. Note that the operands on the MSGRT parameter on the CONSOLE statement in the CONSOLxx parmlib member are not always the same as the operands on the D= operand of the MSGRT command.

The following DISPLAY operands are not supported:

JOBS, J, TS, T

MSGRT Command

All others may be specified on the MSGRT command in the same way as they are specified on the DISPLAY command, with the exception of SLIP, which is specified by S.

TR=A

The system is to route the TRACK A command display and the action of the STOPTR command to the specified MCS or SMCS console. The TR=A option is not supported on systems at z/OS V1R7 and higher.

- K** Subsequent CONTROL command action is to affect the specified MCS or SMCS console until the option is discontinued or you IPL the system. You can use MSGRT by default to route only those CONTROL functions that you can explicitly route with the L=name-a operand.

CF

The system is to route the reply to the CONFIG command to the specified MCS or SMCS console.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display is to be presented.

NONE

For the console on which you enter the command, the system is to discontinue all current message routing instructions for displays.

REF

The system is to display the current message routing instructions.

CONT_n

The system is to continue the display response to a previous MR REF command. When CONT_n appears in the display response, all current routing instructions have not been displayed. Issue MR CONT_n to continue the display with the *n*th line, where *n* is a decimal digit from 1 to 9.

MN

The system is to route the displays produced by the MONITOR command and the action of the STOPMN command to the specified MCS or SMCS console.

Notes:

1. You can enter multiple DISPLAY (D), MONITOR (MN), and CONTROL (K) command routings in one MSGRT command by enclosing the operands in parentheses and separating them with commas. For example, MR (D=(A,C,R),L=CON1),(K,L=CON2).
2. The MSGRT routing instructions for a console can be temporarily overridden by one of the following:
 - A command the system generates internally
 - A command on which you specify the display area and name (L=name-a) operand

The overridden MSGRT instructions, however, go back into effect as soon as the overriding command finishes. For example, when a VARY command is issued for a range of devices, the VARY command processor internally issues a DISPLAY command, defining the target console for the output as the console that issues the VARY command. For that one issuance of the DISPLAY command, the routing instructions for the target console are overridden. In the case of an instream command or a command from parmlib member COMMNDxx, the master console receives the output.

Example 1

To display the MSGRT operands currently in effect, enter:

MR REF or MR

Example 2

To present all DISPLAY UNIT status displays in area B of console named CON12, and all display consoles status displays in area A of the console through which you enter the MSGRT command, enter:

MR (D=U,L=CON12-B),D=CONSOLES,L=A

Stopping Message Routing

To stop message routing you established with the MSGRT command, enter the applicable MSGRT command without the L= operand. For example, to remove routing instructions established for the CONTROL command, enter:

MR K

PAGEADD Command

PAGEADD adds auxiliary storage space (local page data sets) to the system. The page data sets added remain available to the system until you IPL with the CLPA (create link pack area) option, IPL with the CVIO (clear virtual I/O) option, or issue a PAGEDEL command. PAGEADD can also direct VIO pages away from a page data set that is being added.

Use the PAGEADD command only at the request of your system programmer.

You might need to add auxiliary storage space if any of these conditions exist:

- The planned system load increases.
- The space provided during system initialization proves insufficient.
- Space is lost because of a hardware failure.

If the system detects a shortage of available auxiliary storage space, it issues the following message:

IRA200I AUXILIARY STORAGE SHORTAGE

The system rejects LOGONs and START commands until the shortage is relieved. If the shortage increases, the system issues the following message:

IRA201I CRITICAL AUXILIARY STORAGE SHORTAGE

The system rejects LOGONs and START commands and might delay the starting of certain initiators until the shortage is relieved.

Requested data sets are placed in use in the order specified in the command. The system informs you when each is available for use.

The number of page data sets that can be in use by the system is limited by the number specified on the PAGTOTL system parameter. (See *z/OS MVS Initialization and Tuning Guide*.) If these limits are exceeded, the system issues a message, and you cannot add any more data sets of that type during this IPL. However, if the limit has been exceeded for one type of data set, you could still add data sets of the other type.

If your paging device contains a subsystem cache, it is important to note that the PAGEADD command determines the status of the subsystem cache and resets it, if necessary. However, MVS does not set the subsystem cache on if it was powered

PAGEADD Command

off with the IDCAMS command SETCACHE SUBSYSTEM OFF. MVS resets the data in the cache only under certain circumstances, such as at a cold start or at the first issuance of a PAGEADD command to the device.

The page data sets must be defined before you can issue the PAGEADD command. If the volume containing the data set is not mounted before you enter the command, the system issues a mount message.

A data set that is draining, as the result of a PAGEDEL DRAIN command, can be made read/write again by issuing a PAGEADD for the data set. A data set that has been marked BAD cannot be made read/write again with the PAGEADD command.

When you issue a PAGEADD command for a data set previously deleted with a PAGEDEL command, the system might indicate that some slots are already in use on the newly allocated data set. These slots contain pages that the system has migrated to another data set but that the owner has not yet referenced. Once the owner references the page, the system frees the slot both from the newly allocated data set and from the data set on which the page actually resides.

Use the DISPLAY ASM command to identify the page data sets the system is currently using.

Syntax

The complete syntax for the PAGEADD command is:

```
PA {[PAGE=]} {dsname[,dsname]...}  
{NONVIO=}
```

Parameters

The parameters are:

[PAGE=]*dsname[,dsname]*...

The name of one or more page data sets to be added. If *dsname* is not the name of a page data set, the system issues message IEE782I.

NONVIO=*dsname[,dsname]*...

The name of one or more page data sets to be added. The system is not to use these added page data sets to receive VIO pages.

Example 1: To add one page data set, enter:

```
PA page3
```

Example 2: To add three page data sets, enter:

```
pa page=sys1.page01,sys1.page02,page3
```

Example 3: To add SYS1.PAGE01 as a page data set and specify not to use it for VIO paging, enter:

```
PA NONVIO=SYS1.PAGE01
```

PAGEDEL Command

Use the PAGEDEL command to delete, replace, or drain (quiesce) local page data sets.

Attention: Use this command only at the request of your system programmer.
Misuse can seriously impair system performance.

This command allows you to remove or replace local page data sets without requiring an IPL.

Note: Draining a data set means freeing its in-use slots. The system effects this by making the data set read-only.

You might need to delete, replace or drain local page data sets for any of the following reasons:

- The hardware is being reconfigured.
- The hardware is generating I/O errors.
- The page configuration is being changed.
- System tuning requires the change.

When you *replace* a local page data set, the system migrates the in-use slots from the old data set to the new one.

When you *delete* a page data set, the system migrates the in-use slots to other data sets before it deletes the data set.

The system keeps track of the in-use slots on both the old or deleted data set and the new data set until the owner references the pages. Thus, when you issue a PAGEADD command to allocate a new data set, the system might indicate that some slots on the newly allocated data set are already in use. As soon as the owner references a page, the system frees the slot both from the newly allocated data set and from the data set to which the page was migrated.

Notes:

1. You cannot use PAGEDEL to delete, replace, or drain the PLPA, common, or the last local page data sets.
2. When you enter a PAGEDEL command, the system issues a highlighted, non-rollable message to indicate that the command is accepted. The message remains on the console screen until the PAGEDEL command completes.
3. If you enter a PAGEDEL command while a PAGEDEL command is already in progress, the system issues a message that it rejects the command.
4. The system rejects a PAGEDEL command that decreases the amount of auxiliary storage below a fixed percentage of the available auxiliary storage.
5. To identify the page data sets the system is currently using or the status of the PAGEDEL command, issue the DISPLAY ASM command.
6. When issuing the PAGEDEL DELETE command, there is the potential for significant storage usage. Several blocks of storage in ESQA are obtained in order to process the PAGEDEL command. This includes a 'x'500' byte block for each cylinder on the deleted data set that contains in-use slots. Some of this storage will remain in use and not freed until all the in-use slots are freed, which could be some time after the PAGEDEL has completed (as indicated by message IEE205I).

PAGEDEL Command

Syntax

The complete syntax for the PAGEDEL command is:

```
PD {DELETE,PAGE={dsname[,dsname]...} }  
{REPLACE,PAGE={(dsname,rdsname)[,(dsname,rdsname)]...}}  
{DRAIN,PAGE={dsname[,dsname]...} }
```

Parameters

The parameters are:

DELETE

Specifies that the system is to remove one or more local page data sets from system use. The system migrates the in-use slots of the deleted data set(s) to other page data sets.

REPLACE

Specifies that a local page data set is to be replaced by a newly-opened data set of equal or greater size. The new data set must previously have been formatted and cataloged. It can be on a different type of device than the original data set. REPLACE fails if an I/O error occurs on either data set. The system migrates the in-use slots from the old data set to the new data set, then remove the old data set from system use.

DRAIN

Specifies that one or more local page data set are to be made read-only. When the current tasks complete, the in-use slots are freed during normal system operation. When you plan a PAGEDEL DELETE or REPLACE operation as part of a system reconfiguration, by allowing the data sets to drain (quiesce) before issuing the DELETE or REPLACE, you will reduce the number of in-use page data sets to migrate. You can make a data set that is draining read/write again by issuing a PAGEADD command for the data set.

PAGE=dsname[,dsname]...

The name of one or more local page data sets. If *dsname* is not the name of an in-use local page data set, the system issues messages IEE201I and IEE202I.

(dsname,rdsname)[(dsname,rdsname)]...

The name of one or more data sets to be replaced by the new data set name(s). If any data set name you specify as *(dsname)* is not the name of an in-use local page data set, if any data set name you specify as *(rdsname)* is in use, the system issues messages IEE201I and IEE202I.

Example 1

To delete a local page data set, enter:

```
PD DELETE,PAGE=page3
```

Example 2

To delete three local page data sets, enter:

```
PD DELETE,PAGE=sys1.page01,sys1.page02,page3
```

Example 3

To replace SYS1.PAGE01, a local page data set, and specify SYS1.PAGE04 to replace it, enter:

```
PD REPLACE,PAGE=(sys1.page01,sys1.page04)
```

Example 4

To replace two local page data sets, enter:

```
PD REPLACE,PAGE=(sys1.page01,sys1.page02),(page3,page7)
```

QUIESCE Command

Use the QUIESCE command to put the system in a manual state without affecting job step timing; for example, when you want to alter storage. You can enter QUIESCE only from a console with MASTER authority. You can restart the system by performing the RESTART function.

Syntax

```
QUIESCE
```

If possible, all jobs currently processing terminate normally. Otherwise, current activity is suspended, and the system enters a manual state or a wait state with a code of hexadecimal 80000CCC. See *z/OS MVS System Codes* for more information on wait state code CCC. You might receive the following message on an MCS console or the system console:

BLW002I SYSTEM WAIT STATE 'CCC'X - QUIESCE FUNCTION PERFORMED

Notes:

1. If this system is actively using global resource serialization to share global resources and the global resource serialization complex is not the same as the sysplex, issue a VARY GRS (*),QUIESCE command before issuing the QUIESCE command. Issuing a VARY GRS (*),QUIESCE command before issuing the QUIESCE command prevents the disruption of the global resource serialization ring.
2. Do not issue a SYSTEM RESET after quiescing the system if you intend to issue a RESTART after the quiesce. Issuing a SYSTEM RESET will cause the system to enter an enabled wait state.

REPLY Command

Use the REPLY command to respond to system requests for information. The system associates an identification number with each information request it makes. The REPLY command for a specific request must contain the same identification number as the request. The verb, REPLY or R, is not required when you respond to a request.

When MVS is running in a single system (or in a sysplex configured for eight or fewer systems), reply IDs are assigned in sequential order. For example, four sequential WTORs might be assigned reply IDs 01, 02, 03, 04. An operator monitoring a console that sees all WTORs could expect to see all four reply IDs (01, 02, 03, 04), without skipping a number. If there were multiple consoles that received WTORs, an operator might track down any missing reply IDs and see who replied (or if it was replied to).

In a sysplex configured for greater than eight systems, reply IDs might not appear in sequential order. For example, an operator might see reply IDs 01, 02, 04, 03. This is normal and does not affect system processing.

Table 4-26 summarizes the system requests for information for which you would use the REPLY command.

Table 4-26. Summary of the REPLY Command

Topic:
“Replying to System Information Requests” on page 4-356
“Replying to System Requests During Recovery Processing” on page 4-357
“Replying to System Security WTORs” on page 4-357
“Setting the Time-of-Day Clock” on page 4-357
“Specifying Component Trace Options” on page 4-359
“Specifying Dump Options” on page 4-359
“Specifying SMF Options” on page 4-360
“Specifying System Parameters” on page 4-360

Using System Symbols in REPLY Commands

When system symbols are specified in the REPLY command, the system that receives the WTOR message substitutes text for the system symbols in the response portion (*'text'*) of the command. There are two exceptions to that rule:

- If the WTOR is synchronous, the system does not substitute text for system symbols in the reply *'text'*.
- If the WTOR is issued early in the IPL (while the NIP console is still in use), the system cannot substitute text for system symbols that are not yet processed. If the WTOR is issued after message IEA347A SPECIFY MASTER CATALOG PARAMETER, the system substitutes text for all system symbols.

Note: The system issues message IEE600I in place of message IEE295I for replies during system initialization that are changed by symbolic substitution.

For more information about using system symbols in system commands, see “Using System Symbols in Commands” on page 1-14.

REPLY Command

Scope in a Sysplex

The REPLY command has sysplex scope. See “Using Commands That Have Sysplex Scope” on page 1-11 for an explanation.

Syntax

The general syntax for the REPLY command is:

```
[R] id[,]['text'|text]
```

Replies to System Information Requests

You can use the REPLY command to respond to system requests. To review outstanding requests before replying, issue DISPLAY R.

```
[R] id[,]['text'|text]
```

The parameters are:

id The identification number (0-9999), as specified in the message requesting a response. Leading zeros can be omitted. You may also specify a value of 00.

'text'

The response to the message. The apostrophes are optional and need only be included if your answer contains uppercase and lowercase characters. If you include the apostrophes and your answer contains an apostrophe, use two apostrophes in the message text.

The short form of the REPLY command does not require that you enter either REPLY or R. The short form of the reply command allows you to enter a total length of 124 character spaces. Anything beyond that length is truncated. If the system console is not in problem determination mode, you cannot use the short form of the REPLY command when responding to WTORs from the system console.

The RMAX value can affect the way you enter the short form of the REPLY command. The RMAX value determines the maximum number of REPLY ids that you can use to respond to WTOR messages.

On JES2 systems, when using the short form of the REPLY command, the operator can omit the comma, but the system might misinterpret the command, depending on the RMAX value. For example, if RMAX is 99, and the operator enters the following:

103NONE

MVS interprets the command as follows:

R 10,3NONE

On JES3 systems, an operator must use a comma to separate the REPLY id from the command text:

5,NONE

Example

To use the short form of the REPLY command to reply 'U' to system message 03

On JES2 systems, enter:

3u

On JES3 systems, enter:

3,u

Replies to System Requests During Recovery Processing

During system recovery processing, normal console operations are suspended. The system uses the first available console specified in SYNCDEST to display synchronous WTOR messages. The system will wait 125 seconds for a reply. If there is no reply, the system displays the WTOR messages on the next available console in SYNCDEST. If there is no reply on the next console, the system displays the WTOR messages to the system console. These messages remain displayed until a reply is given on any console.

```
R [00|0][,]['text'|text]
```

Replies to System Security WTORs

System security WTORs are specified with ROUTCDE=9. When you reply to a security WTOR, instead of seeing the actual response text on the display screen, you will see the text "SUPPRESSED" and the SYSLOG and SMF record type 80 reports will also show "SUPPRESSED" instead of the actual response text.

Setting the Time-of-Day Clock

Once the system has been initialized, it can issue one of two messages, depending on whether or not the time-of-day clock is set.

If the time-of-day (TOD) clock is not set, the system asks you to set it:

* 00 IEA886A TOD CLOCK(S) MUST BE SET

Use the following form of the REPLY command to set the time of day clock:

```
R 00,'[DATE=yyyy.ddd][,CLOCK=hh.mm.ss][,UTC|GMT]'
```

Where *yyyy* is the year (1924-2042), *ddd* is the day (001-366), *hh* is the hour (00-23), *mm* is the minute (00-59), and *ss* is the second (00-59). Note that you must specify the year *yyyy* using four digits.

Note: The apostrophes in the above reply are optional.

If you include UTC in your reply, the time and date are Coordinated Universal Time. If you include GMT in your reply, the time and date are Greenwich mean time. The term GMT is obsolete, but will be accepted. Without the UTC or GMT parameter the system assumes the values are the local time and date, converts them to UTC or GMT values, and sets the clock(s) with those values.

When you have entered a valid reply to message IEA886A, the system issues message IEA903A, requesting a response. There are two possible responses, depending on the environment in which MVS is running. The first requests you to

REPLY Command

reply U to message IEA903A and, at the exact time that matches the TOD clock setting, press the TOD clock security switch. The second version does not request you to press the TOD clock security switch. You reply U to message IEA903A and, at the exact time that matches the TOD clock setting, press the ENTER key for the reply text. Once you have successfully set the TOD clock, or if the TOD clock is already set but you are allowed to alter it, the system displays the time and date and gives you the option of accepting or changing them:

```
* id IEA888A UTC DATE=yyyyddd,CLOCK=hh.mm.ss  
IEA888A LOCAL DATE=yyyyddd,CLOCK=hh.mm.ss REPLY U, OR UTC/LOCAL TIME
```

If the values are acceptable, reply 'U'. If you want to change either the local date or time (or both) or the TOD clock, enter the new value(s) as follows, remembering that you must include the UTC or GMT parameter to change the value of the TOD clock:

```
R id,'[DATE=yyyyddd][,CLOCK=hh.mm.ss][,UTC]'
```

Again, the year *yyyy* must have four digits, and the apostrophes are optional.

Note: The system automatically issues message IEA888A at IPL time if the OPERATOR PROMPT parameter is included in the active CLOCKxx parmlib member. (See *z/OS MVS Initialization and Tuning Guide* for details.)

If you specified a different clock setting, the system issues message IEA903A (described above). If you omitted UTC or GMT, the system assumes local date and/or time. Once you have set the new time and/or date, the system re-issues message IEA888A with new values. Reply to the message as described above.

Resetting UTC or GMT time causes the system to reset the TOD clock and recalculate the local time value, using the new UTC or GMT and the system time zone constant.

Resetting local time does not affect UTC or GMT time or the TOD clock. However, it will cause the system to recalculate the system time zone constant (which is initialized at IPL from the CLOCKxx parmlib member). The new time zone constant remains in effect until either local time is modified again or the next IPL.

All of the real time TQEs get adjusted when the local time is updated. Outstanding real time TQEs have their time adjusted based on the local time change that was made. For the external timer reference (ETR), an external interrupt occurs with the time zone offset change, and the timer supervisor code invokes the same TQE time adjustment routine.

If message IEA888A indicates that both UTC or GMT and local time values are incorrect, you should reset the UTC or GMT values first.

Note: You should set the TOD clock to a value based on zero being equivalent to 00 hours, 00 minutes, 00 seconds on January 1, 1900 UTC. During an IPL, the TOD clock might contain a value that, relative to this base, is not correct. This can happen, for example, when a customer engineer (C.E.) left the clock in the error state. In such a case, to ensure that the local time and date are correct, specify UTC or GMT before setting the local time and date.

Specifying Component Trace Options

After you issue a TRACE CT command, the system prompts you for the options you want to specify with message ITT006A. Use the following form of the REPLY command to respond to this message:

```
R id[,ASID=(nnnn[,nnnn]...)]
  [,JOBNAME=(name[,name]...)]
  [,OPTIONS=(name[,name]...)]
  [,WTR={membername|DISCONNECT}]
  [,CONT|,END ]
```

Note: When you specify CONT or END, it must be the last parameter on the input line.

For a detailed explanation of the TRACE CT options, see “Specifying TRACE CT Options” on page 4-607.

Specifying Dump Options

After you issue a DUMP command, the system prompts you for the DUMP options you want to specify with message IEE094D. Use the following form of the REPLY command to respond to this message:

```
R id,U
  or
R id[,ASID=(n[,n]...)] [,JOBNAME=(name[,name]...)] [,TSONAME=(name[,name]...)]
  [,DSPNAME=(dspname-entry[,dspname-entry]...)]
  [,,{PROBDESC|PROB|PD}=key-spec] [,REMOTE=(request[,request]...)]
  [,SDATA[=(option[,option]...)] [,STOR=(beg,end[,beg,end]...)]]
  [,STRLIST=(s-option[,s-option]...)]
  [,CONT|,END]
```

Notes:

1. When you specify CONT or END, it must be the last parameter on the input line.
2. The CONT keyword does not work within a SYSP= list.
3. When you specify U, it must be the first parameter following the identification number.

Where **request** represents:

```
{GRPLIST={group(member)
  {(group(member[,member]...) [,group(member[,member]...)]...) } }

{SYSLIST={sysinfo|(sysinfo[,sysinfo]...)}
  [,DSPNAME|,DSPNAME=(dspname-entry[,dspname-entry]...)]
  [,SDATA|,SDATA=(option[,option]...)]
  [,STOR|,STOR=(beg,end[,beg,end]...)] }
```

REPLY Command

Where **s-option** represents:

```
STRNAME=strname
[,CONNAME=connname]
[,ACCESSTIME={ENFORCE|NOLIMIT|NOLIM}]
[,LOCKENTRIES]
[,USERCNTLS]
[,EVENTQS]
[,({EMCONTROLS={ALL|(list)})}
[,({COCLASS|STGCLASS|LISTNUM}={ALL|(list)}
{[,ADJUNCT={CAPTURE|DIRECTIO}] [,ENTRYDATA={UNSERIALIZE|SERIALIZE}]}
{[,SUMMARY]})]
```

For a detailed explanation of the DUMP options, see “Specifying Dump Options” on page 4-263.

Specifying SMF Options

If PROMPT (LIST or ALL) is specified in the SMFPRMxx parmlib member, you can change the SMF options, and you receive this message:

IEE354I SMF PARAMETERS

The system lists the SMF parameters specified at system installation time and issues the following message:

* id IEE357A REPLY WITH SMF VALUES OR U

```
R id,{U
      {options}}
```

Respond to message IEE357A by specifying that you do not want to change the SMF values (U) or by specifying the options separated by commas as provided by your system programmer. A description of these parameters can be found in *z/OS MVS System Management Facilities (SMF)*.

If PROMPT (IPLR or ALL) is specified, the system issues message IEE956A when an IPL occurs. This message prompts you to reply with the time the failure occurred, the name of the operator, and the reason for the IPL.

Once specification of system options is complete, JES2 or JES3 is automatically invoked if your installation has provided for that capability.

Specifying System Parameters

During system initialization, you receive the following message:

IEA101A SPECIFY SYSTEM PARAMETERS FOR RELEASE xx.yy.zzz

where xx is the release number, yy is the release level, and zzz is the system type.

You must reply to this message with the REPLY command. The short form reply is valid. You can accept the default system parameters by using the following form of the REPLY command with the 'U' operand. However, if your system programmer has given you parameters to enter, use the following form of the REPLY command to enter them.

```
R [0|00],{U
  {
    {{ 'parm=' }} [,CONT]
    {{ 'parm=parm,' }} []
    {{ 'parm=value' }} []
    {{ 'parm=(value[,value]...[,L])' }} []
    {{ 'parm=(value[,value]...[,L]),parm=value' }} []
```

The parameters are:

- 00** The identification number (00) as specified in the message requesting information. A single zero can be used.
- U** No parameters are to be changed. The system uses the default list of system parameters in the parmlib data set.

'parm=,'

The parameter, as specified in the parmlib data set, is to be canceled for this IPL. If a system default exists for this parameter, it is used.

'parm=,parm,'

The parameters, as specified in the parmlib data set, are to be canceled for this IPL. If system defaults exist for these parameters, they are used.

'parm=value'

'parm=(value[,value]...[,L])'

'parm=(value,value), parm=value'

The specified parameters are to override the corresponding parameters in the parmlib data set. When specifying system parameters, (1) A blank or comma must separate multiple parameters and (2) U is not a valid value for a parameter. The reply can be at most 80 characters per line. If the reply is longer than one line, follow the last parameter with a comma or a blank and CONT. The system prompts you for the remaining values. Example:

```
R 00,'MLPA=(00,01,02,CONT'
IEA116A CONTINUE SYSTEM PARAMETERS
R 00,'03,L),BLDL=02'
```

- L** List the parameters as they are processed.

If you are uncertain of the format of a system parameter, see *z/OS MVS Initialization and Tuning Reference* or ask your system programmer.

RESET Command

Use the RESET command to:

- Force a hung MCS or SMCS console into the offline state
- Change the performance group of a non-privileged job currently in execution.

Note: Do not use the RESET command to change the performance group of a privileged job. The system assigns privileged jobs to a special performance group (0) and, therefore, will ignore any PERFORM value for such jobs.

- Change the service class of work currently in execution
- Quiesce a problem job or address space
- Resume a quiesced job or address space

Note: All of the above RESET tasks, except forcing a hung MCS or SMCS console into the offline state, cause SMF to create an SMF 90 subtype 30 record to log the reset operation.

Table 4-27 summarizes the functions that the RESET command provides.

Table 4-27. Summary of the RESET Command

Command:	Topic:
RESET CN	"Forcing a Hung MCS or SMCS Console Offline"
RESET jobname	"Changing Service Classes or Quiescing Work" on page 4-363

Scope in a Sysplex

The RESET command has sysplex scope only when you specify the CN parameter. See "Using Commands That Have Sysplex Scope" on page 1-11 for an explanation of sysplex scope.

Issue the command from the system where the console was attached to avoid inconsistent sysplex results.

Syntax

The syntax for each variation of the RESET command is shown immediately preceding its respective parameter list.

RESET or E

Forcing a Hung MCS or SMCS Console Offline

There are two forms of the RESET command. Use the following form to force a hung MCS or SMCS console into the offline state.

E [CN(consname)]

CN(*consname*)

Specifies the hung MCS or SMCS console device that the system is to force into the offline state. The console name can be 2 to 8 characters in length.

After issuing the RESET CN command, you might need to vary the console between the offline state and the console state to complete device recovery. Issue this command only after exhausting all other means of console recovery such as:

- Verifying the physical path to the device
- Verifying the control unit the device is attached to
- If the device is attached to a channel extender, verifying that the channel extender is in working order
- Issuing a VARY OFFLINE command that fails
- Issuing a SWITCH CN command that fails.

Note: The SWITCH CN command is not supported on systems at z/OS V1R8 and higher.

For SMCS consoles you should issue a VARY NET,INACT,ID=consolelu command. If the console is not deactivated, try a VARY NET,INACT,ID=consolelu,I command. If that is not successful, try a VARY NET,INACT,ID=consolelu,F command.

Resetting an MCS or SMCS console may take up to 30 seconds to complete. If it is taking a long time to process the command, the system may issue this message: IEE059I RESET CN(consname) COMMAND IN PROGRESS, to indicate that the command is in progress but cannot complete quickly. The system will issue this message: IEE712I RESET PROCESSING COMPLETE, after the command finally completes.

Changing Service Classes or Quiescing Work

Important

Beginning with z/OS V1R3, WLM compatibility mode is no longer available. Accordingly, you can no longer use the RESET command to change the performance group of a job currently in execution. The information has been left here for reference purposes, and for use on backlevel systems.

Use the following form of the RESET command for systems running in workload management compatibility mode, to change the performance group of a job currently in execution. For systems running in workload management goal mode, use this command to:

- Change the service class of work currently in execution, with the SRVCLASS keyword. Resetting to a new service class also resumes quiesced work.
- Quiesce work currently in execution, with the QUIESCE keyword.
- Reclassify work currently in execution according to the service policy in effect, with the RESUME keyword. If there had been a prior quiesce you can specify a quiesced piece of work and the RESUME keyword to reclassify the work and resume processing.

Workload management goal mode refers to the management of resources according to the goals specified in the active workload management policy. Workload management compatibility mode refers to the management of resources according to the values in the IEAICSxx and IEAIPSxx parmlib members.

RESET Command

The RESET command causes SMF to create an SMF 90 subtype 30 record to log the reset operation.

Use this command only at the direction of the system programmer.

The syntax for this form of the the RESET command is:

```
E jobname[,A=asid],{PERFORM=nnn      }
                  {SRVCLASS=classname}
                  {QUIESCE|Q      }
                  {RESUME         }
```

jobname

The name of the job, time-sharing user, or started task whose performance characteristics you want to change. This command affects the current job step and all subsequent job steps in this execution.

A=asid

The hexadecimal address space identifier (ASID) of the job, time-sharing user, or started task you want to change. **A=asid** applies in both workload management goal mode and workload management compatibility mode. You can specify this keyword before or after the PERFORM=, SRVCLASS=, QUIESCE, or RESUME parameters.

This parameter is required if there are two jobs running which have the same *jobname*.

PERFORM=nnn

The performance group number, between 1 and 999, to be assigned to the job, user, or started task. **PERFORM** applies only on a system operating in workload management compatibility mode. It does not apply in workload management goal mode.

SRVCLASS=classname

The name of the service class to be assigned to the job or address space. Resetting to a new service class also resumes quiesced work. **SRVCLASS** applies only on a system operating in workload management goal mode. It does not apply in workload management compatibility mode.

When you issue a RESET against a server (for example, an address space) to a new service class, the goals associated with that service class are ignored. However, the resource group associated with the new service class is honored. The one exception is the case where the goal for a server is honored when the transactions it is serving have been assigned a discretionary goal.

There may be special circumstances under which you would wish to reset an address space with a SYSTEM or SYSSTC service class. See the “Defining Classification Rules” chapter in *z/OS MVS Planning: Workload Management* for information about the use of the SYSTEM and SYSSTC service classes.

RESET SRVCLASS= will remain in effect until one of the following occurs:

- End of job
- The policy is switched to a new policy in which the target service class has been deleted
- The WLM mode is switched to COMPAT
- A RESET RESUME command is issued.

QUIESCE|Q

Requests that the target job or address space be quiesced; that is, given the

lowest possible performance characteristics. QUIESCE swaps out swappable work, effectively shutting off that work. QUIESCE just lowers the performance of non-swappable work, leaving it swapped in. QUIESCE applies only on a system operating in workload management goal mode. It does not apply in workload management compatibility mode.

RESET QUIESCE will remain in effect until one of the following occurs:

- End of job
- The WLM mode is switched to COMPAT
- A RESET RESUME command is issued.

RESUME

Specifies that a job or address space be reclassified. If the job or address space was quiesced by a previous RESET jobname,QUIESCE command, or if the job or address space was assigned to a different service class, RESUME causes the work to be reclassified according to the service policy in effect and resumes processing at the performance targets specified in the service policy. RESUME applies only on a system operating in workload management goal mode. It does not apply in workload management compatibility mode.

The classification rules used are those in effect at the time the RESET command is issued.

RESET PERFORM will remain in effect until one of the following occurs:

- End of job
- The WLM mode is switched to GOAL.

Example 1

To change the performance group value of job TMASGJ02 to 1, enter:

```
e tmasgj02,perform=1
```

The system responds with:

```
IEE304I TMASGJ02      JOB RESET
```

If the system is operating in workload management goal mode the above command is rejected.

The system responds with:

```
IRA701I RESET KEYWORD PERFORM  NOT VALID IN GOAL MODE
```

Example 2

If there are two jobs running with the name of JLKSORT1 and you want to change the performance group value of one of them to a value of 6, first determine the ASID of the address space associated with the job by using either the DISPLAY JOBS,jobname or DISPLAY JOBS,ALL command. If, for example, you are in workload management compatibility mode and the ASID of the one you want to change is 1A8, enter:

```
e jlksort1,perform=6,a=1a8
```

Example 3

To assign the performance goals associated with service class QUICK to job BEEMER, enter:

```
e beemer,srvclass=quick
```

RESET Command

Example 4

To assign the lowest system performance goals to job MARCUS, enter:
e marcus,quiesce

Example 5

To resume execution of job MARCUS in the service class specified by the active service policy (after a previous QUIESCE request) enter:
e marcus,resume

Example 6

To reclassify a job according to the service policy in effect, enter:
e beemer,resume

Once this is done, job BEEMER no longer has the service class QUICK as assigned in Example 3 above.

Restrictions

- The MASTER address space, idle INIT / ASCHINT initiator address spaces, and the WLM address space may not be reset.
- In compatibility mode:
 - Attempts to reset an address space with the privileged or high dispatching priority attributes are rejected with message IRA702I RESET NOT VALID.
- In goal mode:
 - There are no restrictions for the RESET command when the originating and target service classes are both customer-defined.
 - Attempts to move a privileged or high dispatching priority address space into a customer-defined service class are rejected with message IRA702I RESET NOT VALID.
 - With APAR OA12625 installed, attempts to move a privileged address space into a customer-defined service class are allowed and no longer rejected with message IRA702I.
 - RESET **can** be used to move eligible address spaces (not just started tasks) from a customer-defined service class to the SYSSTC service class. If the address space originally was privileged, the privileged attribute is restored.
 - RESET **can** be used to move started tasks eligible for high dispatching priority into the SYSTEM service class. The high dispatching attribute is restored when the started task is moved into SYSTEM.
- Once a mode switch occurs (F WLM,MODE=), the system does not “remember” that jobs had previously been reset. Instead, classification is carried out according to the values in the IEAICSxx and IEAIPSxx parmlib members if you have switched to compatibility mode. You must issue a SET ICS=xx and SET IPS=xx. See “Changing Workload Manager Resource States” on page 4-337 for important information on what to do following a switch to compatibility mode. If you have switched to goal mode, the active service policy is used to assign a service class to the work.

ROUTE Command

Use the ROUTE command to direct a command to one or more systems in a sysplex for processing. You can direct a command to:

- All systems in the sysplex
- A subset of the systems in the sysplex
- One system in the sysplex.

You can enter this command from any MCS, SMCS or extended MCS console with INFO authority.

You can enter most system commands using the ROUTE command, including MVS, JES2, JES3, and other commands.

For most system commands routed to multiple systems, the system combines the command responses into an aggregated response. The combined response sorts the command responses by system name. For more information, see “How MVS Displays Aggregated Response from ROUTE” on page 4-368.

You cannot send more than one command on a single invocation of the ROUTE command. If you need to route multiple commands in strict sequential order, you should route one command, wait for successful response from all systems to which you routed the command, and then route the next command.

Restrictions

The following restrictions apply to this command:

1. When you specify *ALL or *sysgrpname* on the ROUTE command, do not also specify the following commands on that ROUTE command:
 - A ROUTE command itself
 - Commands that display or change an MCS or SMCS console’s attributes, such as:

D PFK
K A
K C
K D
K E
K N
K Q
K S
K V,USE
 - Commands that specify “*”, where “*” means the console issuing the command, such as:

D C,*
V CN(*),ACTIVATE
 - The DUMP command.

Note: Instead of specifying the DUMP command on the ROUTE command, you can request a remote dump, which does not use the ROUTE command. See the DUMP command and the associated REMOTE REPLY option.

- Commands that display an excessive amount of data, such as:
D U,,,100,999
- Commands that are sysplex-wide in scope. For example, the following command would show identical data for each system in the sysplex:
D XCF

ROUTE Command

2. Do not issue ROUTE commands to multiple systems at a sysplex rate that exceeds the rate indicated by the ROUTTIME value. For example, if the ROUTTIME is the default 30 seconds, issuing ROUTEs to multiple systems in the sysplex at a rate faster than 1 per half-minute could lead to resource contention and delays in processing the commands.
3. Do not use the L keyword abbreviation with the ROUTE command when other commands you are routing contain an L keyword abbreviation.
4. If your ROUTE command has the keyword parameter "L=", the result may not be what you want. The first "L=" will be interpreted as a parameter on ROUTE, not on the ROUTEd command.

How MVS Displays Aggregated Response from ROUTE

When you route a command to multiple systems, command responses are returned to the issuing console as part of message IEE421I.

If an out-of-line display area exists on the issuing console, IEE421I is written to the out-of-line display area.

If the L= parameter is specified on a ROUTE *ALL or ROUTE *sysgrpname* command, the aggregated response will be redirected as specified by the L= parameter.

Note: Do not use system symbols on the L= parameter when aggregating command responses.

MVS returns an aggregated response when one of the following occurs:

- MVS has received at least one response from all systems and a period of time has elapsed during which it has received no additional responses. MVS calculates this period of time based on the pattern of responses received for the command.
- The maximum timeout interval in effect for the ROUTE command is reached.

For command responses to be aggregated, the maximum timeout interval that is in effect for the ROUTE command cannot be zero. The maximum timeout interval in effect for a ROUTE command is determined as follows:

1. The current sysplex-wide default, which is determined as follows:
 - a. The IBM-defined default for the maximum timeout interval is 30 seconds. This can be changed by any of the following conditions.
 - b. An installation-defined default for the maximum timeout interval can be specified in the CONSOLxx parmlib member of the first system to join the sysplex. (This is specified on the ROUTTIME keyword of the INIT statement.) This installation-defined default applies to all other systems joining the sysplex.
 - c. The operator can dynamically change the default maximum timeout interval for all systems in the sysplex by issuing the CONTROL M,ROUTTIME= command.
 2. When entering the ROUTE command, the operator can specify the maximum timeout interval in effect for that one time by specifying the T= operand on the ROUTE command itself.

For a specific command response to appear in the aggregated response, individual responses to the routed command must meet these criteria:

- The command processor for the routed command must direct command response(s) to the console ID where the command originated.
- If, when issuing a command response, a command processor does not use the console ID of the command issuer, MVS cannot return the command responses to the ROUTE command issuer.
- The specific command response must be received at the originating system before the originating system stops aggregating responses.

Command responses that do not meet the above criteria are not aggregated. However, if MVS receives command responses after the timeout period, MVS attempts to return the responses to the originator of the ROUTE command.

Using System Symbols in ROUTE Commands

You can specify system symbols in commands that are routed to one or more systems in a sysplex. This section explains special considerations for using system symbols in routed commands. Before you read this section, see “Sharing System Commands” on page 1-14 for information about how to use system symbols in commands and lists of system symbols that the system provides.

When you enter a ROUTE command, the system views the command in two parts:

- The actual ROUTE command, which indicates where and how the command is to be routed. The system on which the ROUTE command is entered processes the system symbols in this part of the command.
- The command that is to be processed on one or more other systems. The system to which the command is routed processes the system symbols in this part of the command.

Because several systems can be involved in processing a ROUTE command, **IBM recommends that you enter a DISPLAY SYMBOLS command on each system that is to process parts of the ROUTE command containing system symbols.**

DISPLAY SYMBOLS shows the current static system symbols and their associated substitution texts. See the description of the DISPLAY SYMBOLS command in this book for more information.

For example, suppose that the following values are defined for system symbols on systems SYS1 and SYS2:

System	System Symbol	Value
SYS1	&T1	999
SYS1	&SYSNAME2	SYS2
SYS1	&SYSCLONE	S1
SYS2	&T1	0
SYS2	&SYSNAME	SYS2
SYS2	&SYSCLONE	S2

Then suppose you enter the following ROUTE command on system SYS1:

RO T=&T1,&SYSNAME2,F JOB&SYSCLONE,*parameters*

Processed by SYS1 Processed by SYS2

ROUTE Command

The systems process the command in the following way:

1. The system on which the ROUTE command is entered, SYS1, substitutes text for the &T1 and &SYSNAME2 system symbols in the first part of the command.
2. The system to which the command is to be routed, SYS2, substitutes text for the &SYSCLONE system symbol in the second part of the command.

The result of the substitution is:

```
RO T=999,SYS2,F JOBS2,parameters
```

If a system group had been specified in place of a system name in the ROUTE command, the MODIFY command would have been routed to all systems in that group. Each system would have substituted text for system symbols in the instance of the command that was routed to that system.

Syntax

The complete syntax for the ROUTE command is:

```
RO {sysname, text }  
{ [T=nnn] {*ALL } [,L={a } ] }  
{ sysgrpname } {name }  
{*OTHER } {name-a }  
{(sysname[,sysgrpname,sysname...])}
```

Parameters

sysname

The system name (1 to 8 characters) that will receive and process the command.

MVS returns the command response to the issuing console (inline area for an MCS or SMCS console) unless redirected by the L= parameter on the routed command.

text

The system command and specific operands of the command being routed. Do not code any leading blanks before the text.

T= Specifies an optional timeout interval. T= is valid with *ALL, *OTHER, sysgrpname, or a list of system names or sysgrpnames. You can specify a value from 0 to 999. This value indicates the maximum number of seconds MVS waits for responses from each system before aggregating the responses.

If you specify T=0, MVS does not aggregate command responses, but individually sends responses to the originator.

Notes:

1. IBM recommends that you specify T=0 when you are routing the START and STOP commands to multiple systems. This is because the system does not collect aggregate responses for routed START and STOP commands. If you attempt to do so (if T= is nonzero), the system states that there is "no response" from all of the systems, and all the START and STOP command responses are displayed inline.
2. IBM does not recommend that you specify T=0 for most DISPLAY commands. Command responses from most DISPLAY commands appear in

an out-of-line display area, and the responses from multiple DISPLAY commands can be written into the same area one right after the other, so that only the last one is readable. If there is no display area defined, or if L=Z is used, the responses are inline, but will probably roll off the console. Responses from ROUTE with T=0 and a DISPLAY command specified could be useful to an automation program and as a hardcopy record, but not for a human operator.

***ALL**

Specifies that the command is to be routed to all systems in the sysplex.

***OTHER**

Specifies that the command is to be routed to all systems in a sysplex *except* the system on which the command is entered.

If you enter a ROUTE *OTHER command on a system that is not a member of a sysplex, or if you enter ROUTE *OTHER from a system that is a member of a sysplex in which no other systems are active, MVS issues message IEE413I.

sysgrpname

Specifies that the command will be routed to a subset of systems in the sysplex. The *sysgrpname* can be 1 to 8 characters and represents the set of systems to which the command is to be routed. System group names are defined by the installation. For information on defining system group names, see *z/OS MVS Planning: Operations*.

(sysname[,sysgrpname,sysname...])

Specifies that the command is to be routed to a list of systems or system groups, or both. System group names are defined by your installation.

If you route a command to a list of systems or system groups, and *none* of the systems or system groups is active, the system issues message IEE413I.

L=a, name, or name-a

Specifies the display area (*a*), console name (*name*), or both (*name-a*) where the display will appear.

For the ROUTE command, you can specify the L= operand as follows:

- When routing a command to only one system, you can specify the L= operand only if supported by the routed command. For example, the following is valid because the D C command supports the L= operand:

```
R0 sys1,D C,L=con1
```

The following is not valid because the D T command does not support the L= operand:

```
R0 sys1,D T,L=con1
```

- When routing a command to all systems, or to a named subset of systems, the L= operand is supported if aggregation of responses occurs (the timeout interval is not zero). For example, the following is valid:

```
R0 T=20,*ALL,D T,L=con1
```

If you do not specify the L= option, the system displays the command responses in the first available display or message area of the console on which you entered the ROUTE command.

Note: Do not use system symbols on the L= parameter when aggregating command responses. For more information about specifying system symbols when routing commands, see “Using System Symbols in ROUTE Commands” on page 4-369.

ROUTE Command

Example 1

To route a DISPLAY UNITS command for device 320 on system SY4 and have the response returned to the issuing console, enter:

```
ROUTE SY4,D U,,,320,1
```

Example 2

To route a DISPLAY UNITS command to system SY1 and have the response returned to the console named CON1A, enter:

```
ROUTE SY1,D U,L=CON1A
```

Example 3

To change the OPNOTIFY value on all systems in the sysplex, enter:

```
ROUTE *ALL,SETXCF COUPLE,OPNOTIFY=15
```

Example 4

To start JES2 on all systems in the sysplex, enter:

```
ROUTE T=0,*ALL,S JES2
```

Note: The system does not aggregate command responses for routed START and STOP commands. If you attempt to do so (if T= is nonzero), the system states that there is “no response” from all of the systems, and all the START and STOP command responses are displayed inline.

Example 5

After JES3 initialization completes on the global, to start JES3 on all local processors in the sysplex, from a console associated with the global, enter:

```
ROUTE T=0,*OTHER,S JES3
```

Note: The system does not aggregate command responses for routed START and STOP commands. If you attempt to do so (if T= is nonzero), the system states that there is “no response” from all of the systems, and all the START and STOP command responses are displayed inline.

Example 6

To issue D A on system SY1 and system SY4, where TEST is a system group name representing both system SY1 and system SY4, enter:

```
ROUTE TEST,D A
```

Example 7

To issue D A on the systems represented by TEST and have the aggregated responses received within 10 seconds, enter:

```
ROUTE T=10,TEST,D A
```

Example 8

To issue D T (DISPLAY TIME) to be processed on system SYS1, enter:

```
ROUTE SYS1,D T
```

The system returns this information:

```
SYS1 IEE136I LOCAL: TIME=09.59.49 DATE=1993.257 GMT: TIME=13.59.43
DATE=1993.257
```

Example 9

To route the command D T (DISPLAY TIME) to be processed on all systems (SYS1, SYS2, and SYS3), enter:

```
ROUTE T=0,*ALL,D T
```

The system returns the following information:

```
SYS1 IEE136I LOCAL: TIME=10.00.59 DATE=1993.257 GMT: TIME=14.00.59
DATE=1993.257
SYS3 IEE136I LOCAL: TIME=10.00.59 DATE=1993.257 GMT: TIME=14.00.59
DATE=1993.257
SYS2 IEE136I LOCAL: TIME=10.00.59 DATE=1993.257 GMT: TIME=14.00.59
DATE=1993.257
```

Notes:

1. Because T=0 is specified, the information is not aggregated, but individually returned.
2. Most DISPLAY commands produce multi-line command responses in an out-of-line display area on the console. Such command responses are written into the same area. IBM does not recommend specifying such DISPLAY commands on the ROUTE command when T=0 is in effect. However, because the D T (DISPLAY TIME) command produces a single-line command response, you can successfully specify the D T command on the ROUTE command, as shown in this example.

Example 10

To route the command D T (DISPLAY TIME) to all systems and receive an aggregated response, use the T= parameter and enter:

```
ROUTE T=5,*ALL,D T
```

The system returns the following **aggregated** information:

IEE421I RO *ALL,D T	FRAME	LAST	F	E	SYS=SYS1
SYSNAME RESPONSES -----					
SYS1 IEE136I LOCAL: TIME=10.06.48 DATE=1993.257					GMT:
TIME=14.06.48 DATE=1993.257					
SYS2 IEE136I LOCAL: TIME=10.06.48 DATE=1993.257					GMT:
TIME=14.06.48 DATE=1993.257					
SYS3 IEE136I LOCAL: TIME=10.06.48 DATE=1993.257					GMT:
TIME=14.06.48 DATE=1993.257					

Example 11

This example illustrates what can happen when the timeout interval occurs before all the command responses are received for aggregation. If you issue the ROUTE command to vary device 414 offline on all systems:

```
ROUTE T=1,*ALL,V 414,OFFLINE
```

and system SYS2 does not respond within one second (the timeout interval) MVS cannot include the command response from SYS2 with the other command responses.

- First, MVS lists the systems from which no response was received in time for aggregation.

ROUTE Command

```
IEE421I RO *ALL,V 414,OFFLINE
NO RESPONSE RECEIVED FROM THE FOLLOWING SYSTEM(S):
SYS2
• After the operator scrolls forward to the second frame of message IEE421I, MVS
displays the aggregated response:
IEE421I RO *ALL,V 414,OFFLINE
SYSNAME  RESPONSES -----
SYS1      IEF281I 0414 NOW OFFLINE
SYS3      IEE303I 0414     OFFLINE
```

Example 12

Route the \$SPRT1 command to all systems in a sysplex except the system on which the command is entered:

```
ROUTE *OTHER,$SPRT1
```

Example 13

Quiesce systems S0 and S9 from system S1:

```
RO (S0,S9),QUIESCE
```

Example 14

Quiesce systems S0 and the subset of systems represented by system group name G8:

```
RO (S0,G8),QUIESCE
```

If the ROUTE *OTHER command is issued on a system which is not a member of a SYSPLEX, or a system that is a member of a sysplex but no other systems are active members in the sysplex, a message is issued (see messages section for details). This same message will be issued if a list of systems/groups is specified and none of the systems/groups represent at least one active system.

SEND Command

Use the SEND command to communicate with other operators in a multiple-console support (MCS) and SNA multiple-console support (SMCS) environment. In a time-sharing environment, use the SEND command to communicate with a specific terminal user or all terminal users, and to manage the SYS1.BRODCAST data set. Table 4-28 summarizes the information that the SEND command provides. Use it to find details about a particular use of the SEND command.

Table 4-28. Summary of the SEND Command

Command:	Topic:
SEND ...,BRDCST SEND ...,OPERATOR=... SEND ...,CN=...	“Communicating with Other Operators”
SEND ...,USER=...	“Communicating with Specified Users” on page 4-376
SEND ...,LOGON SEND ...,NOW	“Communicating with All Logged-On Terminal Users” on page 4-377
SEND ...,SAVE	“Saving Messages in the Broadcast Data Set” on page 4-379
SEND ...,LIST	“Listing the Notices Section of the Broadcast Data Set” on page 4-379
SEND ...,DELETE	“Deleting a Message from the Broadcast Data Set (Notices Section)” on page 4-381

Scope in a Sysplex

The SEND command has sysplex scope only when sending to consoles; SEND does not have sysplex scope when sending to TSO users. See “Using Commands That Have Sysplex Scope” on page 1-11 for an explanation of sysplex scope.

Syntax

The syntax for each variation of the SEND command is shown immediately preceding its respective parameter list.

SEND or SE

Communicating with Other Operators

Use the SEND command to communicate with other operators at MCS and SMCS consoles.

<pre>SE { 'message' },{BRDCST {msgno } {OPERATOR=routecode} {CN=console }</pre>

The parameters are:

'message'

The message to be sent.

msgno

The number of the message in SYS1.BRODCAST to be sent. (See “Saving

SEND Command

Messages in the Broadcast Data Set" on page 4-379 for information on how to save and later send messages by message number.)

BRDCST

The specified message is to be sent to all active consoles that have not specified the 'no broadcast' option (using the K V,LEVEL command).

OPERATOR=routecode

The installation area (such as tape library) to receive the message, specified as a one or three digit number between 1 and 128 (see Table 3-6 on page 3-30).

CN=console

The name of the console where the message is to be sent. The console name is 2 to 8 alphanumeric characters; the first character must be alphabetic or one of the following characters: \$, #, or @. Ensure that MCS and SMCS consoles are already defined by your installation.

When you are uncertain of the routing codes and console identifiers in effect, enter the DISPLAY CONSOLES command. See "Displaying Console Status Information" on page 4-112.

Example 1

To send message number 46 to the console named CON12, enter:

```
se 46,cn=CON12
```

Example 2

To send the following message to all active consoles, enter:

```
se 'Close down in 15 minutes',brdcst
```

Communicating with Specified Users

Use the SEND...,USER command to communicate with specific time-sharing users.

```
SE {'message'},USER=(userid[,userid]...),{NOW|LOGON},{WAIT|NOWAIT}  
{msgno}
```

The parameters are:

'message'

The message to be sent to the terminal users.

msgno

The number of the message to be sent. (See "Saving Messages in the Broadcast Data Set" on page 4-379. for information about how messages can be saved and later sent by message number.)

USER=(userid[,userid]...)

The identifiers of those users who are to receive the message.

NOW

Specifies that the message is to be sent immediately. If the recipient is not logged on, you are notified and the message is deleted.

When NOWAIT and USER are specified and the user's terminal is busy:

- The user does not receive the message
- You are notified which users did not receive the message

- The message is deleted

LOGON

If any specified user is currently logged on and is accepting messages, the user receives the message. If the user is logged on but is not receiving messages, the message is stored in the mail section of the broadcast data set until the user requests it. If the user is not logged on, the message is stored in the mail section of the broadcast data set until requested when the user logs on.

When NOWAIT is specified and the user's terminal is busy, the message is stored in the mail section of the broadcast data set until the user requests it.

WAIT

Specifies that the message is held until system output buffers are available for the specified logged on users. This option ensures that the message is received by all the specified users. When a user's terminal is busy, other users will not receive the message until that user's terminal is free.

NOWAIT

Specifies that the message is not held. When USER is specified, you are notified of any users who do not receive the message. If LOGON is specified, the message is saved as mail for those user's who's terminal is busy or who were not logged on.

Note: When possible, use the LOGON parameter so you do not interrupt the user's terminal session unnecessarily.

Example 1

To send the following message to users D58 and D04 immediately, if they are receiving messages, or when they request messages, enter:

```
se 'your listings are ready',user=(d58,d04),logon
```

If they are not logged on the system, they receive the message when they log on.

Example 2

To send the message to the specified user immediately, if he is logged on, enter:

```
se 'getting I/O errors on your pack',user=(payroll)
```

Communicating with All Logged-On Terminal Users

You can use the SEND command to send a message to all terminal users currently logged on the system.

```
SE {'message'}, {NOW|LOGON}, {WAIT|NOWAIT}, {ROUTE={*ALL|systemname|groupname}}
{msgno}
```

The parameters are:

'message'

The message that is to be sent to all time-sharing terminal users.

msgno

The number of the message to be sent. (See "Saving Messages in the Broadcast Data Set" on page 4-379 for information about how to save and later send messages by message number.)

SEND Command

NOW

Specifies that the message is to be sent immediately to all users currently logged on; the message is not retained for users not logged on.

When NOWAIT is specified and the user's terminal is busy:

- the user does not receive the message
- you are NOT notified which users did not receive the message
- The message is deleted

LOGON

All users logged on and accepting messages receive the message. Those users logged on but not receiving messages receive it upon requesting messages.

The message is stored in the notices section of the broadcast data set and is sent to those users requesting messages when they log on. The message is retained until you delete it.

When NOWAIT is specified and the user's terminal is busy, the message is stored in the mail section of the broadcast data set until the user requests it.

WAIT

Specifies that the message is held until system output buffers are available for the specified logged on users. This option ensures that the message is received by all the specified users. When a user's terminal is busy, other users will not receive the message until that user's terminal is free.

NOWAIT

Specifies that the message **not** be held. If you specify LOGON, the system saves the message as mail for those users whose terminals are busy or who were not logged on.

ROUTE

Sends the message to all users logged onto the specified system(s). If you do not specify the ROUTE= parameter, the system sends the message only to the users logged onto the system where you issue the SEND command. Valid values for the ROUTE parameter are:

***ALL**

Directs the system to send the message to all users logged onto all systems participating in the sysplex

systemname

Directs the system to send the message only to users logged onto *systemname*

groupname

Directs the system to send the message to all users logged onto the sysplex subset defined by *groupname*

Note: When possible, use the LOGON parameter so you do not interrupt the users' terminal sessions unnecessarily.

Example 1

To send the following critical message to all users immediately, enter:

SE 'system going down in 5 minutes'

Example 2

To send the following general interest message to users when they request messages or at LOGON time, enter:

```
SE 'time-sharing will not be up next Thursday',LOGON
```

Saving Messages in the Broadcast Data Set

Use the SEND...,SAVE command to store messages in the broadcast data set to be issued only at LOGON time, or when requested.

```
SE { 'message' },{USER=(userid[,userid]...)},SAVE
{ msgno } {ALL}
```

The parameters are:

'message'

The message to be sent to the terminal users.

msgno

The number of the message to be sent.

USER(userid)

The identifications of those users to receive the message. The message is stored in the mail sections for those users. (There is a mail section for each TSO user.)

ALL

All terminal users are to receive the message. Terminal users who are currently using the system receive the message immediately. In addition, the message is placed in the notices section and assigned a number. This number, printed when the message is stored, can be used as *msgno* in this and other forms of the SEND command.

SAVE

The message is to be stored in the appropriate section of the broadcast data set until a user logs on or requests messages. If ALL is specified, the message is stored in the notices section of the broadcast data set and is retained until explicitly deleted. If *userid* is specified, the message is stored in the mail section of the broadcast data set and deleted after it is sent to the intended user. No attempt is made to send it immediately, even to those users logged on and receiving messages.

Note: WAIT and NOWAIT have no effect when specified with SAVE.

Example

To submit messages to the broadcast data set before stopping time-sharing for the day, enter:

```
se 'time-sharing will close down at 5:00 p.m. today.',save
```

When you start time-sharing the next day, the messages are available for users logging on. The above command does not affect those users currently logged on and receiving messages.

Listing the Notices Section of the Broadcast Data Set

Use the SEND...,LIST command to keep track of accumulated messages in the notices section of the broadcast data set. You can list one or all of the messages.

SEND Command

```
SE [msgno,]LIST
```

The parameters are:

msgno

The number of the message to list. Omitting this operand results in all messages in the notices section, and the message numbers assigned to them, being listed.

LIST

The requested message or all messages in the notices section of the SYS1.BROADCAST data set are to be listed on the console.

Example 1

To list all messages in the notices section of the SYS1.BRODCAST data set, enter:
SE LIST

Example 2

To list message number 21, enter:
SEND 21,LIST

Deleting a Message from the Broadcast Data Set (Notices Section)

If you find, after listing the notices section of the broadcast data set, that a message is no longer needed, use the SEND...,DELETE command to delete it.

```
SE msgno,DELETE
```

The parameters are:

msgno

The number of the message to be deleted.

DELETE

The specified message is to be deleted.

Example: To delete message number 23, enter:

```
SE 23,DELETE
```

SET Command

Use the SET command to:

- Set or reset the local time and date.
- Change the local time offset value.
- Change the storage management subsystem (SMS) parameters, or start SMS if it was not started at IPL, or restart SMS if it could not be automatically restarted.
- Specify the method that VSAM Record Level Sharing (RLS) uses to determine the size of the data placed in the coupling facility cache structure.
- Change the system resources manager (SRM) parameters.
- Change messages processed by the message processing facility (MPF), or the color, intensity, highlighting, and foreign message handling options.
- Change the system management facilities (SMF) parameters or restart SMF.
- Change the dump analysis and elimination (DAE) parameters.
- Change the commands SLIP is to process.
- Change the command installation exits the system is to use.
- Change the set of available PFK tables.
- Change all the MIH time intervals.
- Change all the I/O timing limits.
- Change the status of FICON switch statistics gathering.
- Change the residency of where IOS storage blocks are obtained.
- Change the excessive spin recovery actions.
- Change the spin loop timeout interval.
- Change or refresh the MVS message service (MMS) parameters.
- Change the GRS resource name lists (RNLs).
- Start, refresh, or stop MMS.
- Change the Advanced Program-to-Program Communication/MVS (APP/C/MVS) address space information.
- Change the APPC/MVS Transaction Scheduler information.
- Change the PPT information.
- Change the active console group definitions in the sysplex.
- Update the format or contents of the APF list.
- Control dynamic exits and exit routines.
- Update the LNKLST set for the LNKLST concatenation.
- Dynamically add modules to, or remove modules from the LPA.
- Start or stop the common storage tracking and GETMAIN/FREEMAIN/STORAGE trace functions.
- Change the product enablement policy.
- Dynamically change the configuration of z/OS UNIX System Services system characteristics.
- Dynamically change the BPXPRMxx parmlib members in use. See 4-393.
- Dynamically change the run-time library services (RTLS) configuration.
- Change the conversion environment when the system is already up and running.
- Dynamically specify which TSO/E parameter set should be activated.
- Change the new log of logs name, used by DFSMStvs.
- Change the maximum number of unique lock requests that a single unit of recovery can make.

- Change the quiesce exit timeout value to specify the amount of time the DFSMStvs quiesce exits allow to elapse before concluding that a quiesce cannot be completed successfully.
- Change the maximum time that a VSAM RLS or DFSMStvs request is to wait for a required lock before the request is assumed to be in deadlock.
- Specify the exclusion list that the Console Id Tracking facility will use.
- Change the system level Language Environment run-time options.
- Dynamically change keyword values in parmlib member DEVSUPxx.

Note: The system allows a maximum of 38 suffixes.

Scope in a Sysplex

The following table describes the conditions under which the SET command has sysplex scope. See “Using Commands That Have Sysplex Scope” on page 1-11 for an explanation of sysplex scope.

Table 4-29. Sysplex Scope for SET Command

Command	Conditions
SET CNGRP	Has sysplex scope provided all systems are sharing the same members of the logical parmlib.
SET DAE	Has sysplex scope only when all systems are sharing the same DAE data set, and the same members of the logical parmlib.
SET GRSRNL	Has sysplex scope only when all systems are sharing the same members of the logical parmlib.
SET SMS	Has sysplex scope when you are issuing the command to change the name of the ACDS or COMMDS. All systems in the sysplex must be in the same SMS complex, and using the same members of the logical parmlib. If you are issuing the command to start or restart SMS on a system, only the system on which you issue the command is affected. RLS_MAXCFFEATURELEVEL has sysplex scope when you IPL the first system in the sysplex to set the value. All other systems will be told the current value when IPLED. You can change the value by issuing the SET SMS=xx command and specifying RLS_MAXCFFEATURELEVEL in the IGDSMSxx parmlib member, or by using the SETSMS command with the RLS_MAXCFFEATURELEVEL keyword.

SET Command

Syntax

The complete syntax for the SET command is:

```
T [[DATE=yyyyddd][,CLOCK=hh.mm.ss]][RESET]
[,TIMEZONE={W|E}.hh[.mm]]
[,OPT=xx][,SMF=xx][,DAE=xx]
[,MPF={(xx[,xx]...)|NO}]
[,SLIP=xx][,PFK=xx][,IOS=xx][,EXS=xx]
[,SMS=xx][,AKP({nnn[,nnn[...,nnn]}|1000})][,LOG_OF_LOGS(logstream)]
[,MAXLOCKS({max|0},{incr|0})][,QTIMEOUT(nn)]
[,RLSTMOUT({nnn|0})][,RLS_MAXCFFEATURELEVEL({A|Z})]
[,MMS={xx|NO}]
[,PROG=(xx[,xx...]")]
[,DIAG=xx][,GRSRNL=(xx[,xx]...)]
[,APPC=(xx[,xx]...,L)][,ASCH=(xx[,xx]...,L)][,SCH=(xx[,xx]...,L)]
[,CNGRP={(xx,[xx]...)}|NO][,PROD=(xx[,xx]...)]
[,OMVS=(xx[,yy...,nn])]
[,RTLS=(xx[,xx...]")]
[,IKJTSO=xx]
[,CNIDTR=xx]
[,UNI=xx]
[,CEE=(xx[,xx]...,L)]
[,DEVSUP=xx]
```

Note: You may specify the operands in any order, and must specify at least one operand. Do not put a comma before the first operand you specify. If you specify DATE or RESET in a position other than the first, be sure to precede it with a comma. If you specify only one parmlib member with APPC=, ASCH=, CEE=, CNGRP=, GRSRNL=, MPF=, OMVS=, PROG=, RTLS=, or SCH=, you do not need to enter the parentheses.

Parameters

DATE=yyyy.ddd

The local date, where

- *yyyy* is the year, in the range 1900-2042, and
- *ddd* is the day, in the range 001-366.

Notes:

1. The most distant date in the future you may specify is 2042.260.
2. The year must be within seventy (70) years of the UTC date or the system ignores the entire SET command.
3. You must specify the year *yyyy* using four digits.
4. If you specify a new time that implies a change of date, you must explicitly specify the new local date.

CLOCK=hh.mm.ss

The local time in hours (00-23), minutes (00-59), and seconds (00-59).

Notes:

1. The system does not change the date when the new time implies a change of date; if you want a new date, use the DATE parameter or wait for the time to pass midnight.
2. If you specify CLOCK for day 2042.260, the last allowable date, the clock value must not exceed 23.53.47. Later values may cause unpredictable results.

RESET

Specifies the time zone constant that is used to calculate the local date and time is reset to the value that was read in from the CLOCKxx member of the logical parmlib during system initialization. The local date and time are changed accordingly. When you specify RESET, omit DATE and CLOCK.

TIMEZONE={W | E}.hh[.mm]

Specifies the local time zone value.

W | E

Specifies the direction from UTC. W for west of UTC or E for east of UTC.

Default: W

hh.mm

Specifies the number of hours (hh) and minutes (mm) for the local time zone value. The value for hh must be between 00 and 15. The value for mm must be between 00 and 59. The mm value is optional.

Default: 00.00

Do not use the SET TIMEZONE command on any z/OS release before Release 7. If a user attempts to issue this parameter on any z/OS Release before Release 7, the system will issue message IEE309I as follows:

```
IEE309I SET      UNIDENTIFIABLE KEYWORD
```

The following parameters determine which members of the logical parmlib the system is to use. Use them only at the direction of the system programmer. For more information on members of the logical parmlib, see *z/OS MVS Initialization and Tuning Guide*.

OPT=xx

The two alphanumeric characters indicating the IEAOPTxx member of the logical parmlib that contains the new parameters SRM is to use.

SMF=xx

The two alphanumeric characters indicating the SMFPRMxx member of the logical parmlib containing the parameters the system is to use when restarting SMF.

DAE=xx

The two alphanumeric characters indicating the ADYSETxx member of the logical parmlib that contains the new parameters that dump analysis and elimination (DAE) program is to use.

Note: The OPT=, SMF=, and DAE= parameters affect the jobs in progress as well as the jobs read and scheduled after the command.

MPF=(xx[,xx]...)

Specifies one or more MPFLSTxx members of the logical parmlib that are concatenated to form the MPF table. The value of xx can represent any of these items:

- The message(s) being suppressed by MPF
- The action message(s) not being retained by the action message retention facility
- The installation exit(s) to receive control for selected messages
- The status of the general WTO installation exit IEAVMXIT
- Whether this message is automated by MPF

SET Command

- The MPFLSTxx member that identifies the message ID, color attribute, or command installation exit definition
- What color, intensity, and highlighting capabilities are in effect
- The status of the command installation exit routines specified in the logical parmlib member MPFLSTxx
- The current installation options for handling foreign messages

The MPF parameter in the INIT statement in the CONSOLxx member of the logical parmlib controls which, if any, MPFLSTxx members are active at IPL.

MPF=NO

Ends MPF processing (message suppression and presentation). NO is ignored when specified in combination with a 2-character suffix.

SLIP=xx

The two alphanumeric characters indicating the IEASLPxx member of the logical parmlib that contains the commands SLIP processing is to use.

PFK=xx

The two alphanumeric characters indicating the PFKTABxx member of the logical parmlib that contains the PFK tables that are to be available for a console. The PFK(xx) keyword on the INIT statement in CONSOLxx identifies the PFKTABxx member that is available at IPL.

Note: The CONTROL command (K N,PFK=nnnnnnnn) must be issued to invoke the PFKTABxx member specified in the SET command.

SMS=xx

The two alphanumeric characters indicating the IGDSMSxx member of the logical parmlib that contains the parameters the system is to use when it starts SMS. Specifying SMS=xx also starts SMS if it was not started at IPL or, restarts SMS if it has stopped and can't restart itself. Depending on the setting of the PROMPT keyword in the IEFSSNxx parmlib member, this command can display the parameters in the IGDSMSxx member. (For a comparison of the SET SMS command with the SETSMS command, see Table 4-34 on page 4-458.)

You can use SET SMS=xx to specify an IGDSMSxx PARMLIB member that contains PDSESHARING(EXTENDED) to migrate members of a sysplex to PDSE extended sharing. This SET SMS command must be routed to every system that was operating with a PDSESHARING(NORMAL) PARMLIB member. This SET SMS command establishes that system's preference and causes it to communicate with the other sysplex members that it would like to switch to extended sharing. When all members have requested extended sharing, the sysplex can migrate to that level of sharing. You might have to issue SET SMS=xx a second time to trigger the switch from normal to extended sharing. Each of the systems issues message IGW306I when it migrates to extended sharing.

When the IGDSMSxx member of SYS1.PARMLIB is read, it can cause changes to any of the parameters that DFSMStvs is using. The SET SMS command affects the following DFSMStvs parameters in the IGDSMSxx member:

- AKP
- LOG_OF_LOGS
- MAXLOCKS
- QTIMEOUT
- RLSTMOUT
- RLS_MAXCFFEATURELEVEL
- RLSABOVETHEBARMAXPOOLSIZE
- RLSFIXEDPOOLSIZE

Refer to the optional keywords for the IGDSMSxx parmlib member in *z/OS MVS Initialization and Tuning Reference*.

AKP({nnn[,nnn[...],nnn]}|1000}

Specifies one or more activity keypoint trigger (AKP) values. Each AKP value (*nnn*) is the number of logging operations between the taking of keypoints. You can specify up to 32 activity keypoint values. AKP values must be specified in the same order as DFSMStvs instance names. Valid values are from 200 to 65535. The default is 1000.

LOG_OF_LOGS(logstream)

Specifies the name of the log-of-logs log stream. If the log of logs is changed, the new name will be saved, but it will not take effect until the next time DFSMStvs is restarted. Changing the name without quiescing the old log of logs before the next DFSMStvs restart can cause a mismatch of the tie-up records written at data set open with file-close records.

MAXLOCKS({maxl0},{incrI0})

Specifies a pair of values in the range of 0 to 999999. The two values are the maximum number of unique lock requests that a single unit of recovery can make and an increment value. Once the maximum number of unique lock requests is reached, warning messages are issued every time the number of unique lock requests over and above the maximum increases by a multiple of the increment. When the maximum number is reached, warning message IGW859I is issued to the system console, and message IGW10074I is issued to the job log. The messages include the name of the job that is holding the locks. This information will help you to determine whether the job should be canceled, in which case the unit of recovery will be backed out, and the locks will remain held until the backout completes. Specifying a value of 0 indicates that warning messages IGW859I and IGW10074I should not be issued.

The TVSNAME(*nnn*) parameter must also be specified with the MAXLOCKS parameter. This parameter applies across all systems.

Notes:

1. Lock requests are considered unique if they lock different records within the base cluster. Repeated requests for the same base cluster records will not result in the count being incremented.
2. Warning messages IGW859I and IGW10074I are not issued for units of recovery that are in backout. This is because a unit of recovery that is in backout cannot obtain locks on any additional records.
3. Messages IGW859I and IGW10074I are issued until the unit of recovery reaches commit. Once the unit of recovery reaches commit, no additional messages will be issued.
4. To avoid flooding the system console with messages, messages IGW859I and IGW10074I are issued by an asynchronous timer driven task that wakes up every 10 seconds. This means that the messages will not necessarily reflect the exact values specified for the maximum and the increment, but rather will reflect the values which represent the state of the unit of recovery at the time the task awakens.
5. MAXLOCKS takes into account the number of unique lock requests. It does not count the actual number of locks obtained. The number of locks requested differs from the number of locks held when alternate indexes are used. If an update modifies alternate keys, a lock is obtained for the base record, for each old alternate key, and for each

SET Command

new alternate key. Therefore, if n alternate keys are modified, a single lock request can result in obtaining $(2n+1)$ locks.

Some examples of how this parameter can be specified are:

MAXLOCKS(0,0)

Valid - messages IGW859I and IGW10074I will never be issued.

MAXLOCKS(,)

Valid - this is the equivalent of specifying MAXLOCKS(0,0); messages IGW859I and IGW10074I will never be issued.

MAXLOCKS(5000,0)

Valid - messages IGW859I and IGW10074I will be issued when the asynchronous task wakes up and a unit of recovery has made its 5000th lock request.

MAXLOCKS(0,2000)

Not valid

MAXLOCKS(4500,1000)

Valid - messages IGW859I and IGW10074I will be issued when the asynchronous task wakes up and a unit of recovery has made its 4500th lock request and again every 1000 unique lock requests thereafter.

MAXLOCKS(1000,2300)

Valid - messages IGW859I and IGW10074I will be issued when the asynchronous task wakes up and a unit of recovery has made its 1000th lock request and again every 2300 unique lock requests thereafter.

MAXLOCKS(3200,)

Valid - this is the equivalent of specifying MAXLOCKS(3200,0); messages IGW859I and IGW10074I will be issued when the asynchronous task wakes up and a unit of recovery has made its 3200th lock request.

MAXLOCKS(,2000)

Not valid

The default for both values is 0.

QTIMEOUT(*nnn*)

Specifies the quiesce exit timeout value in seconds. The quiesce timeout value specifies the amount of time the DFSMStvs quiesce exits allow to elapse before concluding that a quiesce cannot be completed successfully. Specify a value between 60 to 3600. Changing the value of QTIMEOUT affects only those quiesce requests that are submitted after the change is made; it has no effect on quiesce requests that are already in progress. The default is 300.

The command format is:

SETSMS QTIMEOUT(300)

RLSTMOUT({*nnnl0*})

Specifies the maximum time, in seconds, that a VSAM RLS or DFSMStvs request is to wait for a required lock before the request is assumed to be in deadlock and ended with VSAM return code 8 and reason code 22 (X'16'). Specify a value in seconds between 0 to 9999. A value of 0 means that the

VSAM RLS or DFSMStvs request has no time out value; the request will wait for as long as necessary to obtain the required lock.

VSAM RLS detects deadlocks within VSAM and DFSMStvs. It cannot detect deadlocks across other resource managers, and uses the timeout value to determine when such deadlocks might have occurred. You can specify a global timeout value in the IGDSMSxx member of SYS1.PARMLIB, a step level timeout value on the JCL, or a timeout value on the RPL passed for each VSAM request.

For a particular VSAM RLS or DFSMStvs request, the value used for timeout is:

1. The value specified in the RPL, if any.
2. The value specified in the JCL at the step level, if any.
3. The value specified in the IGDSMSxx member of SYS1.PARMLIB, if any.

RLSTMOUT is valid a parameter for either VSAM RLS or DFSMStvs. If you specify RLSTMOUT but do not specify the TVSNAME parameter, the value is used only by RLS. For DFSMStvs, the first instance of DFSMStvs brought up within the sysplex determines the value. Subsequent DFSMStvs instances use the value established by the first system, regardless of what might be specified in their members of SYS1.PARMLIB.

RLSTMOUT can be specified only once in a sysplex and applies across all systems in the sysplex.

The default is 0.

RLS_MAXCFFEATURELEVEL({A|Z})

Specifies the method that VSAM RLS uses to determine the size of the data that is placed in the CF cache structure. If you specify A, caching proceeds using the RLSCFCACHE keyword characteristics that are specified in the SMS data class that is defined for the VSAM sphere. If you do not specify a value, or if you specify Z, then only VSAM RLS data that have a Control Interval (CI) value of 4K or less are placed in the CF cache structure. The default is Z.

Restrictions:

- If A is specified for the RLS_MAXCFFEATURELEVEL parameter, systems lower than V1R3 will not be able to connect to the CF cache structure.
- If a lower-level system is the first system activated in the sysplex, RLS_MAXCFFEATURELEVEL defaults to Z, and all systems will be able to connect to the CF cache structure.
- If the SETSMS command is used to change the RLS_MAXCFFEATURELEVEL value to A on a mixed-level system, the command is rejected and message IGW500I is issued.

MMS=xx

When the MVS message service (MMS) is not active, SET MMS=xx starts the message translation service. When MMS is active, SET MMS=xx changes the MMSLSTxx member. The two alphanumeric characters indicate the MMSLSTxx member of the logical parmlib the system is to use.

MMS=NO

Ends MMS processing and frees all allocated resources.

SET Command

GRSRNL=(xx[,xx]...)

Specifies one or more GRSRNLxx members of the logical parmlib that contain the specified GRS resource name lists. Each value of xx is two alphanumeric characters that indicate a GRSRNLxx member. GRSRNL enables you to change the current RNLs specified in one or more GRSRNLxx member of SYS1.PARMLIB. Do not use parentheses when only one parmlib member is specified.

Attention: Use extreme caution when issuing the SET GRSRNL command to change heavily used or highly critical resources. Work that requires resources for a critical application, or resources used by the operating system, might become suspended or delayed, which can impair the performance of that critical application or the operating system itself.

For more information about the use of the SET GRSRNL command, see *z/OS MVS Planning: Global Resource Serialization*.

Restriction: You cannot specify the GRSRNL= parameter if GRSRNL has already been set to EXCLUDE in the logical parmlib member IEASYSxx.

IOS=xx

The two alphanumeric characters indicating the IECIOSxx member of the logical parmlib that contains the parameters the system is to use to control MIH processing, I/O timing processing, and other IOS functions.

You can change the MIH timing intervals, or the I/O timing intervals, or the MIH timing intervals and the I/O timing intervals ONLY for devices that have MIH statements coded in IECIOSxx.

Notes:

1. IECIOSxx can also contain parameters that control hot I/O processing. Using SET IOS=xx to change to another member does not affect the hot I/O parameters; hot I/O processing is unchanged. You can only change hot I/O processing parameters at system initialization time in response to message IEA101A.
2. During IPL (if the device is defined to be ONLINE), or during the VARY ONLINE process, some devices present their own MIH timeout values through the primary/secondary MIH timing enhancement contained in the self-describing data for the device. The primary MIH timeout value is used for most I/O commands. However, the secondary MIH timeout value can be used for special operations such as long-busy conditions or long-running I/O operations. Any time a user specifically sets a device or device class to have an MIH timeout value that is different from the IBM-supplied default for the device class, the user-specified value overrides the device-established primary MIH time value. This implies that if an MIH time value that is equal to the MIH default for the device class is explicitly requested, IOS does NOT override the device-established primary MIH time value. To override the device-established primary MIH time value, you must explicitly set aside a time value that is not equal to the MIH default for the device class.

Overriding the device-supplied primary MIH timeout value can adversely affect MIH recovery processing for the device or device class.

See the specific device's reference manuals to determine if the device supports self-describing MIH time values.

DEVSUP=xx

The two alphanumeric characters indicating the DEVSUPxx member of the logical parmlib that contains the parameters the system is to use to set device related controls.

| **Note:** The category codes for tape library partitioning can not be changed.

EXS=xx

The two alphanumeric characters indicating the EXSPATxx member of the logical parmlib that contains the excessive spin recovery actions and the excessive spin loop timeout interval.

APPC=(xx[,xx]...,L)

The two alphanumeric characters indicating the APPCPMxx parmlib member that contains the desired APPC/MVS address space configuration that is appended to the existing configuration. The APPCPMxx member can reside in a logical parmlib or if no logical parmlib is specified, in any parmlib data set specified on an //IEFPARM DD statement in the master scheduler JCL. The L is optional and causes the system to display parmlib statements on the operator console as they are processed.

ASCH=(xx[,xx]...,L)

The two alphanumeric characters indicating the ASCHPMxx parmlib member that contains the desired APPC/MVS scheduler configuration that is appended to the existing configuration. The ASCHPMxx member can reside in a logical parmlib or if no logical parmlib is specified, in any parmlib data set specified on an //IEFPARM DD statement in the master scheduler JCL. The L is optional and causes the system to display parmlib statements on the operator console as they are processed.

SCH=(xx[,xx]...,L)

The two alphanumeric characters indicating the SCHEDEXX members of the logical parmlib that contain the desired program properties table (PPT) configuration. The L is optional and causes the system to display parmlib statements on the operator console as they are processed.

The SET SCH command causes the system to replace the current PPT definitions with the IBM-supplied default PPT definitions and the PPT definitions from one or more SCHEDEXX members that you specify on the command. The effect of the command is not cumulative. The new PPT definitions take effect immediately, without requiring a re-IPL of the system.

Notes:

1. The SET SCH command only affects the PPT configuration statement.
2. If the SET SCH command fails, the current PPT configuration remains active.

CNGRP=(xx,[xx]...)

The two alphanumeric characters indicating the CNGRPXX member of the logical parmlib to be activated. This data is broadcast across the sysplex. It references the logical parmlib only on the system where the command executes, and activates only the CNGRPXX members found in that logical parmlib.

CNGRP=NO

The system is to remove all active console group definitions from the sysplex.

PROG=(xx[,xx...])

The two alphanumeric characters that specify one or more PROGXX parmlib members. The system processes the members in the order specified. If it encounters a member that does not exist, command processing stops. Each PROGXX member contains definitions that:

- Control the format and contents of the list of APF-authorized libraries
- Control the use of exits and exit routines
- Control the LNKLST concatenation by defining and modifying LNKLST sets

SET Command

- Control the addition of modules to, and removal of modules from, the LPA after IPL

You can use the SET PROG=xx command to control exits previously defined to the dynamic exits facility. Dynamic exits services are implemented by:

- The EXIT statement of the PROGxx parmlib member. The EXIT statement of PROGxx allows an installation to add exit routines to an exit, delete an exit routine for an exit, change the state of an exit routine, change the attributes of an exit, and undefine an implicitly defined exit.

The PROGxx EXIT statement interacts with the PROG=xx parameter of IEASYSxx and the SET PROG=xx command. At IPL, you can use PROG=xx to specify the particular PROGxx parmlib member the system is to use.

During normal processing, you can use the SET PROG=xx command to set a current PROGxx parmlib member. See *z/OS MVS Initialization and Tuning Reference* for information about the PROGxx parmlib member.

- The SETPROG EXIT operator command. This command performs the same functions as the EXIT statement of the PROGxx parmlib member.
- The CSVDYNEX macro. The CSVDYNEX macro can be used to define exits to the dynamic exits facility, control their use within a program, and associate one or more exit routines with those exits. It can also be used to associate exit routines with the existing SMF and allocation exits, which have been defined to the dynamic exits facility.

You can use the SET PROG=xx command to control the LNKLST concatenation. The PROGxx LNKLST statement interacts with the PROG=xx parameter of IEASYSxx and the SET PROG=xx command. At IPL, you can use PROG=xx to specify the particular PROGxx parmlib member the system is to use. During normal processing, you can use the SET PROG=xx command to set a current PROGxx parmlib member, or use the SETPROG LNKLST operator command to modify LNKLST sets. This command performs the same functions as the LNKLST statement of the PROGxx parmlib member and allows you to make dynamic changes to a LNKLST set. See "SETPROG Command" on page 4-441.

You can use the SET PROG=xx command to control the content of the LPA dynamically following IPL. The PROGxx LPA statement can specify modules that are to add to the LPA following IPL, those to delete from the LPA, and threshold values for minimum amounts of CSA storage that must still be available after an ADD operation.

You can also initiate a change to LPA from a program via the CSVDYLPA macro, or by an operator using the SETPROG command. See *z/OS MVS System Commands*. However, modules accessed through a Program Call (PC) instruction cannot be replaced using a SETPROG LPA command. That is because even though the addresses of those modules are stored in the PC table, that table is not updated by the SETPROG LPA command.

DIAG=xx

The two alphanumeric characters indicating the DIAGxx member of the logical parmlib containing definitions that control:

- Common service area (CSA), extended CSA (ECSA), system queue area (SQA), and extended SQA (ESQA) tracking
- GETMAIN/FREEMAIN/STORAGE (GFS) trace

PROD=(xx[.xx]...)

The two alphanumeric characters indicating the IFAPRDxx members of the

logical parmlib that contain the desired product enablement policy. If a policy already exists, the system performs the actions defined in the specified member(s) to modify the existing policy.

The system processes the members in the order specified. If it encounters a member that does not exist, command processing stops.

OMVS=(xx[,xx...,nn])

The two alphanumeric characters that specify one or more BPXPRMxx parmlib members. If you specify only one member, putting parenthesis around the member is optional. If you specify more than one parmlib member, you must put parenthesis around the members.

RTLS=(xx[,yy...])

The two alphanumeric characters indicating the CSVRTLxx members of the logical parmlib that contain the desired run-time library services specification. The system processes the members in the order specified. If it encounters a member that does not exist, command processing stops.

UNI=(xx[,xx...])

The two alphanumeric characters indicating the CUNUNIx parmlib member that controls the conversion environment (Unicode services). Use this command to change the environment when the system is already up and running.

IKJTSO=xx

The two alphanumeric characters indicating the IKJTSOxx parmlib member to be activated. This command will perform processing similar to the TSO/E PARMLIB UPDATE command. A switch will be attempted when a broadcast data set that is different from the one currently being used is specified in the IKJTSOxx parmlib member. If the IKJTSOxx parmlib member specifies a new broadcast data set, operator confirmation of the switch will be required unless NOPROMPT is specified in the parmlib member.

CNIDTR=xx

The two alphanumeric characters indicate the CNIDTRxx parmlib member to be used. This command specifies the exclusion list that the Console Id Tracking facility will use. The facility will use this list for all new violations. Previously recorded violations will continue to be displayed by the DISPLAY OPDATA,TRACKING command, even if they now match an exclusion statement.

Note: Once a parmlib member has been activated, the only way to not have an exclusion list active is to activate a member that contains no exclusion definitions.

CEE=(xx[,xx]...,L)

The two alphanumeric characters that specify one or more CEEPRMxx parmlib members. If you specify only one member, putting parenthesis around the member is optional. If you specify more than one parmlib member, you must put parenthesis around the members. The L is optional and causes the system to display parmlib statements on the operator console as they are processed.

Example 1

When the displayed local time and date are 19.00.00 and 191.141, respectively, to set the local time ahead to 1:00 a.m., enter:

SET Command

T DATE=1991.142,CLOCK=01.00.00

OR

T DATE=91.142,CLOCK=01.00.00

It is necessary to enter DATE because the time change, in this example to 1:00 a.m., implies a change of date.

Example 2

To reset the time and date to the values set during IPL, enter:

T RESET

Example 3

To restart SMF with the parameters found in the SMFPRMAA member of the logical parmlib, enter:

T SMF=aa

Example 4

To change SMS parameters to those found in the IGDSMS21 member of the logical parmlib, or to start or restart SMS by using the parameters in that member, enter:

SET SMS=21

Example 5

To change MMS parameters to the parameters found in the MMSLST3A member of the logical parmlib or to start MMS using the parameters in that member, enter:

SET MMS=3A

Example 6

To change the current GRSRNLs to those found in the GRSRNL01, GRSRNL09, and GRSRNL12 members of the logical parmlib, enter:

SET GRSRNL=(01,09,12)

Example 7

To change the desired APPC/MVS address space configuration with the parameters found in the APPCPM01 member of the logical parmlib and the desired APPC/MVS scheduler configuration in ASCHPM12, ASCHPM03, and ASCHPM09, enter:

SET APPC=01,ASCH=(12,03,09)

Example 8

To change the desired PPT configuration with the parameters found in the SCHED04 and SCHED05 members of the logical parmlib and also list the parmlib statements to the operator console as they are processed, enter:

SET SCH=(04,05,L)

Example 9

To SET the console group definitions in the CNGRPAA member, enter:

```
SET CNGRP=AA
```

Example 10

To SET the console group definitions in the members CNGRPAA and CNGRPBB, enter:

```
SET CNGRP=(AA,BB)
```

Example 11

To remove all console group definitions from the sysplex, enter:

```
SET CNGRP=NO
```

Example 12

To change the MPFLSTxx member that builds the MPF table the system uses, enter:

```
SET MPF=06
```

Example 13

To change the MPFLSTxx members that builds the MPF table the system uses, enter:

```
SET MPF=(A1,A2,B4)
```

Example 14

To set the PROGxx member that the system uses to reference the APF list, enter:

```
SET PROG=03
```

Example 15

To set the current DIAGxx member to DIAG05, enter:

```
SET DIAG=05
```

Example 16

To change the product enablement policy with the parameters found in IFAPRDA2 and IFAPRDA3, enter:

```
SET PROD=(A2,A3)
```

Example 17

To change the RTLS configuration with the parameters found in parmlib members CSVRTLA2 and CSVRTLA3, enter:

```
SET RTLS=(A2,A3)
```

Example 18

To change to using the TSO/E parameters found in IKJTSOA1, enter:

```
SET IKJTSO=A1
```

SETAPPC Command

Use the SETAPPC command to dynamically define or modify the APPC/MVS configuration. Using this command, you can dynamically add or modify definitions for the APPC configuration without the need to edit an APPC parmlib member and issue a separate SET APPC command. However, if changes made to the APPC configuration using SETAPPC are permanent in nature, incorporate the configuration modifications into the appropriate APPC parmlib member. In this way, whenever APPC is recycled or if the system is re-IPLED, the configuration reflects the changes made by the SETAPPC command.

Syntax

The syntax of the SETAPPC command is:

```
SETAPPC {LUADD,ACBNAME=uname
          [,SCHED=schedname|NOSCHED]
          [,BASE ]
          [,PSTIMER=value|NONE|INDEFINITE]
          [,TPDATA=(dsname)]
          [,TPLEVEL={SYSTEM|GROUP|USER}]
          [,ALTLU=scheduler-supplied-value]
          [,USERVAR=scheduler-supplied-value]
          [,GRNAME=genericname]
          [{NQN | NONQN}]}}

{LUDEL,ACBNAME=uname
   ,[{PERSIST | NOPERSIST}]} }
```

{SIDEINFO,DATASET=(dsname)}

Parameters**LUADD**

Defines a local LU for the APPC/MVS configuration.

Use the SETAPPC LUADD command to define a local APPC/MVS LU to the APPC configuration.

The LUADD command must specify an LU name and (optionally)

- An indication of whether the LU is associated with a transaction scheduler
- The name of the transaction scheduler, if one is to be associated with this LU
- The amount of time the LU's sessions will persist in the event the LU becomes unavailable
- The TP profile file associated with the LU
- The level of TP profile from which the LU starts to search
- Optional values to be passed to an alternative transaction scheduler, or to any other member of the APPC XCF group, such as an APPC/MVS server
- A VTAM generic resource name to associate with the LU
- An indication of whether the LU is enabled to support network-qualified names for its partner LUs.

Each LU managed by APPC/MVS must be defined by either an LUADD statement previously invoked through the SET APPC command or through the SETAPPC command. When an installation uses the ASCH transaction scheduler exclusively, only one LU is required. If other transaction schedulers

are used, each scheduler requires a separate LU. An installation might choose to define additional LUs to isolate TPs for security or testing.

An installation can also define LUs that are not associated with transaction schedulers. These LUs handle work that is processed by APPC/MVS servers, rather than scheduled by a transaction scheduler. Such LUs are indicated by using the NOSCHED keyword on LUADD. Installations can also use NOSCHED LUs when they want to flow outbound allocate requests without having a transaction scheduler active. (Note that APPC/MVS servers can also run under LUs that are associated with transaction schedulers.)

You can modify an LU by overriding previously defined LUs made through either the SET APPC or SETAPPC commands. In this case, the SETAPPC LUADD command specifies an ACBNAME that names an existing LU and then the parameters to be modified. The only parameters you cannot modify with an overriding LUADD are the SCHED, NOSCHED, ALTLU, USERVAR, GRNAME, NQN and NONQN parameters. To change these parameters, first delete the LU with a SETAPPC LUDEL command and then issue a SETAPPC LUADD command to re-add the LU with a changes to the parameters.

Example:

The following example defines LU MVSLU01 to be associated with the transaction scheduler provided with APPC/MVS:

```
SETAPPC LUADD,ACBNAME=MVSLU01,SCHED=ASCH,TPDATA=(SYS1.APPCTP),TPLEVEL=USER
```

ACBNAME(luname)

The required name of the LU that APPC/MVS is to remove. If this LU was defined to VTAM®, its association with VTAM is terminated after active conversations end.

Value Range: A one- to eight-byte character string of uppercase letters A through Z, numerals 0-9, national characters (@,\$,#) that must begin with an alphabetic or national character.

The SNA LU 6.2 architecture defines a network-qualified LU name to be up to 17 bytes in length and in the form *network_id.network_LU_name*, where *network_id* is the optional 8-byte id of the network and *network_LU_name* is the 8-byte local LU name. SAA® CPI Communications allows the full 17-byte network-qualified LU name. However, for the ACBNAME keyword, specify only the 8-byte local LU name.

Default: None, this parameter is required.

SCHED(ASCH|schedname)

NOSCHED

An optional parameter that indicates whether the LU is to be associated with a transaction scheduler. LUs associated with a transaction scheduler cannot become active until that scheduler identifies itself to APPC/MVS. LUs not associated with a transaction scheduler become active as soon as APPC/MVS becomes active.

SCHED indicates that the LU is associated with a transaction scheduler. *schedname* must match the name the transaction scheduler specifies when it calls the Identify service. For more information about the Identify service and its scheduler_name parameter, see *z/OS MVS System Messages, Vol 3 (ASB-BPX)*.

SETAPPC Command

NOSCHED indicates that the LU is not to be associated with a scheduler. When NOSCHED is specified, the LU becomes active as soon as APPC/MVS becomes active. Installations can use NOSCHED LUs to isolate work from schedulers when the work is to be processed by APPC/MVS servers. Installations can also use NOSCHED LUs to flow outbound allocate requests without having a transaction scheduler active.

Value Range: For *schedname*, the value is a one- to eight-byte character string and each character must be an uppercase letter (A-Z) or a numeral (0-9).

Note: SCHED and NOSCHED are mutually exclusive keywords; you cannot specify both SCHED and NOSCHED in a single LUADD statement. Doing so causes the system to ignore the statement and issue message ATB041I to the system operator.

Default: When you omit both SCHED(*schedname*) and NOSCHED, the default is SCHED(ASCH).

BASE

An optional parameter that designates the LU as the base LU. Base LUs are default LUs assigned to handle outbound work. A base LU can be the default LU associated with a particular transaction scheduler or a NOSCHED LU.

When a NOSCHED LU is defined with the BASE option, the LU becomes the *system base LU*. That means the LU is to be the default LU used for outbound allocate requests from MVS programs, such as batch jobs, TSO/E users, started tasks, and other work requests that attempt to enter the network without being associated with a scheduler or an LU.

Example: The following example defines a NOSCHED LU, MVSLU02, to be the system base LU.

```
SETAPPC LUADD,ACBNAME=MVSLU02,NOSCHED,BASE,TPDATA=(SYS1.APPCTEST),TPLEVEL=SYSTEM
```

If you do not define a NOSCHED LU as a base LU, the base LU defined for the APPC/MVS transaction scheduler (ASCH) becomes the system base LU. If the system base LU does not exist, APPC/MVS rejects conversations allocated by MVS programs that are not associated with a scheduler or an LU.

IBM recommends that you define one LU per transaction scheduler as the base LU for the scheduler. In addition, define a NOSCHED LU as the system base LU if you want to allow outbound requests from the system when no transaction schedulers are active.

When more than one LU is defined as the base LU, the one most recently defined is the base.

PSTIMER(*value*)

An optional parameter that sets the maximum amount of time for which the LU's sessions **persist** (are maintained) during interruptions in APPC/MVS or a transaction scheduler's service.

When you specify a valid value other than NONE, the LU's sessions persist when the APPC address space is canceled, forced, terminated, or automatically restarted. The sessions also persist during interruptions in scheduler service.

Any conversations that were active at the time of the interruption are lost. When APPC service is resumed, the conversation partners can re-establish these conversations, if desired.

Sessions do not persist in the event the LU is deleted.

Value Range:

- 0 or INDEFINITE (Sessions persist indefinitely)
- 1 - 86400 (Number of seconds the sessions can persist)
- NONE (Sessions are not to persist)

Default: NONE

TPDATA(dsname)

An optional parameter that specifies the name of the VSAM key-sequenced data set that contains TP profiles, along with an optional data base token for the LU. The data base token is used for verifying access authority to TP profiles. If this LU is a NOSCHED LU, APPC/MVS uses only the data set's data base token, if any. The data set specified on TPDATA must be catalogued in either a user catalog or the master catalog.

Value Range: Up to 44 characters in length consisting of one- to eight-byte character string of uppercase letters A through Z, numerals 0-9, national characters (@,\$,#) that must begin with an alphabetic or national character.

Default: SYS1.APPCTP

TPLEVEL({SYSTEM|GROUP|USER})

An optional parameter that identifies the level of TP profiles for which the LU searches in response to an inbound allocate request. TPLEVEL limits the search to the levels desired.

Each TP can have different levels of TP profiles with scheduling characteristics associated with a user, a group of users, or all users (system). The TPLEVEL parameter tells the LU which of those levels of TP profile to search.

Value Range:

SYSTEM means that the LU searches for system-level TP profiles only (NOT for a specific user or group of users).

GROUP means that the LU searches for TP profiles associated with (1) a specific group of users and (2) system-level TP profiles, in that order.

USER means that the LU searches for TP profiles associated with (1) a specific user, (2) a group of users, and (3) system-level TP profiles, in that order.

Note: If you specify NOSCHED, TPLEVEL must be SYSTEM. Also, TP profile entries in the data set specified in TPDATA are not used for NOSCHED LUs; only the data base token is used.

Default: SYSTEM

ALTLU(scheduler-supplied value)

This parameter allows optional, installation-supplied data to be passed to a member of the APPC XCF group, such as an alternative transaction scheduler or an APPC/MVS server.

If specified, the data is passed to the APPC XCF group member at the activation and deactivation of the associated LU. For information about the APPC XCF group, see *z/OS MVS System Messages, Vol 3 (ASB-BPX)*.

SETAPPC Command

Value Range: A one- to eight-byte character string of uppercase letters A through Z, numerics 0-9, or national characters (@, \$, #), with the exception that the first character cannot be numeric (0-9).

Default: None

USERVAR(scheduler-supplied value)

This parameter allows optional, installation-supplied data to be passed to a member of the APPC XCF group, such as an alternative transaction scheduler or an APPC/MVS server.

If specified, the data is passed to the APPC XCF group member at the activation and deactivation of the associated LU. For information about the APPC XCF group, see *z/OS MVS System Messages, Vol 3 (ASB-BPX)*.

Value Range: A one- to eight-byte character string of uppercase letters A through Z, numerics 0-9, or national characters (@, \$, #), with the exception that the first character cannot be numeric (0-9).

Default: None

GRNAME(genericname)

This optional parameter specifies a VTAM generic resource name to be associated with the LU. The LU may be one of multiple LUs in the same generic resource group, represented by *genericname*. This parameter cannot be dynamically modified or added to an existing LU definition.

See *z/OS MVS Planning: APPC/MVS Management* for advice and restrictions about selecting a generic resource name, and deciding which LUs should become members of a generic resource group.

Value Range: A one- to eight-byte character string of uppercase letters A through Z, numerals 0-9, national characters (@, \$, #) and must begin with an alphabetic or national character.

Default: None. If the GRNAME parameter is not specified, the LU is activated but is not part of a generic resource group.

NQN

NONQN

An optional parameter that specifies whether the APPC/MVS LU is enabled to use a network-qualified partner LU name when first allocating outbound conversations. If you specify NQN, APPC/MVS uses the 17-byte network-qualified LU name when both verifying the partner LU, and sending the outbound Allocate request to the partner LU. If you specify NONQN (or allow the system to use the default), APPC/MVS uses the entire name when verifying the partner, but only the 8-byte network-LU-name portion when sending the outbound Allocate request, as in OS/390® V1R2 and previous releases.

See *z/OS MVS Planning: APPC/MVS Management* for the requirements for enabling APPC/MVS LUs to support network-qualified names.

Default: NONQN

LUDEL

The **LUDEL** command deletes a local APPC/MVS LU from the APPC configuration. One LUDEL statement must be specified for each LU to be deleted. The LUDEL statement contains:

- The LU name
- An indication of whether APPC/MVS keeps all persistent sessions active between this LU and all of its partners

When an LUDEL statement is processed, incoming allocation requests to the named LU are rejected; however, all active conversations are allowed to continue until completed. The LU is removed only after all active conversations have ended.

ACBNAME(luname)

The required name of the LU that APPC/MVS is to remove. If this LU was defined to VTAM, its association with VTAM is terminated after active conversations end.

Value Range: A one- to eight-byte character string of uppercase letters A through Z, numerals 0-9, national characters (@,\$,#) and must begin with an alphabetic or national character.

For an explanation of why SAA CPI partner LU names can be 17 characters, see the note under the ACBNAME parameter in 4-397.

Default: None; this parameter is required.

PERSIST | NOPERSIST

An optional parameter that specifies whether APPC/MVS will deactivate all sessions between this LU and its partners when the LU is deleted. If you specify PERSIST, and if the LU was previously enabled to support persistent sessions through the PSTIMER keyword on the LUADD statement, APPC/MVS does not deactivate sessions between the LU and its partners. VTAM keeps these sessions active as long as the LU is re-added to the APPC configuration on the same OS/390 image within the PSTIMER time limit (single-node persistent sessions) or in any OS/390 image in the sysplex within the PSTIMER time limit (multi-node persistent sessions). See *z/OS MVS Planning: APPC/MVS Management* for further information. If you specify NOPERSIST (or allow the system to use the default), APPC/MVS deactivates all sessions between this LU and its partners when the LU is deleted.

Default: NOPERSIST

SIDEINFO

The SIDEINFO statement names the VSAM key sequenced data set that contains side information. Only one side information file is allowed per MVS system.

DATASET(dsname)

An optional parameter that specifies the name of the VSAM key sequenced data set that contains side information. The file must be cataloged in either a user catalog or the master catalog.

Value Range: Up to 44 characters in length consisting of one- to eight-byte character string of uppercase letters A through Z, numerals 0-9, national characters (@,\$,#) and must begin with an alphabetic or national character.

Default: SYS1.APPCSI

SETCEE Command

Use the SETCEE command to change Language Environment run-time options after the Parmlib member has been read. You can modify multiple options in one SETCEE command; however, there is a limit of 126 characters per command. You can not continue the SETCEE command on a second line, each option must be completed in the 126 character limit. To synchronize the setting of multiple options, use the SET CEE command to use additional Parmlib members.

Syntax

The complete syntax for the SETCEE command is:

```
SETCEE [CEEDOPT,opt,opt,...]
        [CEECOPT,opt,opt,...]
        [CELQDOPT,opt,opt,...]
```

Parameters

CEEDOPT

Sets your specified Language Environment run-time options in a non-CICS environment.

CEECOPT

Sets your specified Language Environment run-time options in a CICS environment.

CELQDOPT

Sets your specified Language Environment run-time options in an AMODE 64 environment.

opt

Specifies the Language Environment run-time option you wish to change. The option can be any option that is valid in the CEEPRM member. For a list of valid options, see the example of the CEEPRM member in *z/OS MVS Initialization and Tuning Reference*.

Example 1

```
SETCEE CEEDOPT,POSIX(ON)
```

Example 2

```
SETCEE CELQDOPT,HEAP64(1M),IOHEAP64(1M,1M)
```

SETCON Command

Use the SETCON command to activate functions pertaining to the console environment and the Console ID Tracking facility. Also use the SETCON MONITOR command to control the monitoring of messages in your installation. The MONITOR option allows you to receive monitored messages without requiring that the messages be queued to a console or be written to SYSLOG or OPERLOG. For more information about the Console ID Tracking facility and the MONITOR option, see *z/OS MVS Planning: Operations*.

Syntax

The syntax of the SETCON command is:

```
SETCON {TRACKING|TR}={ON|OFF|ONWITHABEND}
        {MONITOR|MN}
        {,JOBNAMES={(ON[,LOG|NOLOG])|OFF}}
        {,SESS={(ON[,LOG|NOLOG])|OFF}}
        {,STATUS={(ON[,LOG|NOLOG])|OFF}}
        {,T={ON|OFF}}
```

Parameters

TRACKING or TR

The system is to make changes to the Console ID tracking facility. The tracking facility records instances of one-byte or migration console ID usage. These instances are known as violations. Programs that use these one-byte or migration IDs are known as violators.

ON

Activates the Console ID tracking facility to accept the recording of instances. No change is made if the facility is already active. If the facility is in ABEND mode, it will be taken out of ABEND mode without losing any recorded instances.

OFF

Deactivates the Console ID tracking facility to reject all attempts to record instances of a one-byte or migration console usage. No change is made if the facility is already inactive. Before deactivation, a DISPLAY OPDATA,TRACKING command will be issued by the tracking facility to record the current violations. The tracking facility will attempt to ensure that the DISPLAY OPDATA,TRACKING command completes before terminating the facility. However, if the facility terminates before the DISPLAY OPDATA,TRACKING command can run, the recorded instances will be lost and will have to be recreated.

Because turning off the facility takes a few seconds, wait until the IEE7121 SETCON PROCESSING COMPLETE message is issued before reactivating the facility.

ONWITHABEND

Activates the Console ID tracking facility to accept the recording of instances. Violators of console ID usage will be ABENDED with a 077 ABEND code (reason code '34'X). The violator is not terminated by this ABEND. If a dump is required to obtain more information about a violator, a SLIP trap should be set for ABEND code 077. The instance will be recorded before the program is

SETCON Command

ABENDED. If the tracking facility becomes full, no new instances are recorded, but the callers will still be ABENDED. When the ABEND occurs, a symptom record will be cut in LOGREC.

Note: If the track value is 0 or 128, no ABEND will be issued even when you specify ONWITHABEND.

MONITOR or MN

Controls whether monitor messages are to be enabled or disabled, as defined by each of the specified message types.

,JOBNAMES

The name of the job is displayed whenever the job starts and terminates, including unit record allocation when the step starts. If a job terminates abnormally, the job name will appear in a diagnostic message.

,SESS

The TSO/E user identifier is displayed whenever the TSO/E session begins and ends. If the session terminates abnormally, the user identifier appears in the diagnostic message.

,STATUS

The data set names and volume serial numbers of data sets with dispositions of KEEP, CATLG, or UNCATLG are displayed whenever data sets are freed.

,T For monitor messages that can optionally contain a timestamp, the timestamp is included in the message.

ON

Controls whether monitor messages for the specified message type are to be enabled, or are to include a timestamp.

,LOG

Monitor messages are also to be sent to the SYSLOG or OPERLOG.

,NOLOG

Monitor messages are not to be sent to the SYSLOG or OPERLOG.

OFF

Controls whether monitor messages for the specified message type are to be disabled, or are not to include a timestamp. Note that when a request to disable this message type is made, production of these messages is disabled only if there are no consoles in the sysplex currently receiving this message type.

SETDMN Command

Important

Beginning with z/OS V1R3, WLM compatibility mode is no longer available. As the SETDMN command was valid only on systems operating in compatibility mode, it is now disabled. The information has been left here for reference purposes, and for use on backlevel systems.

Use the SETDMN command to change existing values of parameters in a single domain. Issue the SETDMN command only at the direction of the system programmer. The keywords that are valid for a given execution of the SETDMN command are determined by:

1. The keywords specified in the current domain description table.
2. The values specified in the current installation performance specification (IPS).

At the system programmer's direction, using the SETDMN command, you can change the relative service distribution among domains. The relative service is specified as a range of service rates for each domain, or as a fixed contention index (FIXCIDX). Each relative service rate pair can be specified as an average service per ready address space (ASRV) in the domain or as domain service totals (DSRV). FIXCIDX is specified as a constant value which determines the relative importance of the domain, regardless of the amount of service the domain consumes.

Note: The SETDMN command is not valid on systems operating in workload management goal mode. The command is supported on systems operating in workload management compatibility mode.

Syntax

The syntax of the SETDMN command is:

```
SD domainnum,{MIN=n1[,MAX=n2][,ASRV=(n0,n9)] }  
|,DSRV=(n0,n9)  
|,FIXCIDX=nnn  
{ {ASRV=(n0,n9)} [,MIN=n1][,MAX=n2] }  
{ {DSRV=(n0,n9)} }  
{ {FIXCIDX=nnn} }  
{ MAX=n2[,ASRV=(n0,n9)][,MIN=n1] }  
|,DSRV=(n0,n9)  
|,FIXCIDX=nnn
```

Restrictions

There are no defaults in the SETDMN command.

At least one keyword must be specified.

Duplicate keywords cannot be specified.

Keywords can be specified in any order.

SETDMN Command

Specifying the ASRV, DSRV, or FIXCIDX keywords overrides any previous value set for them either in the current IPS or in another SETDMN command.

The value for *n2* in the MAX=*n2* parameter must be greater than or equal to the value of *n1*.

Parameters

domainnum

The domain table entry (1-128) to be modified.

MIN=*n1*

The minimum multiprogramming level (0-999).

MAX=*n2*

The maximum multiprogramming level (0-999).

ASRV=(*n0,n9***)**

Allows you to specify the average service per ready address space in the domain. The value range is 0-999999999.

DSRV=(*n0,n9***)**

Allows you to specify the total service rate for each domain. The value range is 0-999999999.

FIXCIDX=*nnn*

Allows you to specify the fixed contention index value for each domain. The value range is 0-655.

Note: For more details about using SETDMN command parameters, see *z/OS MVS Initialization and Tuning Guide*.

Example 1

To set the maximum multiprogramming level (MPL) to 2 in domain 5, enter:

SETDMN 5,MAX=2

All other values in domain 5 remain unchanged.

Example 2

To set the minimum MPL to 3 and the maximum MPL to 4 in domain 6, enter:

SD 6,MIN=3,MAX=4

Example 3

For domain 2, to set the minimum MPL to 0, the maximum MPL to 255, and the relative service at (1,5000) to control the average service rate per domain, enter:

SD 2,MIN=0,MAX=255,ASRV=(1,5000)

Example 4

To set the contention index of domain 4 to a constant value of 300, enter:

SD 4,FIXCIDX=300

SETETR Command

Use the SETETR command to enable external time reference (ETR) ports that have been disabled. An ETR port disabled by a hardware problem can be enabled after the problem has been corrected.

Also you might use SETETR to indicate to MVS that an adjustment has been made to the time from the 9037 Sysplex Timer. This use of SETETR is necessary for an MVS system using the 9037 Sysplex Timer and when it is running on a processor that follows:

- 3090 model J's
- 9121-320 based models
- 9021-340 based models

Syntax

The complete syntax for the SETETR command is:

```
SETETR PORT=n
```

Note: The SETETR command does not have an abbreviation.

Parameters

PORT=n

Specifies the number of the ETR port to be enabled. The valid values for *n* are 0 and 1.

Example

To enable ETR port 1, enter:

```
SETETR PORT=1
```

SETGRS Command

SETGRS Command

- Use the SETGRS command to:
- Migrate a currently active global resource serialization ring complex to a star complex
 - Modify the current RESMIL or TOLINT values
 - Set the system values for
 - GRSQ
 - SYNCHRES
 - ENQMAXA
 - ENQMAXU
 - Change the contention notifying system (CNS) in a global resource serialization complex.

Syntax

The complete syntax for the SETGRS command is:

```
SETGRS {MODE=STAR}
        {[RESMIL=nnnnnnnn | RESMIL=OFF] [,TOLINT=nnnnn] [,SYNCHRES={YES|NO}]}
        {[ENQMAXA|ENQMAXU=nnnnnnnn] [, {NOPROMPT|NP}]}
        {[CNS=sysname] [, {NOPROMPT|NP}]}
        {[GRSQ=ALL|CONTENTION|LOCAL]}
```

Note: The installation's system programmer should direct use of this command.

Parameters

MODE=STAR

Directs the system to convert a global resource serialization ring complex to a global resource serialization star complex.

MODE=STAR is mutually exclusive with the RESMIL and TOLINT parameters.

RESMIL=nnnnnnnn | RESMIL=OFF

Specifies the RSA-message residency time. The value indicates the minimum RSA-message residency time in milliseconds (that is, the least amount of time that the RSA-message is to spend in this system). The actual amount of time that the RSA-message is to spend in this system will vary between the time you specify in milliseconds and a maximum value calculated by global resource serialization. In this way, global resource serialization balances CPU use and ENQ response time.

If you specify RESMIL=OFF, the RSA-message residency time is set to zero and global resource serialization does no tuning. If you specify RESMIL=0, the system tunes the residency time in a range with a minimum of zero.

The RESMIL value can be from 0 to 99999999 milliseconds, or OFF. If you omit the RESMIL parameter, the current RESMIL value remains in effect. The current value was specified either by the GRSCNFxx parmlib member or by a previous SETGRS command.

TOLINT=nnnnn

Specifies, in seconds, the maximum tolerance time interval global resource serialization allows the RSA-message to return to this system, before it considers the RSA-message overdue.

The value of TOLINT can be from 1 to 86399 seconds. If you omit the TOLINT parameter, the current TOLINT value remains in effect. The current value was specified either by the GRSCNFxx parmlib member or by a previous SETGRS command.

SYNCHRES=YES | NO

Specifies whether synchronous reserve processing is activated. Action is only taken on the system where the command is issued.

ENQMAXA | ENQMAXU=nnnnnnnn [,NOPROMPT]

Assigns a new global resource serialization system enqueue maximum value for concurrent authorized (ENQMAXA) or unauthorized (ENQMAXU) requesters in a single system. This function enables you to dynamically update workload estimation for enqueue processing. Global resource serialization attempts to update the appropriate enqueue maximum as specified. The ISGADMIN service can update the maximum values for a specific address space. Global resource serialization uses the greater of the two maximums for purposes of its checking. For more information, see *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG*.

value

A required parameter specifies the new concurrent maximum ENQ request value.

The ENQMAXA range is 250,000 to 99,999,999. The default is 250,000.

The ENQMAXU range is 16,384 to 99,999,999. The default is 16,384. See *z/OS MVS Planning: Global Resource Serialization* for complete guidance information.

NOPROMPT

An optional parameter that informs the system not to issue the confirmation message.

Abbreviation: NP

CNS=system-name [,NOPROMPT]

CNS assigns a new global resource serialization contention notifying system (CNS) in a star complex.

Restriction: CNS is only applicable in a star complex.

NOPROMPT

An optional parameter that informs the system not to issue the confirmation message.

Abbreviation: NP

GRSQ={LOCAL | CONTENTION | ALL**}**

This parameter is specific to star mode and affects how quickly a dump is taken. GRSQ is system-specific; action is only taken on the system where the command is issued.

Restriction: GRSQ is only applicable in a star complex.

Notes:

1. The system on which you enter the SETGRS MODE=STAR command controls the migration.
2. When global resource serialization completes the transition to the star complex, the system issues the following message to indicate that the migration has completed and global resource serialization is active for the complex:
ISG334I GRS STAR COMPLEX INITIALIZATION COMPLETE

SETGRS Command

3. During processing of a SETGRS MODE=STAR command, no global resource requests (ENQ, DEQ, or RESERVE) will be processed. The length of time global resource serialization requesters are suspended may be several minutes, because the global resource serialization lock structure and sysplex couple data set records are going to be initialized with all of the complex-wide information, along with significant changes to the internal control block structures. IBM recommends invoking the migration capability at a time of minimal global resource serialization activity.
4. A SETGRS MODE=STAR request is valid if the following criteria are met:
 - Global resource serialization is running a ring complex.
 - All systems in the global resource serialization ring complex support a star complex.
 - There are no systems in the global resource serialization ring complex that are interconnected through the global resource serialization channel-to-channel support rather than the coupling facility.
 - All systems can access the ISGLOCK lock structure on the coupling facility.
 - The global resource serialization records are defined on the sysplex couple data set.
 - There are no dynamic RNL changes still in progress.
5. The RESMIL and TOLINT parameters are not valid on a SETGRS command issued in a global resource serialization star complex.
6. The RESMIL and TOLINT parameters of the SETGRS command affect only the system on which the SETGRS command is issued.
7. A SETGRS CNS=sysname request is only valid if the following criteria are met:
 - Global resource serialization is running in star complex.
 - All systems in the global resource serialization star complex support SETGRS CNS command (systems are at or above z/OS V1R7 with the required PTF).
 - Target system is an active system in the global resource serialization star complex.

Examples

Example To migrate from a global resource serialization ring complex to star complex, enter:

```
SETGRS MODE=STAR
```

Example

```
SETGRS CNS=TEST2
XX ISG366D CONFIRM REQUEST TO MIGRATE
THE CNS TO TEST2. REPLY SYSNAME=TEST2
TO CONFIRM OR C TO CANCEL.
R XX,CNS=TEST2
Result: (when current CNS is TEST1)
ISG364I CONTENTION NOTIFYING SYSTEM MOVED
FROM SYSTEM TEST1 TO SYSTEM TEST2. OPERATOR COMMAND INITIATED.
```

Example

```
SETGRS CNS=TEST2,NP
Result: (when current CNS
is TEST1) ISG364I CONTENTION NOTIFYING SYSTEM
MOVED FROM SYSTEM TEST1 TO SYSTEM TEST2.
OPERATOR COMMAND INITIATED.
```

Example

```
| SETGRS ENQMAXU=25000  
| XX ISG366D CONFIRM REQUEST TO SET THE ENQMAXU ON  
| SYSTEM TEST1 TO 25000.  
| REPLY ENQMAXU=25000 TO CONFIRM OR C TO CANCEL.
```

```
| R XX,ENQMAXU=25000  
| result: ISG370I ENQMAXU ON SYSTEM  
| TEST1 HAS BEEN SET TO 25000.
```

Example

```
| SETGRS ENQMAXA=400000,NP  
| result: ISG370I ENQMAXA FOR SYSTEM  
| TEST1 HAS BEEN SET TO 400000.
```

Example

```
| SETGRS GRSQ=CONTENTION  
| result: ISG370I GRSQ FOR SYSTEM TEST1 HAS BEEN SET TO CONTENTION.
```

Example: The GQSCAN parameters for a dump with GRSQ requested is one of the following:

```
| LOCAL: XSYS=NO  
| CONTENTION: WAITCNT=1  
| ALL: neither of the above
```

SETIOS Command

In contrast to the SET command, which allows an installation to specify a different IECIOSxx parmlib member, the SETIOS command can dynamically add a parameter, as well as delete, modify, or replace any previously-specified missing interruption handler (MIH) or I/O timing (IOT) parameter. The parameters can appear in any order in the command, but there can only be one DEV and TIME parameter pair or DEV and IOTIMING pair in a command. You can create user classes for particular situations such as test environments and special job processing.

In addition, you can use the SETIOS command to do the following:

- Enable or disable the dynamic channel path management function.
- Refresh the control unit model table for the single point of failure detection function in dynamic channel path management.
- Enable or disable the gathering of FICON switch statistics.
- Enable or disable the MIDAW facility.
- Enable or disable UCB overlay protection.
- Indicate whether IOS blocks are obtained in 24 or 31 bit storage.
- Indicate the actions to be taken for an IO Timing HyperSwap trigger.
- Indicate the use of HyperPAV mode for the system.
- Enable or disable the encryption key management.

Syntax

The complete syntax for the SETIOS command is:

```
SETIOS [MIH[,class=mm:ss[,class=mm:ss]...]
[,MOUNTMSG={YES|NO}]
[,DEV={(/devnum[,/devnum]...)},TIME=mm:ss,IOTIMING=mm:ss]
{(/lowdevnum-/highdevnum)}
[,MSGONLY={YES|NO}]
[,IOTHSWAP={YES|NO}[,IOTTERM={YES|NO}]]
[DCM={ON|OFF|REFRESH}]
[MIDAW={YES|NO}]
[FICON,STATS={YES|NO}]
[CAPTUCB,PROTECT={YES|NO}]
[STORAGE,IOSBLKS={24|31}]
[HYPERPAV={NO|YES|BASEONLY}]
[EKM[,PRIMARY={host_name[:port]}]
{ip_address[:port]}
{NONE}
[,SECONDARY={host_name[:port]}]
{ip_address[:port]}
{NONE}
[,MAXCONN=dd1]
[,MAXPCONN=dd2]]
```

Notes:

1. The SETIOS command does not have an abbreviation.
2. DEV and TIME together specify a user device class for one or more devices.
3. DEV and IOTIMING together specify a user device class for one or more devices.

4. During IPL (if the device is defined to be ONLINE), or during the VARY ONLINE process, some devices may present their own MIH timeout values, via the primary/secondary MIH timing enhancement contained in the self-describing data for the device. The primary MIH timeout value is used for most I/O commands. However, the secondary MIH timeout value may be used for special operations such as long-busy conditions or long-running I/O operations. Any time a user specifically sets a device or device class to have an MIH timeout value that is different from the IBM-supplied default for the device class, that value will override the device-established primary MIH time value. This implies that if an MIH time value that is equal to the MIH default for the device class is explicitly requested, IOS will NOT override the device-established primary MIH time value. To override the device-established primary MIH time value, you must explicitly set aside a time value that is not equal to the MIH default for the device class.

Note that overriding the device-supplied primary MIH timeout value may adversely affect MIH recovery processing for the device or device class. Please refer to the specific device's reference manuals to determine if the device supports self-describing MIH time values.
5. IOHSWAP and IOTTERM together specify how an I/O timing timeout condition is handled with respect to triggering a HyperSwap.

Parameters

The parameters are:

MIH, class=mm:ss

Specifies the time interval in the form *mm:ss*, where *mm* is minutes and *ss* is seconds. The value range for *mm* is 00-99 and for *ss* is 00-59. When you set a class to 00:00, MIH or IOT no longer monitors the class.

You can specify the time interval for one or more of the following classes:

CHAR

The character reader device class.

COMM

The communications device class.

CTC

The channel-to-channel device class.

DASD

The DASD device class. This device class name represents the MIH.

GRAF

The graphics device class.

TAPE

The tape drive device class.

UREC

The unit record device class.

USnn

A user-specified device class, where nn can be any two-digit number from 01 through 99 that matches a device group created by MIH or I/O timing processing. A user-specified device group is a set of devices associated with a specific time interval. The system creates this type of group and assigns the user class number (USnn) when either of the following is true:

- The MIH time interval is not equal to the time interval of its device class.

SETIOS Command

(Note that some devices present their own MIH timeout values, via the primary/secondary MIH timing enhancement contained in the self-describing data for the device. If the primary MIH timeout value for the device does not equal the timeout value for the device class and the device's timeout value has not been altered by the user, the system will create a user-specified class to contain the timeout value for the device. The user-specified class for these devices will be created at IPL (if the device is defined to be ONLINE) or at VARY ONLINE time.)

- The IOT time interval is not equal to the time interval of its device class.

Other time intervals that you can specify using the *class* parameter are:

HALT

The time interval for halt (HSCH) and clear (CSCH) subchannel operations. Setting this device independent keyword affects all devices on the system.

IOTDASD

The I/O timing (IOT) limit for the DASD device class. The maximum I/O timing limit is 5,999 seconds.

Note: Paging devices are not supported for I/O timing.

MNTS

The time interval for monitoring 'mount pending' conditions for DASD and TAPE drives.

STND

Specifies the MIH time interval for all of the following device classes: CHAR, COMM, CTC, GRAF, TAPE, and UREC.

If you code STND following any of those class names, the value for STND overrides the values for those device classes. Similarly, if you code any of those class names following STND, the values for those device classes override the value for STND.

Note: During IOS recovery processing, the system will override your time interval specification and may issue MIH messages and MIH logrec error records at this IOS-determined interval.

MIH,MOUNTMSG={YES or NO}

Indicates whether or not the system is to display the mount pending messages. Specify YES to have the message displayed; specify NO to suppress the message display.

MIH,DEV={(/devnum,[/devnum...)} or ({/lowdevnum-[/highdevnum)}

The specific device identified by a device number, *devnum*, or all devices in the range of *lowdevnum-highdevnum*. A device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/).

MIH,TIME=mm:ss

Specifies the time interval in the form *mm:ss*, where *mm* is minutes and *ss* is seconds. The value range for *mm* is 00-99 and for *ss* is 00-59.

When you set TIME to 00:00, MIH no longer monitors the device.

If you specify TIME you must also specify DEV. The system accepts only one pair of TIME and DEV keywords per command line.

MIH,IOTIMING=mm:ss

Specifies the I/O timing limit in the form *mm:ss*, where *mm* is minutes and *ss* is seconds. The value range for *mm* is 00-99, and for *ss* is 00-59. The maximum

I/O timing limit is 5,999 seconds. When IOTIMING is set to 00:00, I/O timing is not in effect for that device or range of devices.

Note: Do not modify the I/O timing limits without first checking with your system programmer.

If you specify IOTIMING you must also specify DEV. The system accepts only one pair of IOTIMING and DEV keywords per command line.

MIH,MSGONLY={YES or NO}

Specifies whether an I/O timeout condition is processed using message-only recovery (MSGONLY=YES) or full I/O timing recovery (MSGONLY=NO).

Message-only processing allows the system to detect I/O timeout conditions while providing the user the ability to decide which I/O requests the system should terminate.

When an I/O request exceeds the I/O timing interval, the system issues a message to the operator and writes a record to SYS1.LOGREC. Then,

1. When MSGONLY=YES is specified, the I/O request is left in the system.
2. When MSGONLY=NO is specified, the system abnormally terminates the I/O request.

The default, when you do not specify MSGONLY, is MSGONLY=NO. The system applies this only to devices it modifies as a result of this command.

If a command contains more than one MSGONLY keyword, the system uses only the last valid MSGONLY keyword.

The MSGONLY keyword is valid only when you specify the IOTDASD keyword or the DEV and IOTIMING keywords. Otherwise, the system ignores MSGONLY. That is, the MSGONLY keyword value relates only to devices affected by the IOTDASD or the DEV and IOTIMING keywords.

MIH,IOTHSWAP={YES or NO}[,IOTTERM={YES or NO}]

Specifies how an I/O timeout condition is handled with respect to HyperSwap processing.

IOTHSWAP indicates whether an I/O timing timeout condition is allowed to trigger a HyperSwap. IOTTERM indicates whether a timed-out I/O operation should be terminated with permanent error when a HyperSwap has been triggered for the I/O timing timeout condition.

The IOTTERM keyword is valid only when you specify the IOTHSWAP keyword on the same command.

Note: An I/O timeout does not trigger a GDPS HyperSwap when message-only recovery is specified for the device or as the result of a timeout condition specified by an I/O driver program.

DCM={ON or OFF or REFRESH}

Specifies that dynamic channel path management is be turned on or off. If REFRESH, then a control unit model table update will be initiated.

MIDAW={YES or NO}

Specifies whether the modified indirect addressing word (MIDAW) facility is enabled or disabled on a system. When disabling with MIDAW=NO, the MIDAW facility will remain in effect for one minute to allow queued I/O using MIDAWs to finish.

SETIOS Command

FICON,STATS={YES or NO}

Specifies whether FICON switch statistics are to be gathered on a system. When specifying FICON,STATS=NO, turn off FICON Director Activity Reporting in Resource Measurement Facility (RMF) in order to avoid the possibility of inconsistent report data.

CAPTUCB,PROTECT={YES or NO}

Specifies whether to enable write protection on captured UCBs.

STORAGE,IOSBLKS={24 or 31}

Use this command to enable 24 or 31-bit storage for IOS blocks.

HYPERPAP={NO or YES or BASEONLY}

Specifies the use of HyperPAV mode. Use of this keyword changes the mode of operation of HYPERPAP-capable DASD control units to the requested mode.

HYPERPAP=NO

Specifies that HyperPAV mode is not to be used.

HYPERPAP=YES

Specifies that HyperPAV mode is to be used.

HYPERPAP=BASEONLY

Specifies that I/O is to be run only on non-PAV-alias devices in HyperPAV mode.

If the HYPERPAP keyword is not specified, the current HYPERPAP setting is not altered.

Notes:

1. Using SETIOS HYPERPAP to change the mode of operation of all DASD control units on the system can take considerable time, depending on how many control units are configured on the system. This operation, if required, should be done during periods of lower system utilization, and should be done without concurrent IODF or microcode changes affecting the control unit.
2. If all devices in a logical control unit (LCU) are offline at IPL, SETIOS HYPERPAP mode changes will only take effect after a device on that LCU is varied online.
3. If a dynamic ACTIVATE and a SETIOS affect the devices on the same control unit, the aliases might not be converted to the correct mode. To detect this condition, use the D M=DEV command and examine the output. To correct the problem, issue VARY *bbbb* ONLINE,UNCOND where *bbbb* is a base device on the affected control unit. For more information about the error condition and how to correct it, see "Placing an I/O Device or a Range of I/O Devices Online or Offline" on page 4-633.

EKM,PRIMARY={*host_name[:port]* or *ip_address[:port]* or NONE}

Specifies the hostname or IP address and port number of the primary key manager. The primary host is used exclusively until a failure is encountered and all attempts to retry are unsuccessful. In subsequent requests after a failure, the primary will be retried before the use of the secondary is attempted. When a connection to the primary is re-established normal operation continues.

host_name

The host name of the encryption key manager.

port

The port number of the encryption key manager. If you include a port

number, use a colon to separate it from the host name or IP address. If the port number is not specified, the default of 3801 is used.

ip_address

The IP address of the host (IP addresses must be specified as a dotted decimal quad ddd.ddd.ddd.ddd).

NONE

Disable the encryption key manager, which is the default.

EKM,SECONDARY={*host_name[:port]* or *ip_address[:port]* or NONE}

Specifies the hostname or IP address and port number of the secondary key manager.

host_name

The host name of the encryption key manager.

port

The port number of the encryption key manager. If you include a port number, use a colon to separate it from the host name or IP address. If the port number is not specified, the default of 3801 is used.

ip_address

The IP address of the host (IP addresses must be specified as a dotted decimal quad ddd.ddd.ddd.ddd).

NONE

Disable the encryption key manager, which is the default.

EKM,MAXCONN=*dd1*

Specifies the maximum number of concurrent socket connections for encryption key management. If in the situation of high network stress due to high socket utilization for encryption key management this number can be lowered.

Value range: 1-255

Default: 255

EKM,MAXPCONN=*dd2*

Specifies the maximum number of permanent concurrent socket connections for encryption key management. The permanent connections remain open to prevent the overhead of opening and closing socket communication.

Value range: 0- the number specified in MAXCONN

Default: 8

Notes:

1. You can specify more than one parameter as long as the length of the command does not exceed 124 characters.
2. The SET IOS, SETIOS, and DISPLAY IOS commands cannot run concurrently. The system processes the first command only.

Example 1

Change the setting of several classes:

```
SETIOS MIH,CTC=01:00,STND=04:00,DASD=00:10,HALT=00:08,TAPE=05:00
```

This command sets time intervals as follows:

CHAR, COMM, CTC, GRAF, and UREC device classes: 4 minutes, 0 seconds
 DASD device class: 0 minutes, 10 seconds

SETIOS Command

HSCH and CSCH I/O instructions: 0 minutes, 8 seconds for all devices in the system
TAPE device class: 5 minutes, 0 seconds

Note that the value for the CTC device class is 4:00, because the value specified for STND overrides the value specified for CTC (STND is coded after CTC on the SETIOS MIH command). However, the value for the tape device class is 5:00, because the value specified for TAPE overrides the value specified for STND. (TAPE is coded after STND on the SETIOS MIH command.)

All other classes remain unchanged.

Example 2

Change the setting of one class and an option:

```
SETIOS MIH,UREC=02:00,MOUNTMSG=YES
```

This command sets a time interval of 2 minutes, 0 seconds for unit record devices, and specifies that the system should display all mount pending messages. Time intervals for all classes other than UREC remain unchanged.

Example 3

Change the setting of one device with a 4-digit device number:

```
SETIOS MIH,DEV=/4472,TIME=01:10
```

This command sets an MIH time interval of 1 minute, 10 seconds for device 4472. All other classes remain unchanged.

Example 4

Create a user class for a device range:

```
SETIOS MIH,DEV=(431-435),TIME=00:45
```

This sets an MIH time interval of 45 seconds for devices 431 through 435. All other classes remain unchanged.

Example 5

To set the I/O timing limit to 2 minutes and 30 seconds for device 008, enter:

```
SETIOS MIH,DEV=008,IOTIMING=02:30
```

Example 6

Establish an I/O timing limit of 10 minutes for all non-paging DASD devices. Also, establish message-only processing for all DASD devices.

```
SETIOS MIH,IOTDASD=10:00,MSGONLY=YES
```

In this example, if any I/O request to any DASD device exceeds the ten minute I/O limit, the system issues a message and records the condition in SYS1.LOGREC, but does NOT abnormally terminate the request. Instead, the system retains the request. Then, if another I/O timing interval expires, the system will again issue a message and record the condition in SYS1.LOGREC.

Example 7

SETIOS Command

Establish an I/O timing limit of one minute for all non-paging DASD devices. Also, set up an I/O timing limit of thirty seconds for devices 180 through 18F.

```
SETIOS MIH,IOTDASD=01:00,DEV=(180-18F),IOTIMING=00:30
```

Note in this example that because MSGONLY is not specified, if I/O timing message-only processing had previously been active on any device this SETIOS command is processing, message-only processing will be reset and full I/O timing recovery will now occur.

SETLOAD Command

The SETLOAD command allows you to switch dynamically from one parmlib concatenation (logical parmlib) to another without having to initiate an IPL. The SETLOAD command specifies the LOADxx member that contains the PARMLIB statements to use for the switch.

Syntax

The complete syntax for the SETLOAD command is:

```
SETLOAD xx,PARMLIB[,{DSNAME|DSN}=dsn][,{VOLUME|VOL|VOLSER}=vol]
```

Parameters

The parameters are:

- xx Specifies the one or two character suffix used to identify the LOADxx member that you want to process.

PARMLIB

Specifies that the system is to process the PARMLIB statements in the LOADxx member according to the filter parameters (HWNAME, LPARNAM, VMUSERID). For more information on filter parameters, see the LOADxx member in *z/OS MVS Initialization and Tuning Reference*.

DSNAME or DSN =dsn

Specifies the 1 to 44 character name of the data set where the LOADxx member resides.

The default is to locate the LOADxx member specified in a data set within the existing parmlib concatenation.

VOLUME or VOL or VOLSER =vol

Specifies the 1 to 6 character serial number identifier of the volume where the specified data set resides.

The default is to locate the data set by the volume information in the master catalog.

Note: After the parmlib changes, the DISPLAY PARMLIB command will no longer show either the master JCL or any errors that occurred during the IPL.

Example 1

Dynamically change the parmlib concatenation

```
SETLOAD 02,PARMLIB
```

This command tells the system to process the PARMLIB statements in member LOAD02, which resides in a data set in the existing parmlib concatenation.

Example 2

```
SETLOAD 03,PARMLIB,DSN=sys4.relson
```

This command tells the system to process the PARMLIB statements in member LOAD03. Member LOAD03 resides in the data set "sys4.relson" which is catalogued in the master catalog.

Example 3

```
SETLOAD 04,PARMLIB,DSN=sys5.relson,VOL=123456
```

This command tells the system to process the PARMLIB statements in member LOAD04. Member LOAD04 resides in the data set "sys5.relson" which can be located on volume "123456."

Note: When a SETLOAD command is issued and fails, messages issued by IEFPRMLB (Logical parmlib Service) that contain jobname and stepname will contain Master's jobname and the stepname of the last step that ran under Master. This is because the SETLOAD command runs under Master but does not run as its own step. In this case the error is related to the SETLOAD processing and NOT the step whose name appears in the message.

SETLOGR Command

Use the SETLOGR command to control z/OS MVS System Logger resources. Table 4-30 summarizes the information that the SETLOGR command provides. Use it to access the pages on which you can find details about a particular use of the SETLOGR command.

Table 4-30. Summary of the SETLOGR Command

Command:	Topic:
SETLOGR FORCE	"SETLOGR FORCE Command"

Scope in a Sysplex

The following table describes the conditions under which the SETLOGR command has sysplex scope. See "Using Commands That Have Sysplex Scope" on page 1-11 for an explanation of sysplex scope.

Table 4-31. Sysplex Scope for the SETLOGR Command

Command:	Topic:
SETLOGR FORCE, DELETE,LSName=	Has sysplex scope because it removes the named log stream from the LOGR couple data set (CDS).

Syntax

The syntax for each variation of the SETLOGR command is shown immediately preceding its respective parameter description.

SETLOGR FORCE Command

Use the SETLOGR FORCE command to clean up log stream resources related to a system logger log stream when the log stream becomes unusable. Logger will attempt to release all the related resources for the log stream based on the request.

The following exemplify two key situations when a log stream might be regarded as unusable:

1. If logger is unable to recover failed-persistent connections the log stream can be left in a failed-persistent connection state, so IXCMIAPIU DATA TYPE(LOGR) DELETE LOGSTREAM requests are unable to remove the log stream and all associated resources from the logger inventory (LOGR CDS).

Symptoms to recognize this type of situation:

D Logger,C,LSN=log-stream-name

shows number of connectors on this system equal to 0.

D Logger,L,LSN=log-stream-name

shows number of connectors to the logstream greater than 0.

If both of these symptoms are present, the logstream has failed-persistent connections and a SETLOGR FORCE,DELETE operation may be necessary to delete the logstream.

2. A log stream is left in a DISCONNECT PENDING state on a system.

Symptoms to recognize this type of situation:

D Logger,C,LSN=log-stream-name

shows logstream status in "disconnect pending" state.

If the logstream is in "disconnect pending" state a SETLOGR FORCE,DISCONNECT operation may be necessary to disconnect the logstream.

Attention: To reduce the risk of losing data, do not force the disconnection of a logstream from a system or force the deletion of a log stream unless you understand its use in the sysplex by applications or subsystems. Note that forcing connections from a system can affect active connectors (subsystems, applications) to the named log stream.

See "SETXCF FORCE Command" on page 4-487 to clean up resources related specifically to structures in a coupling facility. For additional information about the circumstances under which to issue the SETXCF or SETLOGR FORCE commands, see Coupling Facility Replacement and Reconfiguration Guidelines in *z/OS MVS Setting Up a Sysplex*.

Syntax

```
SETLOGR FORCE, {DISCONNECT | DISC | DEL | DELETE} ,{LSN | LSNNNAME}=logStreamName
```

Parameters

The parameters are:

DISCONNECT or DISC

Directs the system to remove (disconnect) all the connections to the named log stream on the system from which you issued the command. Note that the force connections from a system command can affect active connectors (subsystems, applications) to the named log stream. You can use the FORCE, DISCONNECT command before deleting the log stream resource from the LOGR CDS.

When active connectors exist for the log stream, on a system where the force disconnect command is directed, Logger will first quiesce the connectors' activity for the log stream and then disconnect the log stream from the system. If Logger is unable to complete the logstream disconnect on the system, it may be necessary to issue another SETLOGR FORCE,DISCONNECT command.

Issuing the SETLOGR FORCE,DISCONNECT command when an offload is being held up, or a task is not responding, and any of the following messages are present: IXG271I, IXG272E, IXG311I, IXG312E, IXG114A, IXG115A, may cause ABEND47B or other ABENDs. Respond to these messages before issuing the SETLOGR FORCE,DISCONNECT command. See *z/OS MVS Setting Up a Sysplex* for more information about these messages.

DELETE or DEL

Directs the system to force the deletion of a named log stream from the LOGR couple data set. You can use SETLOGR FORCE,DELETE only to delete a log stream with no connections or with only failed-persistent connections remaining. It may be necessary to use the SETLOGR FORCE,DISCONNECT command on systems where normal log stream disconnections are not responsive.

If you issue the SETLOGR FORCE,DELETE command and the operation is unable to continue after the log stream had already been marked in the LOGR CDS as started to be deleted, then future attempts to connect to the log stream will fail and the log stream delete operation will be re-attempted at that time.

There might be cases when FORCE DELETE completes successfully, but the system is unable to clean up the following resources:

SETLOGR Command

Staging Datasets

Cleanup will be attempted if the logstream is connected to again.

Structure Connections

Cleanup will be attempted if the logstream is connected to again.

Offload Datasets

The dataset resources can be cleaned up manually, by deleting datasets identified as 'orphans' in the IXCMIAPU TYPE(LOGR) LIST Report.

LSNAME or LSN=*logstreamname*

Identifies the log stream resource to be acted upon.

Example 1, SETLOGR FORCE,DISC,LSN=SYSPLEX.OPERLOG

Initial display shows the logstream is in Disconnect Pending State.

```
SY1 d logger,c,lsn=SYSPLEX.OPERLOG
SY1 IXG601I 12.42.53 LOGGER DISPLAY 459
CONNECTION INFORMATION BY LOGSTREAM FOR SYSTEM SY1
LOGSTREAM           STRUCTURE      #CONN   STATUS
-----             -----        -----  -----
SYSPLEX.OPERLOG       LIST01        000001  DISCONNECT PENDING

NUMBER OF LOGSTREAMS: 000001
```

Force disconnect command is entered.

```
SY1 setlogr force,disc,lsn=SYSPLEX.OPERLOG
SY1 IXG651I SETLOGR FORCE DISCONNECT COMMAND ACCEPTED
FOR LOGSTREAM=SYSPLEX.OPERLOG
SY1 IXG661I SETLOGR FORCE DISCONNECT PROCESSED SUCCESSFULLY
FOR LOGSTREAM=SYSPLEX.OPERLOG
```

A final display command shows the logstream is no longer connected.

```
SY1 d logger,c,lsn=SYSPLEX.OPERLOG
SY1 IXG601I 12.43.15 LOGGER DISPLAY 466
CONNECTION INFORMATION BY LOGSTREAM FOR SYSTEM SY1
LOGSTREAM           STRUCTURE      #CONN   STATUS
-----             -----        -----  -----
NO MATCHING INFORMATION FOUND.
```

SETLOGRC Command

Use the SETLOGRC command to change the logrec error and environmental recording medium originally specified in the IEASY\$xx parmlib member during initial program load (IPL). You can specify one of the following options for logrec error recording:

- LOGSTREAM
- DATASET
- IGNORE

Once the system processes the command, one of the following can occur:

- If the change of medium is successful, the system issues message IFB097I to indicate the change and the new medium to the requesting console.
- If the change of medium is to DATASET and the system was not originally initialized with a data set specified as the recording medium, the system issues message IFB099I to indicate that the medium was not changed and that a data set was not defined to be used as a logrec data set.
- If the invoker is attempting to set the logrec recording medium to a setting that happens to be the current setting, the system issues message IFB096I to the invoking console to indicate that the desired medium is the current setting.

Note: There is one exception. If the current and desired settings are both to LOGSTREAM, the system issues message IFB094I stating that the command has been accepted. If the connection to the log stream fails because system logger is unavailable, the system issues message IFB100E and internally buffers logrec records until the system logger becomes available. The recording medium remains LOGSTREAM.

- If the desired setting is to LOGSTREAM and the connection to the log stream fails, the system issues message IFB094I to indicate the successful change of medium from LOGSTREAM to LOGSTREAM. If the change of medium is unsuccessful, the system issues message IFB099I. The system also issues message IFB100E to indicate that the system logger is unavailable. Logrec error and environmental records will be internally buffered until the system logger becomes available.
- If the desired setting is to IGNORE, logrec error and environmental records will not be recorded and will not be provided in an ENF 36 signal.

Note: IBM recommends that you use the IGNORE setting in testing environments only.

Syntax

The complete syntax for the SETLOGRC command is:

```
SETLOGRC {LOGSTREAM|DATASET|IGNORE}
```

Note: The SETLOGRC command does not have an abbreviation.

Parameters

The parameters are:

LOGSTREAM

Indicates that the desired medium for recording logrec error and environmental

SETLOGRC Command

records is a log stream. To use a log stream your installation must be operating at an MVS/ESA SP 5.2.0 level or higher and the logrec log stream must be defined. See *z/OS MVS Setting Up a Sysplex* for information about logrec log stream definitions.

DATASET

Indicates that the desired medium for recording logrec error and environmental records is a data set, which is the medium used prior to MVS/ESA SP 5.2.0. Setting the medium to data set works only if the system had originally been initiated with a data set as the logrec recording medium. If the system was not initiated with a data set logrec recording medium and the attempt is made to change to DATASET, the system rejects the attempt and maintains the current logrec recording medium.

IGNORE

Indicates that recording logrec error and environmental records is to be disabled.

Note: IBM recommends that you use this setting only in a test environment.

SETOMVS Command

Use the SETOMVS command to change dynamically the options that z/OS UNIX System Services currently is using. These options are originally set in the BPXPRMxx parmlib member during initial program load (IPL). For more information on the BPXPRMxx parmlib member, see *z/OS UNIX System Services Planning*.

Changes to all of the system-wide limits take effect immediately. When a process limit is updated, all processes that are using the system-wide process limit have their limits updated. All process limit changes take effect immediately except those processes with a user-defined process limit (defined in the OMVS segment or set with a SETOMVS PID= command). Exceptions are MAXASSIZE and MAXCPUTIME, which are not changed for active processes.

Note: If a process-level limit is lowered with the SETOMVS command, some processes may immediately hit 100% usage. Depending on the process limit specified and what the process is doing, this could cause some processes to fail.

Syntax

The complete syntax for the SETOMVS command is:

SETOMVS Command

SETOMVS	SETOMVS EXTENSIONS (sysplex exclusive)
<pre> SETOMVS [AUTHPGMLIST='authprogramlist' NONE] [,FORKCOPY=(COPY COW)] [,IPCSEMNIDS=ipcsemnids] [,IPCSEMNOPTS=ipcsemnops] [,IPCSEMNSEMS=ipcsemnsems] [,IPCMMSGQBYTES=ipcmmsgqbytes] [,IPCMMSGNIDS=ipcmsgnids] [,IPCSHMMPPAGES=ipcschmmpages] [,IPCSHMNIDS=ipcschmnids] [,IPCSHMNSEGS=ipcschmnsegs] [,IPCSHMPAGES=ipcschmmpages] [,IPCMMSGQNUM=ipcmmsgqnum] [,LIMMSG=[NONE SYSTEM ALL]] [,MAXASSIZE=maxassize] [,MAXCORESIZE=maxcoresize] [,MAXCPUTIME=maxcpitime] [,MAXFILEPROC=maxfileproc] [,MAXFILESIZE=(maxfilesize NOLIMIT)] [,MAXMMAPAREA=maxmmaparea] [,MAXPROCSYS=maxprocsys] [,MAXPROCUSER=maxprocuser] [,MAXPTYS=maxptys] [,MAXSHAREPAGES=maxsharepages] [,MAXTHREADS=maxthreads] [,MAXTHREADTASKS=maxthreadtasks] [,MAXUIDS=maxuids] [,PID=pid,processlimitname=newvalue] [,PRIORITYGOAL=(n) NONE] [,RESET=(xx)] [,SERV_LPALIB=('dsname','volser')] [,SERV-LINKLIB=('dsname','volser')] [,SHRLIBRGNsize=shrlibrgnsize] [,SHRLIBMAXPAGES=shrlibmaxpages] [,STEPLIBLIST='stepliblist'] [,SUPERUSER=superuser] [,SYNTAXCHECK=(xx)] [,SYSCALL_COUNTS=(YES NO)] [,TTYGROUP=ttygroup] [,USERIDALIASTABLE='useridaliastable'] [,VERSION='string'] </pre>	<pre> SETOMVS FILESYS ,FILESYSTEM=filesystem ,AUTOMOVE=YES NO UNMOUNT indicator(sysname1 ,sysname2,...,sysnameN) or SETOMVS FILESYS ,FILESYSTEM=filesystem ,SYSNAME=sysname * or SETOMVS FILESYS ,MOUNTPOINT=mountpoint ,AUTOMOVE=YES NO UNMOUNT indicator(sysname1 ,sysname2,...,sysnameN) or SETOMVS FILESYS ,MOUNTPOINT=mountpoint ,AUTOMOVE=YES NO UNMOUNT indicator(sysname1 ,sysname2,...,sysnameN) or SETOMVS FILESYS ,MOUNTPOINT=mountpoint ,SYSNAME=sysname * or SETOMVS FILESYS ,FROMSYS=sysname ,SYSNAME=sysname * </pre> <p>Notes:</p> <ol style="list-style-type: none"> 1. FILESYSTEM, FROMSYS, and MOUNTPOINT are mutually exclusive parameters. When you specify FILESYS, you must supply one of these three parameters. 2. SETOMVS RESET=(xx) has been changed to allow SETOMVS RESET=xx as well as SETOMVS RESET=(xx). The parentheses are now optional.

Rather than defining parameter limit values in their full decimal or hexadecimal form, you can use the following 1-character multiplier (denomination values) suffix to specify them. The system also uses this value in displays when it returns responses to respective D OMVS commands.

Notes:

1. Only those SETOMVS parameters that support this "C" suffix specifically note that support and refer to Table 4-32 on page 4-428.
2. Values that **contain** a multiplier are limited to 8 digits (nnnnnnnnC) and those values are limited to X'00FF FFFF' (16 777 215 decimal). Limits that support values above the bar have a range of 1M-16383P. However, do not exceed a parameter-specific maximum value.
3. Values that **do not contain** a multiplier are limited to X'7FFF FFFF' (2 147 483 647 decimal).

Table 4-32. 1-Character Parameter Limit Multipliers

Denomination Value	1-Character Abbreviation	Bytes
--------------------	--------------------------	-------

Table 4-32. 1-Character Parameter Limit Multipliers (continued)

null	n/a	1
Kilo	K	1,024
Mega	M	1,048,576
Giga	G	1,073,741,824
Tera	T	1,099,511,627,776
Peta	P	1,125,899,906,842,624

Parameters

AUTOMOVE = YES | NO | UNMOUNT | indicator(sysname1,sysname2,...,[sysnameN | *]), FILESYS=filesys, FILESYSTEM=filesystem, FROMSYS=sysname, MOUNTPOINT=mountpoint, SYSNAME=sysnamel*, and VERSION='nnnn' are parameters that are used in a sysplex environment where systems are exploiting shared file system. For more information on sharing file system in a sysplex, see *z/OS UNIX System Services Planning*.

The parameters are:

AUTOMOVE=YES | NO |

UNMOUNT|indicator(sysname1,sysname2,...,sysnameN)

AUTOMOVE applies only in a sysplex where systems are participating in shared file system. These parameters indicate what happens to the ownership of the file system when a shutdown, PFS termination, dead system takeover, or file system move occurs.

AUTOMOVE=YES allows the system to automatically move logical ownership of the file system as needed. AUTOMOVE=YES is the default; you can specify it as AUTOMOVE.

AUTOMOVE=NO prevents ownership movement in some situations.

AUTOMOVE=UNMOUNT unmounts the file system in some situations.

AUTOMOVE=indicator(sysname1,sysname2,...,sysnameN) specifies a list of systems to which the ownership of file system should or should not be moved when ownership of the file system changes.

- If indicator is specified as INCLUDE (or I), the list must provide a comma-delimited, priority-ordered list of systems to which ownership of the file system can be moved. For example, AUTOMOVE=INCLUDE(SYS1, SYS4, SYS9). You can specify an asterisk (*) as the last (or the only) system name to indicate any active system. For example, AUTOMOVE=INCLUDE(SYS1, SYS4, *).

Note: Do not use an asterisk in a mixed sysplex environment where any system is not at z/OS Version 1 Release 6 or later. Doing so will produce unpredictable results. The asterisk is not supported before z/OS Version 1 Release 6.

- If indicator is specified as EXCLUDE (or E), the system list must provide a comma-delimited list of systems to which the ownership of file system must not be moved. For example, AUTOMOVE=EXCLUDE(SYS3, SYS5, SYS7).

Restriction: The AUTOMOVE parameter is not permitted when using SETOMVS to move a file system.

SETOMVS Command

| **Guideline:** To ensure that the root file system is always available, use the default AUTOMOVE value (AUTOMOVE=YES).

| For more information about the behavior of the AUTOMOVE option, see *z/OS UNIX System Services Planning*.

FILESYS=filesystem

In a sysplex environment, this parameter alerts the parser that commands that change mount attributes are to follow.

For examples on the use of this parameter when making move or change requests, see *z/OS UNIX System Services Planning*.

FILESYSTEM=filesystem

In a sysplex environment, **FILESYSTEM** is the 44 character alphanumeric field that denotes the name of the filesystem to be changed or moved. This filesystem name must be in the following form: 'OMVS.USER.JOE'.

Note: The filesystem name must be in quotes, and mixed-case filesystem names are supported.

FILESYSTEM, **MOUNTPOINT**, and **FROMSYS** are mutually exclusive parameters.

For examples on the use of this parameter when making move or change requests, see *z/OS UNIX System Services Planning*.

FROMSYS=sysname

In a sysplex environment, this parameter indicates the system where all the filesystems will be moved from. The filesystems will be moved to the system identified by the **sysname** keyword. **FILESYSTEM**, **MOUNTPOINT**, and **FROMSYS** are mutually exclusive parameters.

MOUNTPOINT=mountpoint

In a sysplex environment, **MOUNTPOINT** is the mountpoint specification.

For example:

'/usr/d1'

It is case sensitive. This is the mountpoint where the filesystem is mounted. If specified, the filesystem associated with this mountpoint will be moved or changed. **FILESYSTEM**, **MOUNTPOINT**, and **FROMSYS** are mutually exclusive parameters.

For examples on the use of this parameter when making move or change requests, see *z/OS UNIX System Services Planning*.

AUTHPGMLIST='authprogramlist'NONE

Points to a z/OS UNIX file containing a list of pathnames, MVS program names, or both that allow an additional level of authorization for program-controlled or for APF-authorized programs. See *z/OS UNIX System Services Planning* for information on constructing this file. The default is NONE.

FORKCOPY = COPY | COW

Specifies how user storage is copied from the parent process to the child process during a **fork()** system call.

If you specify **FORKCOPY=COW**, all **fork()** calls are processed in copy-on-write (COW) mode if the suppression-on-protection hardware feature is available. Before the storage is modified, both the parent and child processes refer to the same view of the data. The parent storage is copied to the child as soon as storage is modified, either by the parent or the child.

Using copy-on-write causes the system to use the extended system queue area (ESQA) to manage page sharing.

If you specify **FORKCOPY=COPY**, **fork()** immediately copies the parent storage to the child, regardless of whether the suppression-on-protection feature is available. Use this option to avoid any additional ESQA use in support of **fork()**.

Follow these guidelines:

- If the run-time library is in the link pack area, specify **FORKCOPY=COPY**.
- If the run-time library is not in the link pack area, specify **FORKCOPY=COW**.

If you do not specify **FORKCOPY**, the default is **FORKCOPY=COW**.

IPCSEMNIDS = ipcsemnids

Specifies the maximum number of unique semaphore sets in the system. The range is from 1 to 20 000. The default is 500.

IPCSEMNOPS = ipcsemnops

Specifies the maximum number of operations for each semaphore operation call. The range is from 0 to 32 767. The default is 25. This is a system-wide limit.

IPCSEMNSEMS = ipcsemnsems

Specifies the maximum number of semaphores for each semaphore set. The range is from 0 to 32 767. The default is 25.

IPCMMSGQBYTES = ipcmsgqbytes

Specifies the maximum number of bytes in a single message queue. The range is from 0 to 1 048 576. The default is 262 144.

IPCMMSGNIDS = ipcmsgnids

Specifies the maximum number of unique message queues in the system. The range is from 1 to 20 000. The default is 500.

IPCSHMPAGES = ipcshmmpages

Specifies the maximum number of pages for a shared memory segment. The range is from 1 to 4P. The default is 25600.

Note: You can set a denomination (or multiplier) value when defining this value. The suffix, "C" can have a 1-character value as presented in Table 4-32 on page 4-428, but must not exceed the parameter-specific upper limit.

MVS retains the denomination value and uses it within a subsequent D OMVS command

IPCSHMNIDS = ipcshmnids

Specifies the maximum number of unique shared memory segments in the system. The range is from 1 to 20 000. The default is 500.

IPCSHMNSEGS = ipcshmnsegs

Specifies the maximum number of shared memory segments attached for each address space. The range is from 0 to 1 000. The default is 10.

IPCSHMPAGES = ipcshmmpages

Specifies the maximum number of pages for shared memory segments in the system. The range is from 0 to 2 621 440. The default is 262 144.

SETOMVS Command

Note: You can set a denomination (or multiplier) value when defining this value. The suffix, "C" can have a 1-character value as presented in Table 4-32 on page 4-428, but must not exceed the parameter-specific upper limit.

MVS retains the denomination value and uses it within a subsequent D OMVS command

IPCMMSGQNUM = ipcmsqgnum

Specifies the maximum number of messages for each message queue in the system. The range is from 0 to 20 000. The default is 10 000.

LIMMSG=(NONE|SYSTEM|ALL)

Specifies how console messages that indicate when system parmlib limits are reaching critical levels are to be displayed:

NONE No console messages are to be displayed when any of the parmlib limits have been reached.

SYSTEM

Console messages are to be displayed for all processes that reach system limits. In addition, messages are to be displayed for each process limit of a process if:

- The process limit or limits are defined in the OMVS segment of the owning User ID
- The process limit or limits have been changed with a SETOMVS PID=*pid*,process_limit

ALL Console messages are to be displayed for the system limits and for the process limits, regardless of which process reaches a process limit.

Default: NONE

MAXASSIZE = maxassize

Specifies the RLIMIT_AS hard limit resource value that processes receive when they are dubbed a process. RLIMIT_AS indicates the address space region size. The soft limit is obtained from MVS. If the soft limit value from MVS is greater than the MAXASSIZE value, the hard limit is set to the soft limit.

This value is also used when processes are initiated by a daemon process using an **exec** after **setuid()**. In this case, both the RLIMIT_AS hard and soft limit values are set to the MAXASSIZE value.

Refer to the description of **setrlimit()** in *z/OS UNIX System Services Programming: Assembler Callable Services Reference* for more information about RLIMIT_AS.

The range is from 10 485 760 (10MB) to 2 147 483 647 ; the default is 41 943 040 (40MB).

Note: You can set a denomination (or multiplier) value when defining this value. The suffix, "C" can have a 1-character value as presented in Table 4-32 on page 4-428, but must not exceed the parameter-specific upper limit.

MVS retains the denomination value and uses it within a subsequent D OMVS command

MAXCORESIZE = maxcoresize

Specifies the RLIMIT_CORE soft and hard limit resource values that processes receive when they are dubbed a process. RLIMIT_CORE indicates the

maximum core dump file size (in bytes) that a process can create. Also, it specifies the limit when they are initiated by a daemon process using an **exec** after **setuid()**.

Refer to the description of **setrlimit()** in *z/OS UNIX System Services Programming: Assembler Callable Services Reference* for more information about RLIMIT_CORE.

The range is from 0 to 2 147 483 647; the default is 4 194 304 (4MB).

Note: You can set a denomination (or multiplier) value when defining this value. The suffix, "C" can have a 1-character value as presented in Table 4-32 on page 4-428, but must not exceed the parameter-specific upper limit.

MVS retains the denomination value and uses it within a subsequent D OMVS command

MAXCPUTIME = maxcputime

Specifies the RLIMIT_CPU hard limit resource values that processes receive when they are dubbed a process. RLIMIT_CPU indicates the CPU time that a process is allowed to use, in seconds. The soft limit is obtained from MVS. If the soft limit value from MVS is greater than the MAXCPUTIME value, the hard limit is set to the soft limit. This value is also used when processes are initiated by a daemon process using an **exec** after **setuid()**. In this case, both the RLIMIT_CPU hard and soft limit values are set to the MAXCPUTIME value.

Refer to the description of **setrlimit()** in *z/OS UNIX System Services Programming: Assembler Callable Services Reference* for more information about RLIMIT_CPU.

The range is from 7 to 2 147 483 647. The default is 1 000.

Specifying a value of 2 147 483 647 indicates unlimited CPU time.

MAXFILEPROC = maxfileproc

Specifies the maximum number of files that a single user is allowed to have concurrently active or allocated. The range is 3 to 524287.

MAXFILESIZE = (maxfilesize | NOLIMIT)

Specifies the RLIMIT_FSIZE soft and hard limit resource values that processes receive when they are dubbed a process. RLIMIT_FSIZE indicates the maximum file size (in 4KB increments) that a process can create. Also, it specifies the limit when they are initiated by a daemon process using an **exec** after **setuid()**.

The range is from 0 to 524 228. If you specify 0, no files will be created by the process. Omitting this statement or specifying NOLIMIT indicates an unlimited file size.

Note: You can set a denomination (or multiplier) value when defining this value. The suffix, "C" can have a 1-character value as presented in Table 4-32 on page 4-428, but must not exceed the parameter-specific upper limit.

MVS retains the denomination value and uses it within a subsequent D OMVS command

SETOMVS Command

MAXMMAPAREA = max mmaparea

Specifies the maximum amount of data space storage (in pages) that can be allocated for memory mappings of z/OS UNIX files. Storage is not allocated until memory mappings are active.

The range is from 1 to 16 777 216. The default is 4 096.

Note: You can set a denomination (or multiplier) value when defining this value. The suffix, "C" can have a 1-character value as presented in Table 4-32 on page 4-428, but must not exceed the parameter-specific upper limit.

MVS retains the denomination value and uses it within a subsequent D OMVS command

MAXPROCSYS = max procsys

Specifies the maximum number of processes that z/OS UNIX System Services will allow to be active at the same time. The range is 5 to 32 767; the default and the value in BPXPRMXX is 200.

MAXPROCUSER = max procuser

Specifies the maximum number of processes that a single OMVS user ID (UID) is allowed to have active at the same time, regardless of how the process became a z/OS UNIX System Services process. The range is 3 to 32 767;

MAXPTYs = max ptys

Specifies the maximum number of pseudo-TTY (pseudoterminal) sessions that can be active at the same time. The range is 1 to 10 000; the default and the value in BPXPRMXX is 256.

MAXPTYs lets you manage the number of interactive shell sessions. When you specify this value, each interactive session requires one pseudo-TTY pair. You should avoid specifying an arbitrarily high value for MAXPTYs. However, because each interactive user may have more than one session, we recommend that you allow 4 pseudo-TTY pairs for each user (MAXUIDS * 4). The MAXPTYs value influences the number of pseudo-TTY pairs that can be defined in the file system.

MAXSHAREPAGES = max sharepages

Specifies the maximum number of shared storage pages that can be concurrently in use by z/OS UNIX System Services functions. This can be used to control the amount of ESQA consumed, since the shared storage pages cause the consumption of ESQA storage.

The range is from 0 to 32 768 000. The default is 131 072 pages.

Notes:

1. You can set a denomination (or multiplier) value when defining the MAXSHAREPAGES value. The suffix, "C" can have a 1-character value as presented in Table 4-32 on page 4-428, but must not exceed the parameter-specific upper limit. MVS retains the denomination value and uses it within a subsequent D OMVS command
2. Use care when you adjust MAXSHAREPAGES on an active system. Dynamically decreasing the number of pages available to EQSA while there is a workload can cause errors, because the EQSA limit can be suddenly reached when the MAXSHAREPAGES limit is no longer as large. As a result, shared programs are not able to be loaded, and new forks are not able to be created. This situation can exist until the workload adjusts to the new lower limit.

MAXTHREADS = maxthreads

Specifies the maximum number of pthread_created threads, including those running, queued, and exited but not detached, that a single process can have currently active. Specifying a value of 0 prevents applications from using pthread_create. The range is 0 to 100 000; the default and the value in BPXPRMXX is 200.

MAXTHREADTASKS = maxthreadtasks

Specifies the maximum number of MVS tasks created with pthread_create (BPX1PTC) that a single user may have concurrently active in a process. The range is 1 to 32 768; the default and the value in BPXPRMXX is 50.

MAXTHREADTASKS lets you limit the amount of system resources available to a single user process.

- The minimum value of 1 prevents a process from performing any pthread_creates.
- A high MAXTHREADTASKS value may affect storage and performance. Each task requires additional storage for:
 - The control blocks built by the z/OS UNIX kernel
 - The control blocks and data areas required by the runtime library
 - System control blocks such as the TCB and RB

Individual processes can alter these limits dynamically.

MAXUIDS = maxuids

Specifies the maximum number of unique OMVS user IDs (UIDs) that can use z/OS UNIX System Services at the same time. The UIDs are for interactive users or for programs that requested z/OS UNIX System Services. The range is 1 to 32 767; the default and the value in BPXPRMXX is 200.

MAXUIDS lets you limit the number of active UIDs. Select a MAXUIDS by considering:

- Each z/OS UNIX System Services user is likely to run with 3 or more concurrent processes. Therefore, z/OS UNIX System Services users require more system resources than typical TSO/E users.
- If the MAXUIDS value is too high relative to the MAXPROCSYS value, too many users can invoke the shell. All users may be affected, because forks may begin to fail.

For example, if your installation can support 400 concurrent processes — MAXPROCSYS(400) — and each UID needs an average of 4 processes, then the system can support 100 users. For this operating system, specify MAXUIDS(100).

In assigning a value to MAXUIDS, consider if the security administrator assigned the same OMVS UID to more than one TSO/E user ID.

MEMLIMIT = maxmemlimit

Specifies the maximum amount (*maxmemlimit*) of allocated, non-shared, 1-megabyte storage segments above the bar allowed for the address space. Both the hard and soft RLMIT_MEMLIMIT values are set to this value, and the address space *memlimit* is modified to reflect his value.

Note: You can set a denomination (or multiplier) value when defining this value (nnnnnnnnC), where nnnnnnnn ranges from 1M — 16383P (noting values are rounded up) and C can have a 1-character value as presented in Table 4-32 on page 4-428. Also, be aware that SMF set override limits to the values you set here.

SETOMVS Command

MVS retains the denomination value and uses it within a subsequent D OMVS command

PID=pid,processlimitname=value

Dynamically changes a process-level limit for the process represented by *pid*.

PRIORITYGOAL = (n) | NONE

Specify from 1 to 40 service classes. These classes can be from 1 to 8 characters. If you do not specify this statement, or if you specify NONE, no array is created for it. All service classes specified on the PRIORITYGOAL option must also be specified in your workload manager service policy.

Generally, we do not recommend that you set PRIORITYGOAL.

RESET = (xx)

Specifies the parmlib member containing parameters to apply immediately to the running z/OS UNIX System Services environment. The variable specifies the character suffix of the BPXPRMxx member to use to change the environment. It can be any properly constructed BPXPRMxx member. This parameter accepts only the single keyword and parmfile specification. It does not accept additional keywords separated by commas.

The SETOMVS RESET command is similar to the SET OMVS command. The following table shows the acceptable parameters for each.

Notes:

1. **SETOMVS RESET** accepts only a single parameter; **SET OMVS** accepts more than one parameter.
2. **SETOMVS RESET=(xx)** has been changed to allow **SETOMVS RESET=xx** as well as **SETOMVS RESET=(xx)**. The parentheses are now optional.
3. **SETOMVS RESET=(xx)** redefines a PFS; it does not start it. Do not use **SETOMVS RESET=(xx)** unless absolutely necessary.

For more detailed information about the RESET parameter see *z/OS UNIX System Services Planning*.

Table 4-33. Acceptable Parameter Statements and Their Applicability

Parameter Statement	SET OMVS= (xx, yy, ...)	SETOMVS RESET= (xx)
AUTOMOVE	No	No
CTRACE	No	No
FILESYS	No	No
FILESYSTEM	No	No
FILESYSTYPE	Yes	Yes
FORKCOPY	Yes	Yes
FROMSYS	No	No
IPCMMSGNIDS	Yes	Yes
IPCMMSGQBYTES	Yes	Yes
IPCMMSGQNUM	Yes	Yes
IPCSEMNIDS	Yes	Yes
IPCSEMNOPS	Yes	Yes
IPCSEMNSEMS	Yes	Yes
IPCSHMMMPAGES	Yes	Yes
IPCSHMNIDS	Yes	Yes

Table 4-33. Acceptable Parameter Statements and Their Applicability (continued)

Parameter Statement	SET OMVS= (xx, yy, ...)	SETOMVS RESET= (xx)
IPCSHMNSEGS	Yes	Yes
IPCSHMPAGES	Yes	Yes
MAXASSIZE	Yes	Yes
MAXCORESIZE	Yes	Yes
MAXCPUTIME	Yes	Yes
MAXFILEPROC	Yes	Yes
MAXFILESIZE	Yes	Yes
MAXMMAPAREA	Yes	Yes
MAXPROCSYS	Yes	Yes
MAXPROCUSER	Yes	Yes
MAXPTYs	Yes	Yes
MAXSHAREPAGES	Yes	Yes
MAXTHREADS	Yes	Yes
MAXTHREADTASKS	Yes	Yes
MAXUIDS	Yes	Yes
MEMLIMIT	Yes	Yes
MOUNT	Yes	Yes
MOUNTPOINT	No	No
NETWORK	Yes	Yes
PRIORITYGOAL	Yes	Yes
PRIORITYPG	Yes	Yes
ROOT	Yes	Yes
RUNOPTS	No	No
SHRLIBRGNSIZE	YES	YES
SHRLIBMAXPAGES	YES	YES
STARTUP_EXEC	No	No
STARTUP_PROC	No	No
STEPLIBLIST	Yes	Yes
SUBFILESYSTYPE	Yes	Yes
SUPERUSER	Yes	Yes
SYSCALL_COUNTS	Yes	Yes
SYSNAME	No	No
SYSPLEX	No	No
TTYGROUP	Yes	Yes
USERIDALIASTABLE	Yes	Yes
VERSION	Yes	Yes

SERV_LPALIB=('dsname','volser')

Specifies the target service library where the UNIX System Services modules that are to be built into LPA are located. *dsname* is a 1-to-44 character value representing a valid MVS load library data set name. The alphabetic characters

SETOMVS Command

in the load library name must be uppercase. *volser* is a 1-to-6 character value representing a valid volume serial number for the volume that contains the specified MVS load library. The alphabetic characters in the volume serial number must be uppercase. The library data set must reside on the volume specified or the specification will not be accepted and an error will occur during the parsing of the statements.

SERV_LINKLIB=(*dsname*,*volser*)

Specifies the target service library where the UNIX System Services modules that are to be loaded into the private area of the OMVS address space are located. *dsname* is a 1-to-44 character value representing a valid MVS load library data set name. The alphabetic characters in the load library name must be uppercase. *volser* is a 1-to-6 character value representing a valid volume serial number for the volume that contains the specified MVS load library. The alphabetic characters in the volume serial number must be uppercase. The library data set must reside on the volume specified or the specification will not be accepted and an error will occur during the parsing of the statements.

SHRLIBRGNsize=shrlibrgnsize

Specifies the maximum size of the shared library region for address spaces that load system shared library modules.

SHRMAXPAGES=shrlibmaxpages

Specifies the maximum number of pages that can be allocated in the system to contain user shared library modules.

STEPLIBLIST = 'stepliblist'

Specifies the path name of a z/OS UNIX file. This file is intended to contain a list of data sets that are sanctioned by the installation for use as step libraries during the running of set-user-ID and set-group-ID executable programs.

SUPERUSER = superuser

This statement specifies a superuser name. You can specify a 1-to-8-character name that conforms to restrictions for a z/OS UNIX user ID. The user ID specified on SUPERUSER must be defined to the security product and should have a UID of 0 assigned to it. The user ID specified with **setuid()** is used when a daemon switches to an unknown identity with a UID of 0.

The default is SUPERUSER(BPXROOT).

SYNTAXCHECK=(*xx*)

Specifies that the operator wishes to check the syntax of the designated parmlib member. For example, to check the syntax of BPXPRMZ1 the operator enters:

SETOMVS SYNTAXCHECK=(Z1)

The system returns a message indicating either that the syntax is correct or that syntax errors were found and written into the hard copy log. This command parses the parmlib member in the same manner, and with the same messages as during IPL.

Note: **SYNTAXCHECK** checks syntax as well as the existence of HFS and zFS data sets specified in the catalog. Mount points are not verified.

SYSCALL_COUNTS = (YES | NO)

Specifies whether to accumulate syscall counts so that the RMF data gatherer can record this information. The default is NO.

If you specify YES, the path length for the most frequently used kernel system calls increases by more than 150 instructions.

SYSNAME=sysname*

sysname is the 1-8 alphanumeric name of a system participating in shared file system. This system must be IPLed with SYSPLEX(YES). **sysname** specifies the particular system on which a mount should be performed. This system will then become the owner of the file system mounted. If *(asterisk) is specified, it represents any other randomly selected system taking part in shared file system. The asterisk specification is not available with the **FROMSYS** parameter.

For examples of the use of this parameter when making move or change requests, see "shared file system in a Sysplex" in *z/OS UNIX System Services Planning*.

TTYGROUP = ttygroup

This specifies a 1-to-8-character name that must conform to the restrictions for a z/OS UNIX group name. Slave pseudoterminals (ptys) and OCS rty's are given this group name when they are first opened. This group name should be defined to the security product and have a unique GID. No users should be connected to this group.

The name is used by certain **setgid()** programs, such as **talk** and **write**, when attempting to write to another user's pty or rty.

The default is TTYGROUP(TTY).

USERIDALIASTABLE = 'useridaliastable'

Enables installations to associate alias names with MVS user IDs and group names. If specified, the alias names are used in z/OS UNIX System Services processing for the user IDs and group names listed in the table.

Specifying USERIDALIASTABLE causes performance to degrade slightly. The more names that you define, the greater the performance degradation. Installations are encouraged to continue using uppercase-only user IDs and group names.

The USERIDALIASTABLE statement specifies the pathname of a z/OS UNIX file. This file is intended to contain a list of MVS user IDs and group names with their associated alias names.

VERSION = 'nnnn'

The VERSION statement applies only to systems that are exploiting shared file systems. VERSION allows multiple releases and service levels of the binaries to coexist and participate in shared file systems. A directory with the value *nnnn* specified on VERSION is dynamically created at system initialization under the sysplex root that is used as a mount point for the version file system. This directory, however, is only dynamically created if the root file system for the sysplex is mounted read/write.

Note: *nnnn* is a case-sensitive character string no greater than 8 characters in length. It indicates a specific instance of the version file system. The most appropriate values for *nnnn* are the name of the target zone, &SYSR1, or another qualifier meaningful to the system programmer. For example, if the system is at V2R9, you can specify REL9 for VERSION. When SYSPLEX(YES) is specified, you must also specify the VERSION parameter.

The VERSION value is substituted in the content of symbolic links that contain \$VERSION. For scenarios describing the use of the version file system, see "Shared file system in a sysplex" in *z/OS UNIX System Services Planning*.

SETOMVS Command

When testing or changing to a new Maintenance Level (PTF), you can change the VERSION value dynamically by using the SETOMVS command:

```
SETOMVS VERSION='string'
```

You can also change the settings of this parameter via SET OMVS=(xx) and SETOMVS RESET=(xx) parmlib specifications.

Note: We do not recommend changing version dynamically if you have any users logged on or running applications; replacing the system files for these users may be disruptive.

SETPROG Command

Use the SETPROG command for:

- Updating the APF List (SETPROG APF)
 - Change the format of the authorized program facility (APF) list from static to dynamic, or dynamic to static
 - Add a library to a dynamic APF list
 - Delete a library from a dynamic APF list.
- Updating Dynamic Exits (SETPROG EXIT)
 - Add an exit routine to an exit
 - Change the state of an exit routine
 - Delete an exit routine from an exit
 - Undefine an implicitly-defined exit
 - Change the attributes of an exit.
- Updating LNKLST Concatenations (SETPROG LNKLST)
 - Define a LNKLST set of data sets for the LNKLST concatenation
 - Add data sets to or delete data sets from the LNKLST set
 - Remove the definition of a LNKLST set from the system
 - Test for the location of a specific module in the LNKLST concatenation
 - Activate a LNKLST set as the LNKLST concatenation for the system
 - Update an address space for jobs to use a LNKLST set.
- Managing Dynamic LPA Content (SETPROG LPA)
 - Specify modules to add to the LPA after IPL
 - Specify modules to delete from the LPA after IPL
 - Specify the minimum amount of CSA storage that must remain available after an ADD operation.

Each subcommand of SETPROG (for example, SETPROG LNKLST,DEFINE) has a direct analog in the PROGxx parmlib member (such as the LNKLST DEFINE statement) activated by SET PROG=xx. You will get the same result whether you activate the function by SETPROG or by SET PROG=xx.

Syntax

The syntax for each variation of the SETPROG command is shown immediately preceding its respective parameter list.

SETPROG

Notes:

1. The SETPROG command does not have an abbreviation.
2. The SETPROG command requires a /* */ around comments. Refer to “System Command Formats” on page 4-13 for further information.

Updating the APF List

Use the SETPROG APF command to:

- Change the format of the authorized program facility (APF) list from static to dynamic, or static to dynamic
- Add a library to a dynamic APF list
- Delete a library from a dynamic APF list.

SETPROG Command

To use the SETPROG APF command to update the contents of the APF list during normal processing, the format of the APF list must be dynamic. SETPROG is a system control command and is issued from a console with system (AUTH(SYS)) or higher authority.

- To add or delete the APF list entry for library *libname*, you must have UPDATE authority to the RACF® FACILITY resource class entity CSVAPF.*libname*, or there must be no FACILITY class profile that protects that entity.
- To change the format of the APF list to dynamic, you must have UPDATE authority to the RACF FACILITY resource class profile CSVAPF.MVS.SETPROG.FORMAT.DYNAMIC, or there must be no FACILITY class profile that protects that entity.
- To change the format of the APF list back to static, you must have UPDATE authority to the RACF FACILITY resource class profile CSVAPF.MVS.SETPROG.FORMAT STATIC, or there must be no FACILITY class profile that protects that entity.

If you authorize users to update the APF list using some other method, you must ensure that there is no FACILITY class profile that matches a profile listed above. If there is such a profile, the system uses it to determine if the requestor is authorized.

You can also use the SET PROG=xx command to update the APF list using parameters specified in the PROGxx parmlib member. See “SET Command” on page 4-382 for more information about using SET PROG=xx. See *z/OS MVS Planning: Operations* for information about defining RACF profiles for the SETPROG and SET PROG=xx commands.

```
SETPROG APF{,FORMAT={DYNAMIC|STATIC}}  
{,{ADD|DELETE},DSNAME|LIBRARY=libname,{SMS|VOLUME=volume} }
```

Notes:

1. You can specify the DSNAME parameter as DSN, LIB, or LIBRARY, the VOLUME parameter as VOL, and the DYNAMIC parameter as DYN.
2. This command requires a /* */ around comments. Refer to “System Command Formats” on page 4-13 for further information.

FORMAT=[DYNAMIC or STATIC]

Indicates that the format of the APF list is to change (from static to dynamic, or vice versa). If the system processes FORMAT=DYNAMIC successfully, authorized users can update the dynamic APF list during normal processing.

Before you change the format of the APF list to dynamic, contact the system programmer to validate that all programs and vendor products are converted to use dynamic APF services and that the proper program products are installed. Also, see the restrictions associated with changing the format of the APF list in *z/OS MVS Initialization and Tuning Reference*.

ADD

Adds the library specified on the DSNAME parameter to the APF list. There is no restriction on the number of libraries you can specify in a dynamic APF list. You can only use this option if the format of the APF list is dynamic.

DELETE

Deletes the library specified on the DSNAME parameter from the APF list. You can only use this option if the format of the APF list is dynamic.

DSNAMELIBRARY=*libname*

The 1-44 character name of the library that you want to add or delete. DSNAME can be an alias for the library name.

This function does not map an alias to the actual library name. Therefore, if you specify an alias, only the alias is added to, or deleted from, the APF list. Similarly, when you specify an actual library name as input, none of the library's aliases are added to, or deleted from, the APF list.

VOLUME=*volume*

The volume identifier for the volume containing the library specified on the DSNAME parameter, which is one of the following:

- The volume serial number
- Six asterisks (*****), indicating that the system is to use the volume serial number of the current system residence (SYSRES) volume.
- *MCAT*, indicating that the system is to use the volume serial number of the volume containing the master catalog.

SMS

Indicates that the library specified on the DSNAME parameter is managed by the storage management subsystem (SMS), and therefore no volume is associated with the library. When you display the APF list entry for an SMS-managed library, the volume appears as ***SMS***.

Example 1

To change the format of the APF list from static to dynamic, enter:

```
SETPROG APF,FORMAT=DYNAMIC
```

Example 2

To add library SYS1.ACCTG.DATA, on the current SYSRES volume, to the APF list, enter:

```
SETPROG APF,ADD,DSNAME=SYS1.ACCTG.DATA,VOLUME=*****
```

Example 3

To add SMS-managed library SYS1.DSSET.LOG to the APF list, enter:

```
SETPROG APF,ADD,DSNAME=SYS1.DSSET.LOG,SMS
```

Example 4

To delete library SYS1.ACCTG.DATA, on volume 617680, from the APF list, enter:

```
SETPROG APF,DELETE,DSNAME=SYS1.ACCTG.DATA,VOLUME=617680
```

Updating Dynamic Exits

Use the SETPROG EXIT command to:

- Add an exit routine to an exit
- Change the state of an exit routine
- Delete an exit routine from an exit
- Undefine an implicitly-defined exit
- Change the attributes of an exit.

SETPROG Command

You can use the SETPROG EXIT command to control exits that have been defined to the dynamic exits facility. Dynamic exits services are implemented by:

- The EXIT statement of the PROGxx parmlib member. The EXIT statement of PROGxx allows an installation to add exit routines to an exit, delete an exit routine for an exit, change the state of an exit routine, change the attributes of an exit, and undefine an implicitly defined exit.

The PROGxx EXIT statement interacts with the PROG=xx parameter of IEASYSxx and the SET PROG=xx command. At IPL, you can use PROG=xx to specify the particular PROGxx parmlib member the system is to use. During normal processing, you can use the SET PROG=xx command to set a current PROGxx parmlib member. See *z/OS MVS Initialization and Tuning Reference* for information about the PROGxx parmlib member.

- The SETPROG EXIT command.
- The CSVDYNEX macro. The CSVDYNEX macro can be used to define exits to the dynamic exits facility, control their use within a program, and associate one or more exit routines with those exits. It can also be used to associate exit routines with the existing SMF and allocation exits, which have been defined to the dynamic exits facility.

An installation can use any of these methods to control dynamic exits. An exit routine, for example, can be associated with an exit using the CSVDYNEX ADD request, the SETPROG EXIT,ADD operator command, or the EXIT statement of PROGxx.

The complete syntax for the SETPROG EXIT command is:

```
SETPROG EXIT,{ADD,EXITNAME=exitname,MODNAME=modname      }
[,STATE={ACTIVE|INACTIVE}]
[,DSNAME=dsname]
[,JOBNAME={jobname|*}]
[,ABENDNUM=(n[,CONSEC])]
[,FIRST|LAST]

{ATTRIB,EXITNAME=exitname,KEEPRC=(compare,kk) }

{DELETE,EXITNAME=exitname,MODNAME=modname      }
[,FORCE={YES|NO}]

{MODIFY,EXITNAME=exitname,MODNAME=modname      }
[,STATE={ACTIVE|INACTIVE}]
[,JOBNAME={jobname|*}]

{UNDEFINE,EXITNAME=exitname      }
```

Notes:

1. This command requires a /* */ around comments. Refer to "System Command Formats" on page 4-13 for further information.
2. The EXITNAME parameter must be the first parameter following the ADD, ATTRIB, DELETE, MODIFY or UNDEFINE keywords.

Note:

ADD

Adds an exit routine to an exit.

DELETE

Deletes an exit routine from an exit.

MODIFY

Changes the state of an exit routine.

UNDEFINE

Undefines an implicitly-defined exit. An exit is implicitly defined when:

- You add exit routines to an exit before the exit is defined
- You set attributes using the ATTRIB parameter before defining the exit.

ATTRIB

Changes the attributes of an exit.

EXITNAME= or EX= or EN=exitname

The 1-16 character name of the exit.

MODNAME= or MOD=modname

The 1-8 character name of the exit routine. If DSNAME is not specified, the system tries to locate the exit routine using the LPA, the LNKLST concatenation, and the nucleus.

DSNAME= or DSN= dsname

The 1-44 character data set name of a load library in which the named exit routine resides. The data set must be cataloged, but does not need to be APF-authorized.

If the data set has been migrated, processing of the SETPROG command is delayed until the data set has been retrieved.

JOBNAME={jobname or *}

The 1-8 character name of the job(s) for which this exit routine is to get control. If some other job calls the exit, this exit routine does not get control.

You can use the JOBNAME parameter to limit most exit routines to processing a particular job. However, you cannot use this parameter to restrict processing of the IEFUJV exit routine to a particular job.

To indicate more than one job name, use an asterisk as the last character. A matching jobname is one that matches all characters preceding the asterisk.

Specify JOBNAME=* to request that the system *not* check for the jobname. The default for the ADD parameter is *. The default for the MODIFY parameter is to leave the jobname unchanged.

STATE

Indicates the state of the exit routine. ACTIVE indicates that the exit routine is to be given control when the exit is called. INACTIVE indicates that the exit routine is not to be given control when the exit is called.

The default for the ADD parameter is ACTIVE. The default for the MODIFY parameter is to leave the state unchanged.

ABENDNUM=n[,CONSEC]

Indicates when the system should stop giving control to the exit routine in case of abends. ABENDNUM=n indicates that the exit routine is not to be given control after the *n*th abend. ABENDNUM=n, CONSEC indicates that there must be *n* consecutive abends before the system stops giving control to the exit routine. CONSEC is not supported if this exit has FASTPATH processing in effect, and either a PSW key 8 to 15 or ANYKEY processing in effect.

SETPROG Command

The default is to use the ABENDNUM characteristics that were specified (or defaulted) when the exit was defined. The ABENDNUM value must not exceed 8 decimal digits.

FIRST

Specifies that the system is to call the exit routine before all other exit routines associated with this exit, unless another exit routine, added after it, also specifies FIRST.

If you specify neither the FIRST nor the LAST parameter, the system may call the exit routines associated with this exit in any order.

LAST

Specifies that the system is to call the exit routine after all other exit routines associated with this exit, unless other exit routines are added after it.

If you specify neither the FIRST nor the LAST parameter, the system may call the exit routines associated with this exit in any order.

FORCE=YES or NO

Indicates that the system is to delete the exit routine. The exit routine will no longer be given control. Specify FORCE=YES for an exit with FASTPATH processing in effect, and either a PSW key 8 to 15 or ANYKEY processing in effect. Assuming the exit has FASTPATH processing in effect, and the PSW key is 8 to 15, or ANYKEY processing is in effect:

- FORCE=NO, the default, changes the state of the exit routine to inactive. The system does not free the storage.
- FORCE=YES frees the storage of the exit routine immediately. Use FORCE=YES only if you are sure that no exit is running that exit routine.

For exits that are non-FASTPATH or whose PSW key is 0 to 7, and are not ANYKEY, the system frees the storage when it determines that no other exits are using the exit routine.

KEEPRC=(*compare*,*kk*)

Specifies a comparison and a return code which, if true, cause the information produced by this exit routine to be returned to the exit caller. The valid choices for *compare* are EQ, NE, GT, LT, GE, and LE. For example, with *KEEPRC=(NE,4)*, if the exit routine produces a return code of 8, the compare for not-equal with 4 is true, and KEEPRC processing causes the information produced by this exit routine to be returned to the exit caller.

The default is not to perform KEEPRC processing. Do not enter more than 8 decimal digits when specifying a value for *kk*.

If return codes from more than one exit routine match the conditions specified, the system returns information from the exit routine that finished first.

Example

Associate exit routine MYMOD with the SMF exit known as SYS.IEFUJI, defined through the SYS statement in a SMFPRMxx parmlib member. The load module is in data set MY.DSN.

```
SETPROG EXIT,ADD,EXITNAME=SYS.IEFUJI,MODNAME=MYMOD,DSNAME=MY.DSN,  
STATE=ACTIVE
```

Updating LNKLST Concatenations

Use the SETPROG LNKLST command to:

- Define a LNKLST set of data sets for the LNKLST concatenation

- Add data sets to or delete data sets from the LNKLST set
- Remove the definition of a LNKLST set from the system
- Test for the location of a specific module in the LNKLST concatenation
- Activate a LNKLST set as the LNKLST concatenation for the system
- Update an address space for jobs to use a LNKLST set

PROGxx is the parmlib member used to define one or more LNKLST sets. You can use PROGxx to activate one of the LNKLST sets as the LNKLST concatenation at IPL. (You can also activate the LNKLST concatenation through LNKLSTxx, but IBM recommends that you use PROGxx.) SETPROG LNKLST allows you to modify the LNKLST concatenation dynamically after IPL. See *z/OS MVS Initialization and Tuning Reference* for information about the PROGxx parmlib member.

The complete syntax for the SETPROG LNKLST command is:

```
SETPROG LNKLST,{DEFINE,NAME=lnklstname[,COPYFROM=lnklstname][,NOCHECK] }
               {ADD,NAME=lnklstname,
                DSNAME=dsname[,VOLUME=volser][,ATBOTTOM      ]
                           [,ATTOP          ]
                           [,AFTER=dsname]
                           [,CONCAT(CHECK | NOCHECK)]           }
               {DELETE,NAME=lnklstname,DSNAME=dsname        }
               {UNDEFINE,NAME=lnklstname                   }
               {TEST,NAME=lnklstname,MODNAME=name         }
               {ACTIVATE,NAME=lnklstname                  }
               {UPDATE,{JOB=jobname} [,DELAY=nn]
                {ASID=asid } [DELAY=delay]             }
               {UNALLOCATE                                }
               {ALLOCATE                                  }
```

Note: This command requires a /* */ around comments. Refer to “System Command Formats” on page 4-13 for further information.

DEFINE

Specifies that you want to define a LNKLST set (a set of ordered data sets for the LNKLST concatenation).

NAME=/lnklstname

The name of the LNKLST set that you want to specify. Naming conventions are as follows:

- You can specify from 1 to 16 characters for name.
- You can use alphanumerics, underscores, periods, and \$, #, or @.
- Do not use imbedded blanks.
- Do not use the name CURRENT. The system uses CURRENT to mean the current LNKLST set.
- For all options except TEST, do not use the name IPL. The system uses IPL to mean LNKLST information specified in parmlib member LNKLSTxx. However, you can specify

SETPROG LNKLST,TEST,NAME=IPL

- Do not begin the name with SYS. SYS is reserved for IBM use.

COPYFROM=/lnklstname

Specifies the name of an existing LNKLST set from which to initialize the LNKLST set you are defining. If you specify CURRENT for the name, the system uses the current LNKLST set.

SETPROG Command

NOCHECK

Indicates that the system does not check to determine if the specified LNKLST set contains SYS1.LINKLIB, SYS1.MIGLIB, and SYS1.CSSLIB before allocating the LNKLST concatenation.

Note: Use NOCHECK with caution. You might use NOCHECK after you have modified SYS1.LINKLIB and want to compress SYS1.LINKLIB. For a procedure, see the description of the PROGxx NOCHECK parameter in *z/OS MVS Initialization and Tuning Reference*.

ADD

Indicates that you want to add a data set to the specified LNKLST set.

You cannot add a data set to either the current or the active LNKLST set. If a data set has been migrated, the request waits until the data set is available. For information about the maximum number of data sets you can define to a LNKLST set, see *z/OS MVS Initialization and Tuning Reference*.

Note that you cannot specify in a SETSSI ADD command a subsystem initialization routine that is added via a SETPROG LNKLST,ADD command. That is because the new LNKLST library will not be picked up until the end of the job that is running. However, the SETSSI command runs in the MASTER ASID, which never ends (until the next IPL). Therefore, the SETSSI command can never pick up a new LNKLST. To correct this problem, you must issue the UPDATE command; however, use caution when you do that. See the “UPDATE option” on page 4-450 for considerations and restrictions.

DSNAME=dsname

The 44-character name of a data set or library that you want to add to the specified LNKLST set or delete from the specified LNKLST set. DSN, LIB, and LIBRARY are accepted synonyms for this parameter.

The data set can be a PDS or a PDSE. IBM recommends that you use PDSEs because of the limitations on the number of extents for a LNKLST concatenation. See *z/OS MVS Initialization and Tuning Reference*.

Data sets to be added can be SMS-managed or non SMS-managed. After the system determines the volume and the SMS status of the data set, the following actions result in an error when the system tries to allocate the LNKLST set:

- If the data set in the LNKLST set changes status from SMS-managed to non SMS-managed, or from non-SMS managed to SMS-managed.
- If a non SMS-managed data set in the LNKLST set is deleted and moved to another volume.

In either case, to add the data set after the change has occurred, you must first delete the data set from the LNKLST set and add it again.

VOLUME=volsr

Specifies the name of the volume on which the data set resides. The data set must be cataloged. If the volume does not match the name in the catalog, the ADD request fails. The name can be from 1 to 6 characters.

When a data set is cataloged in a user catalog instead of the master catalog, you can use this parameter. If the data set is cataloged in a user catalog, but not in the master catalog, you must specify the VOLSER of the volume on which the data set resides.

ATBOTTOM

ATTOP

AFTER=dsname

Indicates where in the LNKLST set you want to place the data set. The default is ATBOTTOM.

ATBOTTOM indicates that you want to place the data set specified on the DSNAME parameter at the bottom of the list of data sets in the LNKLST set.

ATTOP indicates that you want to place the data set specified on the DSNAME parameter at the beginning of the LNKLST set. The system places the LINKLIB, MIGLIB, and CSSLIB data sets in that order at the beginning of every LNKLST set in the LNKLST concatenation. If you use ATTOP, the system always places the data set after the CSSLIB data set.

AFTER =dsname indicates that the system places the data set specified on the DSNAME parameter after the data set specified by *dsname*. You cannot use this parameter to place a data set after the LINKLIB, MIGLIB, or CSSLIB data set in the LNKLST set. Instead, use ATTOP if you want to place the data set immediately after the CSSLIB data set.

Default Value: If you omit ATBOTTOM, ATTOP, or AFTER, the system adds the data set to the bottom of the LNKLST set.

CONCAT(CHECK | NOCHECK)

Specifies whether or not to check if the concatenation defined by the LNKLST set is full. The parameter is optional. CONCAT(NOCHECK) is the default.

CONCAT(CHECK) specifies that the system *is* to check if the concatenation is full. This requires that all data sets in the LNKLST be allocated and concatenated together, and will require more processing time than the default.

CONCAT(NOCHECK), the default option, specifies that the system *is not* to check whether the concatenation is full. (If the concatenation actually is full, it will be detected when the LNKLST set is activated.)

DELETE

Indicates that you want to delete a data set from the specified LNKLST set.

You cannot delete a data set from either the current or the active LNKLST set.

UNDEFINE

Removes the definition of the LNKLST set specified by NAME=*lnklstname* from the system. You cannot remove the definition of the current LNKLST set, another LNKLST set that is being actively used by a job or address space, or the LNKLST defined at IPL through LNKLSTxx and the LNK parameter of IEASYSxx. See "Removing or Compressing a Data Set in an Active LNKLST Set" in *z/OS MVS Initialization and Tuning Reference* for information about LLA management of the LNKLST data set.

TEST

Indicates that you want to locate a specific routine associated with a data set in the LNKLST set. If the system locates the data set, the system indicates the name of the data set. If a data set has been migrated, the request waits until the data set is available.

MODNAME=name

MODNAME specifies the name of a module to be located in the LNKLST set. MODULE and MOD can be used as synonyms for MODNAME.

ACTIVATE

Indicates that you want to activate the specified LNKLST set as the current LNKLST concatenation. When you use SETPROG LNKLST to activate the LNKLST set after IPL, jobs or address spaces that are still active continue to

SETPROG Command

use the previous current LNKLST set. To associate a job in an address space to the current LNKLST set after IPL, see UPDATE. See "Removing or Compressing a Data Set in an Active LNKLST Set" in *z/OS MVS Initialization and Tuning Reference* for information about LLA management of the LNKLST data set.

If a data set in the LNKLST set has been migrated before the LNKLST set is activated, the request waits until the data set is available.

When the ACTIVATE request completes, the system issues an event (ENF) signal (event code 52). Depending on the options specified in SMFPRMxx, whenever a LNKLST set is activated, the system records SMF record type 90 subtype 29. See "SETSMF Command" on page 4-456.

UPDATE

Indicates that the system is to update an address space so that a specified job or jobs associated with that space can use the current LNKLST set. If the job is using another LNKLST set when the current LNKLST set is activated, it will continue to use the original LNKLST set until it completes operations. When the job completes and restarts, it then uses the data sets defined in the new currently active LNKLST set. See "Removing or Compressing a Data Set in an Active LNKLST Set" in *z/OS MVS Initialization and Tuning Reference* for information about LLA management of the LNKLST data set.

Be careful when you use UPDATE. Updating an address space while a program in that address space is fetching a module can cause the fetch to fail or to locate an incorrect copy of the module. The system does not attempt to verify the validity of the data for UPDATE.

JOB=jobname

Specifies the name of the job or jobs to update. You can use wildcard characters (?) or (*) for *jobname*. UPDATE updates any job whose name matches the specified criteria. The system compares *jobname* to the name of any initiated job or jobs that match, or to the name of the address space.

ASID=asid

Specifies the address space id for the job.

DELAY=nn

Indicates the number of seconds to delay the completion of the UPDATE operation.

UNALLOCATE

Indicates that you want to undo all existing allocations obtained while processing active LNKLST sets. This also releases the SYSDSN ENQ.

Notes:

1. Make sure that you do not delete or move the LNKLST data sets while the allocations are not in effect (from the time that you use the UNALLOCATE request until the time that you use the ALLOCATE request).
2. Once you have completed everything associated with the UNALLOCATE, you must specify LNKLST ALLOCATE to re-obtain the remaining ENQs.

ALLOCATE

Indicates that you want to re-obtain the allocation (and SYSDSN ENQ) for every data set in every active LNKLST.

Example 1

Add the data set DATA.SET.A to the LNKLST set MY.LNKLST.SET. The system places the data set after the CSSLIB data set in the LNKLST set.

```
SETPROG LNKLST,ADD,NAME=MY.LNKLST.SET,DSNAME=DATA.SET.A,ATTOP
```

Example 2

Change the job MY.JOB to use the current LNKLST set:

```
SETPROG LNKLST,UPDATE,JOB=MY.JOB
```

Managing Dynamic LPA Content

Use the SETPROG LPA command anytime after IPL to specify:

- Modules to add to the LPA
- Modules to delete from the LPA
- The minimum amount of CSA storage that must remain available after an ADD operation.

You can also initiate a change to the LPA from a program via the CSVDYLPA macro, or by an operator using the SET PROG command. You can use the PROG system parameter to specify CSA threshold values, but not to request ADD or DELETE operations.

You can exercise certain controls over the modules to be loaded:

- You specify a data set from which the system is to load the modules. You must be authorized to make the request.
- You can request that the modules be placed into fixed common storage.
- You can request that only the full pages within a load module be page-protected. This does leave the likelihood of the beginning and/or end of a load module not being page protected. By default, each module is individually page-protected. This is, however, wasteful of common storage, as each module needs then to occupy a whole number of 4096-byte pages. In all cases the module will be in key 0 storage.

The system considers LPA modules as coming from an authorized library. As part of its LPA search, the system will find modules that had been added dynamically. It will find a module that had been added dynamically before it finds one of the same name that had been added during IPL.

Use the SETPROG LPA command to replace modules only where the owning product verifies the replacement. Otherwise, replacement could result in partial updates. If the owning product has already saved the module address, the system will not conduct an LPA search and will not find the updated module. Also, the addresses of all modules that are accessed via a program call (PC) instruction are stored in the PC table. That table is not updated by the SETPROG LPA command. Therefore, these modules cannot be replaced using the SETPROG LPA command. You must IPL for the updates to take effect.

Note: Dynamic LPA does not update system control blocks, which includes the SVC table. To add or replace an SVC routine, the user of the SETPROG LPA command (or the owner of the SVC) must first write a program that gets control as the CSVDYLPA exit routine, looks for a given routine name, and issues the appropriate SVCUPDTE service call to update to the SVC table. For more information on the CSVDYLPA exit routine, see "Monitoring Dynamic LPA Processing" in *z/OS MVS Programming: Authorized Assembler Services Guide*, SA22-7608.

SETPROG Command

It is sometimes necessary to re-IPL the system to replace LPA modules. For example, many service updates of LPA modules will require a re-IPL.

The complete syntax for the SETPROG LPA command is:

```
SETPROG LPA, {ADD,[MODNAME=(modname...,modname) | MASK=mask]
                ,DSNAME=[dsname | LNLST]
                [,FIXED] [,PAGEPROTPAGE]

                {DELETE,MODNAME=(modname...,modname)
                 FORCE=YES [CURRENT | OLDEST]

                {CSAMIN=(below,above)
                 }
```

Notes:

1. You may alternatively express the parameters associated with the keywords MASK, DSNAME, and FORCE in the preceding syntax diagram using parentheses rather than equals signs, as shown in the descriptions below.
2. This command requires a /* */ around comments. Refer to "System Command Formats" on page 4-13 for further information.

LPA

Statement type indicating that an action may be performed on the LPA.

ADD

Specifies that one or more modules be added to the LPA.

Default Value: None

Attention: Modules added to the system by dynamic LPA processing are placed into CSA or ECSA storage. Therefore, it is important to ensure that the system CSA and ECSA sizes are adequately defined to handle the additional consumption of CSA storage resulting from the issuance of the dynamic LPA request. Further protection can be gained through the use of the CSAMIN parameter described below.

DELETE

Specifies that one or more modules be deleted from the LPA. Only modules added to the LPA after an IPL are eligible for dynamic deletion.

Default Value: None

CSAMIN

Specifies the minimum amount of CSA and ECSA that must remain after a module is added to the LPA. If the requested ADD operation would reduce the CSA or ECSA below the defined minimum, the system rejects the operation.

Default Value: (0,0)

below

The minimum amount of below-16M CSA storage that must remain after the ADD operation. The value can be expressed as n, nK, and nM.

above

The minimum amount of above-16M CSA storage that must remain after the ADD operation. The value can be expressed as n, nK, and nM.

MODNAME(modname,...,modname) | MODNAME=(modname,...,modname)

modname is the 1-8 character LPA module name or alias. If a modname has

aliases, you must specify the module name and all of its aliases. If the last character of the modname is an asterisk (*), it will be treated as X'C0' -- which lets you specify the name of a load module that ends with that nonprintable character. You cannot use wildcard characters within *modname*. You can submit up to 128 module names, and can use MOD or MODULE as synonyms of MODNAME.

Default Value: None

MASK(mask) | MASK=mask

mask is the 1-8 character mask that is to be applied to all the members of the specified data set. It can contain wildcard characters "*" and "?" and all members that match will be processed.

Default Value: None

DSNAME(dsname) | DSNAME=dsname

dsname is the 1-44 character data set name that contains the module(s) or alias(es). When MODNAME is specified, you can specify DSNAME(LNKLST) if you want the system to search the lnklst instead of a particular data set. The data set must be cataloged. It may be allocated as a PDS or PDSE program library.

The attribute of the CSA for each module is assigned as OWNER=SYSTEM. DSN, LIB, and LIBRARY can be used as synonyms of DSNAME.

Default Value: None

FIXED

Indicates that the modules are to be placed in fixed storage.

Default Value: If FIXED is not specified, the modules will be placed in pageable storage.

PAGEPROTPAGE

Indicates whether or not to page protect entire modules. You can use PPPAGE or PPP as synonyms of PAGEPROTPAGE. Protecting entire modules requires more storage than just that necessary to contain the modules, because each module gets allocated a number of whole pages.

PAGEPROTPAGE causes the system to protect only the full pages within each load module. This reduces the storage requirement, but makes it possible for a storage overlay of the beginning or end of the load module to occur.

Default Value: Page protect entire modules.

FORCE(YES) | FORCE=YES

Confirms that the delete requestor understands the ramifications of deleting a module from the LPA, when the system can have no knowledge of whether any code is currently executing within the specified module.

Default Value: None. Required parameter.

CURRENT | OLDEST

CURRENT specifies to delete the current copy. OLDEST specifies to delete the oldest dynamic copy. You can use CUR as a synonym of CURRENT and OLD as a synonym of OLDEST.

Default Value: CURRENT

SETRRS CANCEL Command

SETRRS CANCEL Command

Use the SETRRS CANCEL command to end resource recovery services (RRS) abnormally.

Use this command only at the direction of the system programmer. (Normally, you will not use this command because RRS should be running at all times; stopping RRS can cause application programs to abend or wait until RRS is restarted.)

While stopping, RRS abends incomplete commit and backout requests and passes return codes to the requesting application programs.

If SETRRS CANCEL does not work, the system programmer might suggest you use FORCE *jobname*,ARM, where *jobname* is that specified on the START command for RRS.

Note: You can also use the CANCEL command to stop RRS, but you cannot request a dump or specify any parameters except the jobname ATRRRS (or the name your installation has chosen).

Syntax

The complete syntax for the SETRRS CANCEL command is:

```
SETRRS CANCEL[,DUMP|NODUMP]
```

Note: The SETRRS command does not have an abbreviation.

Parameters

DUMP | NODUMP

Specify whether or not the system is to request an SVC dump of the RRS address space before RRS is stopped. If you omit the parameter, the default is NODUMP.

Example

To stop RRS from running and request an SVC dump, enter:

```
SETRRS CANCEL,DUMP
```

SETRRS SHUTDOWN Command

Issuing SETRRS SHUTDOWN provides a normal shutdown command to bring down RRS without resulting in a X'058' abend. In order to notify RRS resource managers that RRS is terminating, all the currently active resource managers will be unset. After the unset processing is completed, the RRS jobstep task and all of its subtasks will normally be terminated to clean up the address space. In addition to the RRS infrastructure tasks, there are also timed process tasks and server tasks running in the RRS address space. These tasks are also shut down normally as well.

Use this command only at the direction of the system programmer. (RRS should be running at all times; stopping RRS might cause application programs to abend or wait until RRS is restarted.)

Syncpoint processing for the outstanding work will be stopped by unsetting exits of resource manager. Resource manager will need to reset its exits and restart with RRS after RRS is restarted.

Unset resource managers:

As part of the RRS shutdown processing, it will queue any AtrRMUnset synchronous server request to perform the resource manager (RM) unset processing. The RM unset processing consists of the following:

1. Transitions the RM to UnsetInProgress state to prevent further RM requests.
2. Performs URI failed processing for all of the RM's interests.
3. Transitions RM to the unset state. RM will need to reset its exits with RRS when RRS comes back up.
4. Notifies the RM that its exits have been unset, and drives RM's notification exit with a specific ATR_RM_EXIT_UNSET reason code. A new reason code "Exit Manager Unavailable" will be returned to further explain why the RM's exits are unset.

If SETRRS SHUTDOWN does not work, the system programmer might suggest you use FORCE *jobname*,ARM, where *jobname* is that specified on the START command for RRS.

Note: You can also use the CANCEL RRS, SETRRS CANCEL and FORCE commands. Each has a different outcome. For more information on these commands, see *z/OS MVS System Commands*, SA22-7627.

Syntax

The complete syntax for the SETRRS SHUTDOWN command is:

```
SETRRS SHUTDOWN
```

Note: The SETRRS command does not have an abbreviation.

Parameters

There are no parameters for this command.

Example

To stop RRS from running, enter:

```
SETRRS SHUTDOWN
```

SETSMF Command

In contrast to the SET command, which allows an installation to specify a different SMFPRMxx parmlib member or restart SMF, the SETSMF command allows an installation to add a SUBPARM parameter or replace any previously-specified parameter in the active SMF parmlib member except the ACTIVE, PROMPT, SID, or EXITS parameters. The SETSMF command cannot add a parameter to the active SMF parmlib member. The SETSMF command cannot be used with a SMFPRMxx member that specified NOPROMPT. To avoid possible confusion with the SET SMF command, use the abbreviation SS for the SETSMF command.

Syntax

The complete syntax for the SETSMF command is:

```
SETSMF parameter(value[,value]...)
```

Parameters

The parameters are:

parameter

specifies any SMF parameter in the SMFPRMxx member except ACTIVE, PROMPT, SID, or EXITS.

value[,value]

specifies the new value for the specified parameter.

Notes:

1. More than one parameter can be changed as long as the length of the command does not exceed 124 characters.
2. Both the SUBSYS and SUBPARM specifications can be changed on the same SETSMF command as long as the subsystem name is the same.
3. SET SMF, SETSMF, and DISPLAY SMF commands cannot run simultaneously. One waits for the other to complete before starting.
4. The new values for STATUS or MAXDORM do not take effect until the old ones, if any, expire.

Example 1

To set the SMF parameters for started tasks so that only system records are collected and checkpoint accounting records are taken every 30 minutes, enter:

```
SETSMF SUBSYS(STC,TYPE(0:127),INTERVAL(003000))
```

Example 2

To set the SMF parameter NOMULCFUNC, which indicates that users of the IFAUSAGE service that registered specifying SCOPE=FUNCTION do not need to use IFAUSAGE with the REQUEST=FUNCTIONxxx parameters, enter:

```
SETSMF NOMULCFUNC
```

This allows a measured usage program using SCOPE=FUNCTION, such as DB2, to record only its registration data and omit recording the usage data.

SETSMS Command

Use the SETSMS command when the Storage Management Subsystem (SMS) is active (running) to change a subset of SMS parameters from the console without changing the active IGDSMSxx member of SYS1.PARMLIB.

Use the SETSMS command to:

- Activate a new SMS configuration by specifying an active configuration data set (ACDS), a source control data set (SCDS), or both data sets. This action affects all MVS systems in the SMS complex.
- Replace the active configuration data set (ACDS) by specifying an ACDS different from the one that SMS is currently using. This action affects all MVS systems in the SMS complex.
- Replace the communications data set. This action affects all MVS systems in the SMS complex.
- Save the active configuration in a data set.
- Change the synchronization interval (INTERVAL). This change applies only to the system on which you issue the SETSMS command.
- Change the interval (DINTERVAL) that SMS waits between reading device statistics from the 3990-3 control unit (applicable only if the 3990-3 is installed and has at least one SMS-controlled volume). This change applies only to the system on which you issue the SETSMS command.
- Change the number of seconds that the dss component of DFMSMS will wait during backup processing for quiesce data set requests to complete (DSSTIMEOUT).
- Change the interval that SMS waits between recording cache control unit summaries (applicable only if the 3990-3 is installed and has at least one SMS-controlled volume). This change applies only to the system on which you issue the SETSMS command.
- Change the interval that SMS waits between recording of BMF (buffer management facility) statistics. This change applies only to the system on which you issue the SETSMS command.
- Change the maximum number of times that the buffer management facility (BMF) least recently used (LRU) routine will pass over inactive buffers before making them available for reuse (LRUCYCLES).
- Change the number of seconds that the buffer management facility (BMF) will wait between calls to the BMF data space cache LRU (least recently used) routine (LRUTIME).
- Specify trace options for SMS. This change applies only to the system on which you issue the SETSMS command.
- Specify deadlock detection intervals.
- Change the SMF interval time for recording SMF type 42 records.
- Change the new log of logs name, used by DFSMStvs.
- Change the maximum number of unique lock requests that a single unit of recovery can make.
- Change the quiesce exit timeout value to specify the amount of time the DFSMStvs quiesce exits will allow to elapse before concluding that a quiesce cannot be completed successfully.
- Change the maximum time that a VSAM RLS or DFSMStvs request is to wait for a required lock before the request is assumed to be in deadlock.

SETSMS Command

- Change the virtual storage size that is used to cache PDSE directory buffers in the SMSPDSE1 restartable address space. The directory buffers reside in 64-bit addressable virtual memory. The PDSE1_DIRECTORY_STORAGE parameter allows a change of the PDSE1_DIRECTORY_STORAGE that is either defaulted at IPL, or specified in the IGDSMSxx parameter in SYS1.PARMLIB. You must restart the SMSPDSE1 address space in order for this change to become effective.
- Change the hiperspace storage size that is used for PDSE member caching in the SMSPDSE1 restartable address space. The PDSE1_HSP_SIZE parameter allows a change of the PDSE1_HSP_SIZE that is either defaulted at IPL or specified in the IGDSMSxx parameter in SYS1.PARMLIB. You must restart the SMSPDSE1 address space in order for this change to become effective.
- Change the buffer-beyond-close option that is used to manage the cache of PDSE directory and member buffers in the SMSPDSE1 restartable address space. The directory buffers reside in 64-bit addressable virtual memory and optionally member data resides in a hiperspace. The PDSE1_BUFFER_BEYOND_CLOSE parameter allows a change of the PDSE1_BUFFER_BEYOND_CLOSE that is either defaulted at IPL, or specified in the IGDSMSxx parameter in SYS1.PARMLIB. You must restart the SMSPDSE1 address space in order for this change to become effective.

For more information about the ACDS, SCDS, and COMMDS data sets, see *z/OS DFSMS Storage Administration Reference* and *z/OS DFSMSdfp Diagnosis*.

If you are not sure about the differences between SET SMS and SETSMS, see Table 4-34.

Table 4-34. Comparison of SET SMS with SETSMS

Difference	SET SMS Command	SETSMS Command
When and how to use the command	Initializes SMS parameters and starts SMS if it has been defined but not started at IPL time. Changes SMS parameters if SMS is already running.	Changes SMS parameters only when SMS is running.
Where the parameters are entered	In the IGDSMSxx member of SYS1.PARMLIB.	At the console.
What default values are available	Default values are used for unspecified parameters except for BLOCKTOKENSIZE. If you omit BLOCKTOKENSIZE, the previous value in effect remains in effect. In z/OS 1.7, the original value of BLOCKTOKENSIZE is REQUIRE and in z/OS 1.8 and later the original value of BLOCKTOKENSIZE is NORQUIRE.	No default values. Unspecified parameters remain unchanged.

Some combinations of SETSMS parameters are not valid. Table 4-35 shows these incorrect combinations.

Requirement: The SCDS is a required parameter if the ACDS does not contain a valid configuration.

Note: The SCDS is a required parameter if the ACDS does not contain a valid configuration.

Table 4-35. Incorrect Combinations of SETSMS Parameters

Parameter	ACDS	SCDS	COMMDS	SAVEACDS
ACDS	N/A		Not Valid	Not Valid
SCDS		N/A	Not Valid	Not Valid
COMMDS	Not Valid	Not Valid	N/A	Not Valid
SAVEACDS	Not Valid	Not Valid	Not Valid	N/A

Scope in a Sysplex

The SETSMS command has sysplex scope only if you are changing the SCDS, ACDS, or COMMDS, and only if all systems in the sysplex are in the same SMS complex. See “Using Commands That Have Sysplex Scope” on page 1-11 for an explanation of sysplex scope.

Other parameters that have sysplex scope are as follows:

- RLSABOVETHEBARMAXPOOLSIZE
- RLSFIXEDPOOLSIZE
- RLSTMOUT
- DEADLOCK

,

Syntax

The complete syntax for the SETSMS command is:

```
SETSMS parameter(value) [,parameter(value)] ...
```

Note: No imbedded blanks are allowed between parameters on this command.

Parameters

parameter(value) is one of the following:

Configuration Parameters

ACDS(*dsname*)

The *dsname* specifies a data set that has been defined as an active control data set. The information in the data set is copied into the SMS address space to create (or replace) the active configuration. This parameter affects all MVS systems in the SMS complex. The command format is:

SETSMS ACDS(*dsname*)

AKP({*nnn*[,{*nnn*[,...,*nnn*] } | 1000})

Specifies one or more activity keypoint trigger (AKP) values. Each AKP value (*nnn*) is the number of logging operations between the taking of keypoints. You

SETSMS Command

can specify up to 32 activity keypoint values. AKP values must be specified in the same order as DFMSMStvs instance names. Valid values are from 200 to 65535. The default is 1000.

The command format follows:

```
SETSMS AKP(1000)
```

COPYSCDS(*scds_dsn, acds_dsn*)

Specifies that SMS is to copy an SCDS (specified by *scds_dsn*) into an ACDS (specified by *acds_dsn*). Although typically the only way to create an ACDS is to activate an SCDS, you might not want to activate an SCDS for this purpose. For example, you might have a production system that creates and maintains an SCDS for a disaster recovery system, and you want the corresponding ACDS to be available during the IPL of the disaster recovery system. However, you do not want to activate the SCDS on the production system. Without using COPYSCDS, activating the SCDS was the only way to create the disaster recover ACDS. COPYSCDS provides an alternate way to create a copy of a SCDS as an ACDS without having to activate the SCDS. Do not specify COPYSCDS in the same command as ACDS, SCDS, SAVESCDS or SAVEACDS.

SCDS(*dsname*)

SMS is to use the specified source control data set (SCDS) to activate a new configuration. This parameter affects all MVS systems in the SMS complex.

If the ACDS is empty or does not contain a valid configuration (possibly because of a damaged data set), use the SCDS parameter to specify a source control data set. The command format is:

```
SETSMS ACDS(dsname),SCDS(dsname2)
```

If the ACDS is empty and no SCDS is specified, SMS issues a message that identifies the specified control data set as non-valid.

Note: Do not specify the SCDS parameter if the ACDS *does* contain a valid configuration; the SCDS will overlay the ACDS. To recover from such accidental loss, it is a good idea to maintain a backup copy of the current ACDS to be activated in case of damage to the main copy.

SAVEACDS(*dsname*)

Specifies the ACDS that SMS is to use to save to DASD the active configuration from the SMS address space. SAVEACDS is one way to create a backup ACDS. Do not specify SAVEACDS in the same command as ACDS, SCDS, or SAVESCDS.

SAVESCDS(*dsname*)

Specifies the SCDS that SMS is to use to save to DASD the active configuration from the SMS address space. SAVESCDS is one way to create a backup SCDS. Do not specify SAVESCDS in the same command as ACDS, SCDS, or SAVEACDS.

COMMDS(*dsname*)

SMS is to use the named data set as the new communications data set.

If the replacement COMMDS is empty, SMS primes it with information from the active configuration. If the data set is not empty, SMS determines which ACDS was used to prime the new data set. If the ACDS named on COMMDS is the same as the one that is active, processing continues with the new COMMDS. Otherwise, SMS prompts the operator (by message IGD076D) to decide whether SMS should use the ACDS named on COMMDS or continue to use the current ACDS.

Notes:

1. If SMS cannot re-access the previously active communications data set, the operator must issue the command to change the COMMDS on each MVS system in the SMS complex.
2. The COMMDS parameter is mutually exclusive with ACDS, SCDS, and SAVEACDS.

FAST_VOLSEL(ONIOFF)

Specifies whether to use the 'fast' approach during SMS volume selection.

If you specify ON, SMS first selects volumes typically until DADSM rejects 100 volumes for insufficient free space. SMS issues message IGD17294I to indicate that 'fast' volume selection has been entered and then excludes volumes that do not have sufficient free space in the volume statistics. This 'fast' approach can inadvertently exclude volumes that have sufficient free space but for which SMS volume statistics indicates that they do not. SMS volume statistics can occur for the following events:

- The VTOC index is broken.
- OEM products bypass CVAF processing.
- In an SMSplex when the SMS synchronization time interval has not yet been driven to update the SMS configuration with the most current space statistics. These statistics are based on updates that can occur on another system in the SMSplex

If you specify OFF, SMS uses the 'normal' approach to select volumes.

The default is OFF.

MAXLOCKS({maxl0},{incr0})

Specifies a pair of values in the range of 0 to 999999. The two values are the maximum number of unique lock requests that a single unit of recovery can make, and an increment value. Once the maximum number of unique lock requests is reached, warning messages are issued every time the number of unique lock requests over and above the maximum increases by a multiple of the increment. When the maximum number is reached, warning message IGW859I is issued to the system console, and message IGW10074I is issued to the job log. The messages include the name of the job that is holding the locks. This information will help you to determine whether the job should be canceled, in which case the unit of recovery will be backed out, and the locks will remain held until the backout completes. Specifying a value of 0 indicates that warning messages IGW859I and IGW10074I should not be issued.

This parameter applies across all systems.

Notes:

1. Lock requests are considered unique if they lock different records within the base cluster. Repeated requests for the same base cluster records will not result in the count being incremented.
2. Warning messages IGW859I and IGW10074I are not issued for units of recovery that are in backout. This is because a unit of recovery that is in backout cannot obtain locks on any additional records.
3. Messages IGW859I and IGW10074I are issued until the unit of recovery reaches commit. Once the unit of recovery reaches commit, no additional messages will be issued.
4. To avoid flooding the system console with messages, messages IGW859I and IGW10074I are issued by an asynchronous timer driven task that

SETSMS Command

wakes up every 10 seconds. This means that the messages will not necessarily reflect the exact values specified for the maximum and the increment, but rather will reflect the values which represent the state of the unit of recovery at the time the task awakens.

5. MAXLOCKS takes into account the number of unique lock requests. It does not count the actual number of locks obtained. The number of locks requested will differ from the number of locks held when alternate indexes are used. If an update modifies alternate keys, a lock is obtained for the base record, for each old alternate key, and for each new alternate key. Therefore, if n alternate keys are modified, a single lock request can result in obtaining $(2n+1)$ locks.

Some examples of how this parameter can be specified are:

MAXLOCKS(0,0)

Valid - messages IGW859I and IGW10074I will never be issued.

MAXLOCKS(,)

Valid - this is the equivalent of specifying MAXLOCKS(0,0); messages IGW859I and IGW10074I will never be issued.

MAXLOCKS(5000,0)

Valid - messages IGW859I and IGW10074I will be issued when the asynchronous task wakes up and a unit of recovery has made its 5000th lock request.

MAXLOCKS(0,2000)

Not valid

MAXLOCKS(4500,1000)

Valid - messages IGW859I and IGW10074I will be issued when the asynchronous task wakes up and a unit of recovery has made its 4500th lock request and again every 1000 unique lock requests thereafter.

MAXLOCKS(1000,2300)

Valid - messages IGW859I and IGW10074I will be issued when the asynchronous task wakes up and a unit of recovery has made its 1000th lock request and again every 2300 unique lock requests thereafter.

MAXLOCKS(3200,)

Valid - this is the equivalent of specifying MAXLOCKS(3200,0); messages IGW859I and IGW10074I will be issued when the asynchronous task wakes up and a unit of recovery has made its 3200th lock request.

MAXLOCKS(,2000)

Not valid

The default for both values is 0.

PDSE1_DIRECTORY_STORAGE (nnn)

The operand size values are defined with nnnM for megabytes, or nnnG for gigabytes. For example, to request a 500 megabytes size for the SMSPDSE1 restartable address space 64-bit virtual directory cache, specify SETSMS PDSE1_DIRECTORY_STORAGE(500M). You must restart the SMSPDSE1 address space in order for this change to become effective.

Maximum: 16 Gigabytes

Minimum: 64 Megabytes

PDSE1_HSP_SIZE(nn nn)

Requests up to 2047 megabytes for the PDSE1 hiperspace. You can also

indicate that the hiperspace is not to be created by setting PDSE1_HSP_SIZE to 0. If the hiperspace is not created, the system does not cache PDSE member data. You must restart the SMSPDSE1 address space in order for this change to become effective.

PDSE1_BUFFER_BEYOND_CLOSE (YES | NO)

For the SMSPDSE1 address space, specifies whether to keep directory and member data in memory beyond the last close on this system of a PDSE data set. For the NO option, a PDSE directory and member data will be purged from the in-memory cache when the last close of the data set occurs. If you specify the YES option, the system retains the PDSE directory and member data in the in-memory cache beyond the last close of the data set. You must restart the SMSPDSE1 address space in order for this change to become effective.

QTIMEOUT({nnn|300})

Specifies the quiesce exit timeout value in seconds. The quiesce timeout value specifies the amount of time the DFSMSStvs quiesce exits will allow to elapse before concluding that a quiesce cannot be completed successfully. Specify a value between 60 to 3600. Changing the value of QTIMEOUT affects only those quiesce requests that are submitted after the change is made; it has no effect on quiesce requests that are already in progress. The default is 300.

The command format is:

```
SETSMS QTIMEOUT(300)
```

RLS_MAXCFFEATURELEVEL({A|Z})

Specifies the method that VSAM RLS uses to determine the size of the data that is placed in the CF cache structure. If you specify A, caching proceeds using the RLSCFCACHE keyword characteristics that are specified in the SMS data class that is defined for the VSAM sphere. If you do not specify a value, or if you specify Z, then only VSAM RLS data that have a Control Interval (CI) value of 4K or less are placed in the CF cache structure. The default is Z.

Restrictions:

- If A is specified for the RLS_MAXCFFEATURELEVEL parameter, systems lower than V1R3 will not be able to connect to the CF cache structure.
- If a lower-level system is the first system activated in the sysplex, RLS_MAXCFFEATURELEVEL defaults to Z, and all systems will be able to connect to the CF cache structure.
- If the SETSMS command is used to change the RLS_MAXCFFEATURELEVEL value to A on a mixed-level system, the command is rejected and message IGW500I is issued.

RLS_MAX_POOL_SIZE

Specifies the maximum size in megabytes of the SMSVSAM local buffer pool. SMSVSAM attempts to not exceed the buffer pool size you specify, although more storage might be temporarily used. Because SMSVSAM manages buffer pool space dynamically, this value does not set a static size for the buffer pool.

Use SMF 42, subtype 19 records to help you determine the maximum size of the SMSVSAM local buffer pool.

You can specify a two to four-digit numeric value, with 10 as the minimum value. If you specify a value less than 10, the field is set to 10. If you specify a value greater than 1500, SMSVSAM assumes there is no maximum limit. IBM recommends that you limit the size of the local buffer pool.

The default is 100.

SETSMS Command

RLSINIT({NO|YES})

Specifies whether you want the SMSVSAM address space started. Specify YES if you want the SMSVSAM address space started as part of system initialization or the V SMS,SMSVSAM,ACTIVE command.

RLSTMOUT({nnn|0})

Specifies the maximum time, in seconds, that a VSAM RLS or DFSMStvs request is to wait for a required lock before the request is assumed to be in deadlock and ended with VSAM return code 8 and reason code 22 (X'16'). Specify a value in seconds between 0 to 9999. A value of 0 means that the VSAM RLS or DFSMStvs request has no time out value; the request will wait for as long as necessary to obtain the required lock.

VSAM RLS detects deadlocks within VSAM and DFSMStvs. VSAM RLS cannot detect deadlocks across other resource managers, and uses the timeout value to determine when such deadlocks might have occurred. You can specify a global timeout value in the IGDSMSxx member of SYS1.PARMLIB, a step level timeout value on the JCL, or a timeout value on the RPL passed for each VSAM request.

For a particular VSAM RLS or DFSMStvs request, the value used for timeout is:

1. The value specified in the RPL, if any.
2. The value specified in the JCL at the step level, if any.
3. The value specified in the IGDSMSxx member of SYS1.PARMLIB, if any.

RLSTMOUT is a valid parameter for either VSAM RLS or DFSMStvs. If you specify RLSTMOUT but do not specify the TVSNAME parameter, the value is used only by RLS. For DFSMStvs, the first instance of DFSMStvs brought up within the sysplex determines the value. Subsequent DFSMStvs instances use the value established by the first system, regardless of what might be specified in their members of SYS1.PARMLIB.

RLSTMOUT can be specified only once in a sysplex and applies across all systems in the sysplex.

The default is 0.

RLSABOVETHEBARMAXPOOLSIZE(ALL,size)

RLSABOVETHEBARMAXPOOLSIZE(sysname1,size1;sysname2,size2;... sysname32,size32)

Specifies the total size of the BMF buffer pool that resides above the bar for either of the following:

- All systems
- Each system referenced in the parameter

Valid values are between 500MB and 2,000,000MB (2 Terabytes).

The default is 0.

RLSFIXEDPOOLSIZE(ALL,size)

RLSFIXEDPOOLSIZE(sysname1,size1;sysname2,size2;...sysname32,size32)

Specifies the amount of the total real storage, both above and below the 2 gigabyte bar, that will be permanently fixed (pinned) on either of the following:

- All systems
- Each system referenced in the parameter

The default is 0.

Interval Parameters

INTERVAL(*nnn*)

SMS on the command-issuing system is to allow *nnn* seconds (1 to 999) to pass before synchronizing with the other SMS subsystems running on other MVS systems in the complex. The default value from SMS initialization is 15 seconds. This parameter applies only to the system issuing the command.

DINTERVAL(*nnn*)

Directs SMS to allow *nnn* seconds (1 to 999) to elapse between reading device statistics from a 3990-3 control unit. The default is 150 seconds.

DSSTIMEOUT(*nnnn*)

Specifies the number of seconds that the dss component of DFMSMS will wait during backup processing for quiesce data set requests to complete. Specify a value from zero to 65536 seconds (which is more than 18 hours). If you specify a value between 1 and 299 seconds, the system uses a value of 300 seconds (which equals 5 minutes). The default is 0 seconds.

The value specified in the DSSTIMEOUT parameter value is activated when the first instance of the SMSVSAM address becomes active in the sysplex. All subsequent SMSVSAM instances will use the same value.

CACETIME(*nnnnn*)

Directs SMS to allow *nnnnn* seconds (1 to 86399) to elapse between recording SMS cache control unit summaries for 3990-3 control units. The default is 3600 seconds.

BMFTIME(*nnnnnn*)

Specifies that SMS is to allow *nnnnnn* seconds (1 to 86399) to elapse between the production of SMS BMF records. The default is 3600 seconds.

DEADLOCK_DETECTION(*iiii,kkkk*)

Specifies the deadlock detection intervals used by SMSVSAM.

iiii 1 to 4 digit numeric value in the range 1-9999 that specifies the length in seconds of the local deadlock detection interval. The default for *iiii* is 15 seconds.

kkkk 1 to 4 digit numeric value in the range 1-9999 that specifies the number of local deadlock cycles that must expire before global deadlock detection is performed. The default for *kkkk* is 4 local cycles.

LRUCYCLES(*cycles*)

Specifies the maximum number of times (5 to 240) that the buffer management facility (BMF) least recently used (LRU) routine will pass over inactive buffers before making them available for reuse. This parameter sets the *maximum* value, and BMF dynamically changes the *actual* number of times that it passes over inactive buffers.

LRUCYCLES is related to LRUTIME. A change to the LRUCYCLES value introduced by this parameter will take effect on the next execution of the LRU routine. Most installations should use the default value. In some very high data rate situations you may want to tune this value. You should monitor the SMF 42 type 1 record to determine the amount of caching activity in the BMF data space. See *z/OS MVS System Management Facilities (SMF)* for information about the buffer management statistics recorded in SMF record type 42. The default value is 240 BMF LRU cycles.

LRUTIME(*seconds*)

Specifies the number of seconds (5 to 60) that the buffer management facility (BMF) will wait between calls to the BMF data space cache LRU (least recently

SETSMS Command

used) routine. That routine releases inactive buffers in the BMF data space that are used to cache PDSE (partitioned data set extended) directory data.

LRUTIME is related to LRUCYCLES. A change to the LRUTIME value introduced by this parameter will take effect on the next execution of the LRU routine. Most installations should use the default value. In some very high data rate situations you may want to tune this value. You should monitor the SMF 42 type 1 record to determine the amount of caching activity in the BMF data space. See *z/OS MVS System Management Facilities (SMF)* for information about the buffer management statistics recorded in SMF record type 42. The default value is 15 seconds.

SMF_TIME(YES or NO)

When SMF_TIME(YES) is specified, DFSMS type 42 SMF records are created at the SMF interval time. This parameter overrides all other DFSMS interval time parameters which relate to SMF type 42 records. SMF_TIME(YES) applies to SMF 42 subtypes 1, 2, 15, 16, 17 and 18.

SMF_TIME(NO) specifies that SMF records will not be synchronized at the SMF interval time.

CF_TIME(nn or 3600)

Specifies the interval (in seconds) for recording SMF record 42 (subtypes 15, 16, 17, 18) for the SMSVSAM address space's use of the coupling facility.

If you record these subtypes, you can use CF_TIME to synchronize SMF type 42 data with SMF and RMF data intervals.

Valid values are from 1 to 86399 (23 hours, 59 minutes, 59 seconds). The default is 3600 (one hour).

The SMF_TIME parameter, if set to YES, overrides the CF_TIME parameter.

CICSVR Option Parameters

CICSVR_INIT(YES or NO)

When CICSVR_INIT(YES) is specified, the CICSVR address space will be started as part of system initialization or by the operator issuing V SMS,CICSVR,ACTIVE command.

CICSVR_DSNAME_PREFIX(user_prefix)

Defines a prefix for all CICSVR data set names, excluding RCDS data sets, DWWCON1, DWWCON2, and DWWCOM3 that are created by CICSVR. This allows users to change the naming convention used by CICSVR.

CICSVR_RCDS_PREFIX(cicsvr_rcds_prefix)

Allows operators to change the prefix of CICSVR Recovery Control Data Set (RCDS) names from the console without changing the active IGDSMSxx parmlib member.

CICSVR_ZZVALUE_PARM(zzvalue_string)

Allows operators to change ZZVALUE strings from the operator console without changing the active IGDSMSxx parmlib member.

Trace and VOLSELMSG Option Parameters

TRACE (ON or OFF)

SMS tracing is to be turned on or off.

The following parameters specify the size of the trace table, the type of errors to be traced, the jobname or ASID to be traced, and the particular events that are to be selected (turned on) for tracing or deselected (turned off). The

TRACE, SELECT, and DESELECT parameters apply only to the system on which the operator issues the SETSMS command.

Default values, if no trace values are specified in SYS1.PARMLIB, consist of TRACE (ON), SIZE (128K), TYPE (ERROR), JOBNM (*), which means all jobs, ASID (*), which means all address spaces, and SELECT (ALL). Final values consist of the sum of the defaults (where not overridden), the values in SYS1.PARMLIB, and those added by SETSMS SELECT or removed by SETSMS DESELECT.

SIZE(*nnnnnn*, or *nnnnnnK*, or *nnnM*)

Specifies the size of the trace table in kilobytes. If you omit K or M, the default unit is K. The default value is 128K. The maximum is 255000K or 255M. This value is rounded up to the nearest 4K.

Note: If you specify a size that is different from the previously-used value, a new trace data area is built, the old trace data area is deleted, and no trace data is saved. If, however, the size you specify is the same as the previously-used size, no new trace data area is built. If you specify zero, the existing trace data area is deleted without being replaced.

TYPE(ERROR or ALL)

Specifies how you want to trace events and issue volume selection analysis messages.

ERROR

Specify ERROR to trace error events (TRACE(ON)) and issue volume selection analysis messages (VOLSELMMSG(ON)) on failure allocations. The default value from SMS initialization is ERROR.

ALL

Specify ALL to trace all events (TRACE(ON)) and issue volume selection analysis messages (VOLSELMMSG(ON)) on all allocations.

VOLSELMMSG(ON|OFF,0|*nnnnn*|ALL)

Allows you to control volume selection analysis messages issued when you create or extend a SMS-managed data set to a new volume. These analysis messages are written to the hardcopy log and the joblog.

ON|OFF

Controls whether or not SMS volume selection analysis messages are being issued. The default is OFF.

0|*nnnnn*|ALL

Controls whether or not detailed analysis messages are being issued and the number of volumes to be included in them. The default is 0.

0 Only summarized analysis messages are issued.

nnnnn

Indicate the number of volumes to be included in the message with a range of 0 to 65535.

ALL

Indicates that all volumes used for volume selection will be included in detailed analysis messages.

If you specify VOLSELMMSG(*nnnnn*|ALL), with *nnnnn* having a value greater than 0 along with TYPE(ALL), you must also specify one of the following parameters to limit the number of detailed analysis messages issued:

- JOBNM

SETSMS Command

- ASID
- STEPNAME
- DSNAME

When all volumes are to be included, volumes are listed by storage group. If only a subset of volumes is to be included, volumes are listed in volume selection preference order with no association to storage group.

The system can issue an excessive number of analysis messages to the spool when the following conditions occur:

- The job or address space creates or extends numerous SMS-managed data sets
- Many volumes are to be included in the analysis messages

JOBNAME(*jobname or **)

Specify JOBNAME to limit tracing (TRACE(ON)), issue volume selection messages (VOLSELMMSG(ON)), or both, to particular jobs. Specify *, to select all jobs. If you specify JOBNAME, omit ASID.

ASID(*asid or **)

Specify ASID to limit tracing (TRACE(ON)) and/or issue volume selection messages (VOLSELMMSG(ON)) to particular address spaces. The default is to permit them for all address spaces. Specify *, to select all address spaces. If you specify ASID, omit JOBNAME.

STEPNAME(*stepnamel)**

Limits the number of issued volume selection analysis messages activated by VOLSEGMMSG(ON) to either a certain stepname or all stepnames. The default is to issue volume selection analysis messages on all stepnames.

DSNAME(*dsnamel)**

Limits the number of issued volume selection analysis messages activated by VOLSEGMMSG(ON) to either a certain data set or all data set names. The default is to issue volume selection analysis messages on all data set names. For a VSAM data set, this is the cluster's entry name.

SELECT(*option[,option]...*)

SMS is to add one or more specific events to those that are to be traced. If tracing had been turned off for these events, SMS turns it back on for the specified events. The default is SELECT(ALL). See "Individual Trace Options" for a complete list of the options.

SELECT only adds events; it does not delete any events. Use DESELECT to turn off one or more events.

Both SELECT and DESELECT affect only the system on which you issue the SETSMS command.

DESELECT(*option[,option]...*)

SMS is to delete one or more events from the list of traced events. There is no default for DESELECT. See "Individual Trace Options" for a complete list of the options.

Individual Trace Options

The individual trace options and associated events that you can specify with SELECT or DESELECT are:

MODULE	Module entry or exit
SMSSJF	Storage management subsystem/scheduler JCL facility interfaces
SMSSSI	Storage management subsystem/SSI interfaces
ACSINT	Automatic class selection services interfaces

OPCMD	Operator commands
CONFC	Configuration changes
CDSC	Control data set changes
CONFS	Configuration services
MSG	Message services
DCF	Trace SMS read statistics, Cache maintenance and attribute selection
DPN	IDAX device pool
ERR	Error recovery and recording services
CONFR	Return data from an active configuration
CONFA	Activate a new configuration
ACSPRO	Perform automatic class selection processing
IDAX	SMS interpreter or dynamic allocation
DISP	DISP processing exit
CATG	SMS catalog services
VOLREF	SMS VOLREF services
SCHEDP	Scheduling services (pre-locate catalog orientation)
SCHEDS	Scheduling services (system-select)
TVR	Tape volume record update SSI
VTOCL	VTOC or data set services (allocate existing data set)
VTOCD	VTOC or data set services (delete existing data set)
VTOCR	VTOC or data set services (rename existing data set)
VTOCC	VTOC or data set services (create new data set)
VTOCA	VTOC or data set services (add a volume to a data set)
RCD	SMS recording services or SMS fast VTOC/VVDS access
DSTACK	Trace execution of the SMS data set stacking SSI
UAFF	Unit affinity exit SSI
DEBUG	Debug service
ALL	All of the options

Examples

Example 1

You find that a system with higher I/O capability is locking out slower systems from accessing the communications data set. After checking the value of the interval, currently set at 15 seconds, you decide the interval should be 20 seconds. To make this change, you enter:

```
SETSMS INTERVAL(020)
```

Example 2

Assume that on system MVS3 you want to set the SMS trace table size to 16KB, deselect all trace options, then select three options. To make these changes, enter:

```
SETSMS DESELECT(ALL),TRACE(ON),SIZE(16K)
SETSMS SELECT(ACSINT,CDSC,CONFC)
```

To determine the current tracing status on MVS3, enter:

```
DISPLAY SMS,TRACE
```

The following display would show the trace table size of 16KB and the ON/OFF status of each of the individual trace entries. In this example only the ACS, CDS, and CONFC options would show the ON status.

10.24.04 DISPLAY SMS

```
IGD002I 11:08:57 DISPLAY SMS 056
TRACE = ON SIZE = 128K TYPE = ERROR
```

SETSMS Command

```
JOBNAME = * ASID = *
TRACING EVENTS:
MODULE = ON SMSSJF = ON SMSSSI = ON ACSINT = ON
OPCMD = ON CONFC = ON CDSC = ON CONFS = ON
MSG = ON ERR = ON CONFR = ON CONFA = ON
ACSPRO = ON IDAX = ON DISP = ON CATG = ON
VOLREF = ON SCHEDP = ON SCHEDS = ON VTOCL = ON VTOCD = ON
VTOCR = ON VTOCC = ON VTOCA = ON
RCD = ON DCF = ON DPN = ON TVR = ON
DSTACK = ON UAFF = ON
VOLSELMMSG = (OFF,0) TYPE = ERROR JOBNAME = *
ASID = * STEPNAME = *
DSNAME = *
```

Example 3

You can use the SETSMS command to turn on tracing for SMS data set stacking and other events:

```
SETSMS SELECT(MODULE,DSTACK,VTOCC)
```

Example 4

You can use the SETSMS command to change the activity keypoint value.

```
SETSMS AKP(300)
```

This results in the following output:

```
SETSMS AKP(300)
IGW467I DFSMS TVS ACTIVITY KEY POINT PARMLIB VALUE 959
CHANGED ON SYSTEM: SYSTEM1
OLD VALUE: 200
NEW VALUE: 300
```

Example 5

You can use the SETSMS command to change the quiesce timeout value.

```
SETSMS QTIMEOUT(500)
```

This results in the following output:

```
SETSMS QTIMEOUT(500)
IGW467I DFSMS TVS QTIMEOUT PARMLIB VALUE 962
CHANGED ON SYSTEM: SYSTEM1
OLD VALUE: 400 1
NEW VALUE: 500 1
```

Example 6

You can use the SETSMS command to change the MAXLOCKS maximum and increment values.

```
SETSMS MAXLOCKS(200,100)
```

This results in the following output:

```
SETSMS MAXLOCKS(200,100)
IGW467I DFSMS TVS MAXLOCKS PARMLIB VALUE 965
CHANGED ON SYSTEM: SYSTEM1
OLD VALUE: 100 50 1
NEW VALUE: 200 100 1
```

These MAXLOCKS values are for example only. In practice, you might want to set the minimum and increment values higher to avoid generating too many messages.

Example 7

You can use the SETSMS command to control how to issue the volume selection analysis messages. Assume you want to issue detailed analysis messages for all volumes on failure allocations, enter:

```
SETSMS VOLSELMMSG(ON,ALL) TYPE(ERROR)
```

After the processing completes, check the parameter settings, enter:

```
DISPLAY SMS,VOLSELMMSG
```

This results the following report:

```
IGD002I 08:14:24 DISPLAY SMS 488
VOLSELMMSG = (ON,ALL)      TYPE = ERROR      JOBNAME = *
ASID = *      STEPNAME = *
DSNAME = *
TRACE      = ON      SIZE = 128K      TYPE = ERROR
JOBNAME = *      ASID = *
```

Note that because the TYPE parameter value is shared by SMS trace facility and volume selection analysis messages, changing it for volume selection analysis messages will therefore affect SMS tracing.

SETSSI Command

Use the SETSSI to add, activate, or deactivate a subsystem dynamically. You can issue the SETSSI command from one of the following:

- A console that has master authority
- A console to which an operator with sufficient RACF authority has logged on.

Each subsystem determines whether it can process the SETSSI command by issuing the options request of the IEFSSI macro. See *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG* for more information about the IEFSSI macro.

If you issue a SETSSI ACTIVATE or DEACTIVATE command for a subsystem that does not allow SETSSI commands, the system ignores the command and issues an error message to the console.

You can use the SETSSI ADD command to define dynamically any subsystem except the primary subsystem.

Attention: Once a subsystem name is defined to the system, any attempt to start that subsystem (or any started task with the same name as that subsystem) via a START command which does not explicitly specify SUB=JES2 (or JES3) will result in that subsystem or started task being started under the Master subsystem rather than under the Job Entry subsystem. Then, because the only procedure libraries available to the Master subsystem are those specified in the MSTJCLxx's IEFPDSI data set, any procedures being started that are defined in the Job Entry subsystem's PROC00 data set but not in the MSTJCLxx's IEFPDSI data set will be unavailable and will therefore not be found; the system will issue message IEFC612I.

Syntax

The complete syntax for the SETSSI command is:

```
SETSSI {ADD,{SUBNAME|SUB|S}=subname
        [, {CONSNAME|C}=consname]
        [, {INITRTN|I}=initrtn[, {INITPARM|P}=initparm]] }

        {DEACTIVATE|DEACT},{SUBNAME|SUB|S}=subname      }

        {ACTIVATE|ACT},{SUBNAME|SUB|S}=subname      }
```

Parameters

The parameters are:

ADD

Directs that a subsystem be added dynamically.

Note that with the exception mentioned in the next paragraph, you cannot specify in a SETSSI ADD command a subsystem initialization routine in a library added in a new LNKLST that was activated after IPL. A job that is already running does not normally use a new LNKLST. Because the SETSSI command runs in the MASTER address space, which started at IPL, it does not normally use a new LNKLST.

There is an exception to the above. You could use a SETPROG LNKLST,UPDATE,ASID=1 command to update the LNKLST prior to issuing the SETSSI ADD command. **However, you should use that SETPROG command with caution**, because you would be depending on it to complete execution prior to the running job's accessing the updated LNKLST.

DEACTIVATE I DEACT

Specifies that a subsystem is to be dynamically deactivated. DEACTIVATE stops any new requests from being passed to the subsystem's function routines. Function requests that are already processing are allowed to complete. Note that a subsystem is still defined to the system, even if you issued the DEACTIVATE parameter.

Only subsystems with **SSI-managed** vector tables can be reactivated with the SETSSI command. SSI-managed vector tables are vector tables that were created with the IEFSSVT macro. See *z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG* for more information on the IEFSSVT macro.

Note: You can only issue the DEACTIVATE command if the target subsystem is dynamic and permits the use of the SETSSI command.

ACTIVATE I ACT

Specifies that a subsystem is to be dynamically activated. You can also use the ACTIVATE command to reactivate a previously deactivated subsystem provided a vector table managed by the SSI is available.

Note: You can only issue the ACTIVATE command if the target subsystem is dynamic and permits the use of the SETSSI command.

SUBNAME I SUB I S=subname

Specifies the subsystem name to be dynamically added, deactivated or activated.

Subsystem names that are not enclosed in apostrophes may contain any character that is valid for operator commands, with the following exceptions:

- , comma
- (left parenthesis
-) right parenthesis
- / slash
- = equals sign

Subsystem names containing these characters must be enclosed in apostrophes.

Subsystem names that contain any character that is not valid for operator commands must be enclosed in apostrophes. See Chapter 4, "MVS System Commands Reference," on page 4-1 for a list of characters supported by commands.

Note that the SUBNAME parameter applies to the ADD command, DEACTIVATE command, and the ACTIVATE command. For the SETSSI ADD command, note the following when selecting subsystem names:

- If you specify a subsystem name with the characters '*' and '?', the DISPLAY SSI command or the IEFSSI REQUEST=QUERY service specifying that subsystem name may return information about subsystems other than this one. The '*' and '?' are treated as wildcard characters for these services.

SETSSI Command

- If you specify a subsystem name of '!PRI', the DISPLAY SSI command or the IEFSSI REQUEST=QUERY service specifying that subsystem name returns information about the primary subsystem, even though there is already a subsystem named '!PRI'.

CONSNAME | C=consname

Specifies the name of the console to which SSI issued messages are routed. CONSNAME is an optional parameter. It can be 2- to 8-bytes long and is also passed to the routine named on the INITRTN keyword (if specified).

INITRTN | I=initrtn

Specifies the name of the subsystem initialization routine. INITRTN is an optional parameter. It can consist of at most eight characters, beginning with an alphabetic or national (\$, #, or @) character. The remaining characters can be either alphanumeric or national (\$, #, or @). The routine receives control in supervisor state key 0. It must be a program that is accessible through a LINKLIB or the LPALIB.

INITPARM | P=initparm

Specifies the input parameter that is passed to the subsystem initialization routine. INITPARM is an optional parameter. It can be no more than 60 characters long. If you use delimiters such as blanks, commas, apostrophes, equal signs, or parentheses or a '/' in the parameter data, you must enclose the entire field in apostrophes. You must code two consecutive apostrophes to pass an apostrophe as part of the parameter data.

Note: The INITPARM parameter must be specified with the INITRTN parameter. If the INITRTN is not specified, the system issues a syntax error message and the command is not processed.

Notes:

1. In the command invocation, anything after the first blank is treated as a comment.
2. The command invocation cannot be more than 126 characters long. You may need to use the 1-character keyword abbreviations to keep the length of the command invocation within this limit.

Example 1

To define the 'CAW' subsystem to the system, call its initialization routine and pass the specified parameter to the initialization routine, enter:

```
SETSSI ADD,SUBNAME=CAW,INITRTN=CAWINIT,INITPARM=HELLO
```

Example 2

To temporarily stop new function requests to the subsystem to see if one of the function routines in the 'CAW' subsystem is causing abends, enter:

```
SETSSI DEACTIVATE,SUBNAME=CAW
```

SETUNI Command

Use the SETUNI command to dynamically set the Unicode environment from the console. Use the SETUNI command to add, delete, and replace tables in storage.

Syntax

The complete syntax for the SETUNI command is:

```

SETUNI {ADD[,FROM(xxxxx),TO(yyyyy)[,TECHNIQUE|TECH(zzzzzzzz)]
      [,PAGEFIX(YES|No)][,DSNAME|DSN(dsname)][,VOLSER|VOL(volser)]]
      [,CASE([LOCAL|SPECIAL|NORMAL])[,PAGEFIX(YES|No)]
      [,DSNAME|DSN(dsname)][,VOLSER|VOL(volser)]]]
      [,NORMALIZE|NORM(normver)[,PAGEFIX(YES|No)][,DSNAME|DSN(dsname)]
      [,VOLSER|VOL(volser)]]
      [,COLLATE|COLL([UCAver)][,PAGEFIX(YES|No)][,DSNAME|DSN(dsname)]
      [,VOLSER|VOL(volser)]
      [[,LOCALE(locale)][,DSNAME(dsname)][,VOLUME|VOL(volser)]]]
      [,COLRULES(colrules)[,DSNAME(dsname)][,VOLUME|VOL(volser)]]]
      [,IMAGE(zzzzzzzz)[,PAGEFIX(YES|No)][,DSNAME|DSN(dsname)
      [,VOLSER|VOL(volser)]]
      [,STRPROFILE=NAME[,PAGEFIX(YES|No)][,DSNAME|DSN(dsname)
      [,VOLSER|VOL(volser )]]}

{DELETE[,FROM(xxxxx),TO(yyyyy)[,TECHNIQUE|TECH(zzzzzzzz)],FORCE(YES)]
 [,CASE([LOCAL|SPECIAL|NORMAL]),FORCE(YES)]
 [,NORMALIZE|NORM([normver]),FORCE(YES)]
 [,COLLATE|COLL([UCAver)][[,LOCALE(locale)] |
 [,COLRULES(colrules)],FORCE(YES)]}

{REPLACE[,FROM(xxxxx),TO(yyyyy)[,TECHNIQUE|TECH(zzzzzzzz)]
 [,PAGEFIX(YES|No)][,DSNAME|DSN(dsname)]
 [,FREE(YES,FORCE)]]
 [,CASE([LOCAL|SPECIAL|NORMAL)][,PAGEFIX(YES|No)
 [,DSNAME|DSN(dsname)][,FREE(YES,FORCE)]]
 [,NORMALIZE|NORM([normver)][,PAGEFIX(YES|No)
 [,DSNAME|DSN(dsname)][,VOLSER|VOL(volser)]
 [,FREE(YES,FORCE)]]
 [,COLLATE|COLL([UCAver)][,PAGEFIX(YES|No)
 [,DSNAME|DSN(dsname)][,VOLSER|VOL(volser)]
 [[,LOCALE(locale)][,DSNAME(dsname)][,VOLUME|VOL(volser)]]]
 [,COLRULES(colrules)[,DSNAME(dsname)][,VOLUME|VOL(volser)]]]
 [,FREE(YES,FORCE)]]

{REALSTORAGE [(]nnnnnn[K|M|G][)]}

{DELETE INACTIVE}

{DELETE ALL,FORCE(YES)}

```

Parameters

The parameters are:

**ADD, FROM(xxxxx), TO(yyyyy)[, TECHNIQUE | TECH(zzzzzzzz)]
[, PAGEFIX(YES|No)][, DSNAME | DSN(dsname)][, VOLSER | VOL(volser)]**
Adds specific tables to the Unicode environment.

xxxxx

Specifies the source CCSID of the character conversion table to be added.
xxxxx is a five-character name that identifies the table.

SETUNI Command

yyyyy

Specifies the target CCSID of the character conversion table to be added.
yyyyy is a five-character name that identifies the table.

zzzzzzzz

Specifies the technique search order for the character conversion table to be added. *zzzzzzzz* is an eight-character alphanumeric field. Possible values are one or more of the following:

- R - Roundtrip conversion
- E - Enforced Subset conversion
- C - Customized conversion
- L - Language Environment-Behavior conversion
- M - Modified Language Environment-Behavior conversion
- 0-9 - User-defined conversions

dsname

Specifies the name of the data set to receive the specific tables.

volser

Specifies the volume serial number of the device on which the tables are to be loaded. *volser* can be from one- to six-characters.

ADD,CASE([LOCAL | SPECIAL | NORMAL])[,PAGEFIX(YES|NO)][,DSNAME | DSN(*dsname*)][,VOLSER | VOL(*volser*)]

Adds the character case conversion tables to the Unicode environment. Local, Special, and Normal are optional and can be defined in the same statement only once.

dsname

Specifies the name of the data set to receive the specific tables.

volser

Specifies the volume serial number of the device on which the tables are to be loaded. *volser* can be from one- to six-characters.

ADD,NORMALIZE | NORM ([*normver*])[,PAGEFIX(YES|NO)][,DSNAME | DSN(*dsname*)][,VOLSER | VOL(*volser*)]

Adds the normalization tables to the Unicode environment.

normver

Specifies the Unicode standard table version to be loaded. Possible values are one of the following:

- UNI301
- UNI320
- UNI401
- UNI410

dsname

Specifies the name of the data set to receive the specific tables.

volser

Specifies the volume serial number of the device on which the tables are to be loaded. *volser* can be from one- to six-characters.

ADD,COLLATE | COLL([UCAver])[,PAGEFIX(YES|NO)][,DSNAME | DSN(*dsname*)][,VOLSER | VOL(*volser*)] [[,LOCALE(*locale*)[,DSNAME(*dsname*)][,VOLUMEIVOL(*volser*)]] | [,COLRULES(*colrules*) [,DSNAME(*dsname*)][,VOLUMEIVOL(*volser*)]]]

Adds the collation tables to the Unicode environment.

UCAVer

Specifies the Unicode Collation Algorithm (UCA) versions . Possible values are one or more of the following:

- UCA301
- UCA400R1
- UCA410

dsname

Specifies the name of the data set to receive the specific tables.

volser

Specifies the volume serial number of the device on which the tables are to be loaded. *volser* can be from one- to six-characters.

locale

Specifies the local member name where collation rules are to be loaded.

colrules

Specifies the User Collation Rules (UCR) member name where collation rules are to be loaded.

ADD,IMAGE=zzzzzzzz[,PAGEFIX(YESNO)][,DSNAME | DSN(dsname)][,VOLSER | VOL(volser)]

Adds an image to the Unicode environment, whether the image is a member of the parmlib concatenation. If the image specified already exists in storage, the table is not added.

zzzzzzzz

Specifies the name of the conversion image to be added. The image member specified must be present in SYS1.PARMLIB or in another data set in the logical parmlib concatenation.zzzzzzzz is an eight-character alphanumeric field.

Value Range: any valid z/OS member name

Example: IMAGE=CUNUNI01

Note: When an image is loaded with the IMAGE statement, the existing table in the Unicode environment are not replaced; only those tables that are not currently available in the Unicode environment are loaded from the Image.

dsname

Specifies the name of the data set to receive the specific tables.

volser

Specifies the volume serial number of the device on which the tables are to be loaded. *volser* can be from one- to six-characters.

ADD,STRPROFILE=NAME[,PAGEFIX(YESNO)][,DSNAME | DSN(dsname)][,VOLSER | VOL(volser)]

Adds the profile to the Unicode environment.

dsname

Specifies the name of the data set to receive the specific tables.

volser

Specifies the volume serial number of the device on which the tables are to be loaded. *volser* can be from one- to six-characters.

DELETE,FROM(xxxxx),TO(yyyyy)[,TECHNIQUE | TECH(zzzzzzz)], FORCE(YES)

Remove specific tables from the Unicode environment.

SETUNI Command

xxxxx

Specifies the source CCSID of the character conversion table to be removed. xxxx is a five-character name that identifies the table to be removed.

yyyyy

Specifies the target CCSID of the character conversion table to be removed. yyyy is a five-character name that identifies the table to be removed.

zzzzzz

Specifies the technique search order for the character conversion table to be removed. zzzzzzzz is an eight-character alphanumeric field. Possible values are one or more of the following:

- R - Roundtrip conversion
- E - Enforced Subset conversion
- C - Customized conversion
- L - Language Environment-Behavior conversion
- M - Modified Language Environment-Behavior conversion
- 0-9 - User-defined conversions

FORCE(YES)

Specifies that the system will not check whether applications are currently using the tables. The storage occupied by the tables will be returned to the system.

FORCE(YES) is a required parameter.

DELETE,CASE([LOCAL | SPECIAL | NORMAL]),FORCE(YES)

Removes the character case conversion tables from the Unicode environment. Local, Special, and Normal are optional and can be defined in the same statement only once.

FORCE(YES)

Specifies that the system will not check whether applications are currently using the tables. The storage occupied by the tables will be returned to the system.

FORCE(YES) is a required parameter.

DELETE,NORMALIZE | NORM([normver]),FORCE(YES)

Removes the normalization tables from the Unicode environment.

normver

Specifies the Unicode standard table version to be deleted. Possible values are one of the following:

- UNI301
- UNI320
- UNI401
- UNI410

FORCE(YES)

Specifies that the system will not check whether applications are currently using the tables. The storage occupied by the tables will be returned to the system.

FORCE(YES) is a required parameter.

DELETE,COLLATE | COLL([UCAVer])[,LOCALE(locale)][,COLRULES(colrules)]],FORCE(YES)

Removes the collation tables from the Unicode environment.

UCAv

Specifies the Unicode Collation Algorithm (UCA) versions . Possible values are one or more of the following:

- UCA301
- UCA400R1
- UCA410

locale

Specifies the local member name where collation rules are to be loaded.

colrules

Specifies the User Collation Rules (UCR) member name where collation rules are to be loaded.

FORCE(YES)

Specifies that the system does not check whether applications are currently using the tables. The storage occupied by the tables is returned to the system.

FORCE(YES) is a required parameter.

DELETE,STRPROFILE=NAME[,DSNAME|DSN(dsname)][,VOLSER | VOL(volser)]

Removes the profile from the Unicode environment.

dsname

Specifies the name of the data set to receive the specific tables.

volser

Specifies the volume serial number of the device on which the tables are to be loaded. *volser* can be from one- to six-characters.

REPLACE,FROM(xxxxx),TO(yyyyy)[,TECHNIQUE | TECH(zzzzzzzz)]**[,PAGEFIX(YESINO)][,DSNAME | DSN(dsname)][,FREE(YES,FORCE)]**

Replaces specific tables that might be currently in the Unicode environment. If a table to be replaced is not in storage, the system adds the table.

xxxxx

Specifies the source CCSID of the conversion table to be replaced. *xxxxx* is a five-character name that identifies the conversion table.

yyyyy

Specifies the target CCSID of the conversion table to be replaced. *yyyyy* is a five-character name that identifies the conversion table.

zzzzzzzz

Specifies the technique search order for the conversion table to be replaced. *zzzzzzzz* is an eight-character alphanumeric field. Possible values are one or more of the following:

- R - Roundtrip conversion
- E - Enforced Subset conversion
- C - Customized conversion
- L - Language Environment-Behavior conversion
- M - Modified Language Environment-Behavior conversion
- 0-9 - User-defined conversions

FREE

Specifies whether the storage associated with the table is to be released.

(YES,FORCE)

Release the storage associated with the table.

SETUNI Command

| **REPLACE,CASE([LOCAL | SPECIAL | NORMAL])[,PAGEFIX(YES|NO)]
[,DSNAME | DSN(dsname)][,VOLSER | VOL(volser)] [,FREE(YES,FORCE)]**
| Replaces the character case conversion tables currently in the Unicode
| environment. Local, Special, and Normal are optional and can be defined in the
| same statement only once.

| *dsname*

| Specifies the name of the data set to receive the specific tables.

| *volser*

| Specifies the volume serial number of the device on which the tables are to
| be loaded. *volser* can be from one- to six-characters.

| **FREE**

| Specifies whether the storage associated with the table is to be released.

| **(YES,FORCE)**

| Release the storage associated with the table.

| **REPLACE,NORMALIZE | NORM([normver])[,PAGEFIX(YES|NO)][,DSNAME |
DSN(dsname)][,VOLSER | VOL(volser)][,FREE(YES,FORCE)]**

| Replaces the normalization tables currently in the Unicode environment.

| *normver*

| Specifies the Unicode standard table version to be replaced. Possible
| values are one of the following:

- UNI301
- UNI320
- UNI401
- UNI410

| *dsname*

| Specifies the name of the data set to receive the specific tables.

| *volser*

| Specifies the volume serial number of the device on which the tables are to
| be loaded. *volser* can be from one- to six-characters.

| **FREE**

| Specifies whether the storage associated with the table is to be released.

| **(YES,FORCE)**

| Release the storage associated with the table.

| **REPLACE,COLLATE | COLL([UCAver])[,PAGEFIX(YES|NO)] [,DSNAME |
DSN(dsname)][,VOLSER | VOL(volser)][[,LOCALE(locale)[,DSNAME(dsname)]
,VOLUMEIVOL(volser)]] | [,COLRULES(colrules)[,DSNAME(dsname)]
,VOLUMEIVOL(volser)]]] [,FREE(YES,FORCE)]**

| Replaces the collation tables currently in the Unicode environment.

| *UCAver*

| Specifies the Unicode Collation Algorithm (UCA) versions . Possible values
| are one or more of the following:

- UCA301
- UCA400R1
- UCA410

| *dsname*

| Specifies the name of the data set to receive the specific tables.

volser

Specifies the volume serial number of the device on which the tables are to be loaded. *volser* can be from one- to six-characters.

locale

Specifies the local member name where collation rules are to be loaded.

colrules

Specifies the User Collation Rules (UCR) member name where collation rules are to be loaded.

FREE

Specifies whether the storage associated with the table is to be released.

(YES,FORCE)

Release the storage associated with the table.

REPLACE,STRPROFILE=NAME[,PAGEFIX(YES|NO)][,DSNAME | DSN(*dsname*)][,VOLSER | VOL(*volser*)]

Replaces the current profile in the Unicode environment.

dsname

Specifies the name of the data set to receive the specific tables.

volser

Specifies the volume serial number of the device on which the tables are to be loaded. *volser* can be from one- to six-characters.

REALSTORAGE *nnnnnn[K | M | G] [:]*

Defines the upper storage limit, in pages, to be used by the conversion environment. For information about the amount of storage required for a conversion environment, see *z/OS Support for Unicode: Using Unicode Services*.

Value Range: 0 to 524287.

Example:

```
REALSTORAGE 0;      /* no explicit limit */
REALSTORAGE 12800; /* 50 MB limit */
```

Notes:

1. The request to load a new conversion environment will be rejected when the value of the REALSTORAGE keyword is lower than the amount of storage needed.
2. The selection of '0' results in no limit (=524287).

DELETE mode

Deletes partially or completely the Unicode environment.

INACTIVE

Deletes all the unreferenced control entries within the current Unicode environment and reorganizes the Unicode environment to eliminate storage gaps in it.

The string literal INACTIVE must be specified.

Note: Unreferenced control entries are entities that contain data of the current supported tables and can be obtained while replacing or deleting tables from the Unicode environment.

ALL Deletes the whole Unicode environment removing all control structures and tables from the environment.

SETUNI Command

Note: FORCE(YES) is required for this keyword.

FORCE(YES)

Specifies that the system will not check whether applications are currently using the tables. The storage occupied by the tables will be returned to the system.

FORCE(YES) is a required parameter.

Example 1:

```
DELETE INACTIVE;
```

Example 2:

```
DELETE ALL,FORCE(YES)
```

Note: DELETE mode commands are intended for Unicode environment maintenance only. No Unicode calls should be running in the Unicode environment during the execution of these commands.

Example 1

```
SETUNI ADD,CASE,DSN=SYS1.SCUNTB1,VOL=XLAD01
```

Example 2

```
SETUNI ADD,FROM=1200,T0=37,TECH=ER,DSN=SYS1.SCUNTB1,VOL=XLAD01
```

SETXCF Command

Use the SETXCF command to control the cross-system coupling facility (XCF). Table 4-36 summarizes the information that the SETXCF command provides. Use it to access the pages on which you can find details about a particular use of the SETXCF command.

Table 4-36. Summary of the SETXCF Command

Command	Topic
SETXCF COUPLE	"SETXCF COUPLE Command"
SETXCF FORCE	"SETXCF FORCE Command" on page 4-487
SETXCF MODIFY	"SETXCF MODIFY Command" on page 4-489
SETXCF PRSMOPOLICY	"SETXCF PRSMOPOLICY Command" on page 4-491
SETXCF START	"SETXCF START Command" on page 4-492
SETXCF STOP	"SETXCF STOP Command" on page 4-501

Scope in a Sysplex

The following table describes the conditions under which the SETXCF command has sysplex scope. See "Using Commands That Have Sysplex Scope" on page 1-11 for an explanation of sysplex scope.

Table 4-37. Sysplex Scope for the SETXCF Command

Command	Conditions
SETXCF COUPLE	Has sysplex scope only when you specify PSWITCH, ACOUPLE, or PCOUPLE, and all systems have access to the specified couple data set.
SETXCF FORCE	Has sysplex scope only when all systems are connected to the same coupling facility.
SETXCF START STOP	Has sysplex scope only when you specify POLICY, REBUILD, REALLOCATE, or MSGBASED.

Syntax

The syntax for each variation of the SETXCF command is shown immediately preceding its respective parameter list.

SETXCF COUPLE Command

Use the SETXCF COUPLE command to:

- Switch a current alternate couple data set to a primary couple data set. The switch can be for either sysplex couple data sets or other types of couple data sets.
- Specify a primary non-sysplex couple data set, such as CFRM, SFM, WLM.
- Specify an alternate couple data set.
- Change options specified in the COUPLExx parmlib member.

For more information about the SETXCF COUPLE parameters see *z/OS MVS Setting Up a Sysplex* and *z/OS MVS Initialization and Tuning Reference*.

SETXCF Command

```
SETXCF COUPLE,{PSWITCH
  {ACOUPLE=(alternatedsname[,alternatevolume])}
  {INTERVAL=timeinterval}
  {OPNOTIFY=timeinterval}
  {CLEANUP=timeinterval}
  {MAXMSG=defaultmaxmsgbuffers}
  {RETRY=defaultretrylimit}
  {CLASSLEN=defaultclasslength}
  {TYPE=(name[,name]...),
   {PCOUPLE=(primarydsname[,primaryvolume])}
   {ACOUPLE=(alternatedsname[,alternatevolume])}
  {PSWITCH}}
```

The parameters are:

PSWITCH

Switches the current alternate sysplex couple data set to become the primary sysplex couple data set. The command is complete when systems in the sysplex acknowledge the switch. This command removes the current primary sysplex couple data set from service.

Note: If the new primary sysplex couple data set has been formatted to support greater than eight systems in the sysplex, the following occurs:

- If the current RMAX value is less than 99, the system automatically increases the RMAX value to 99. The system issues message IEA403I to indicate this change. You cannot lower the value of RMAX. A minimum RMAX value of 99 is enforced for performance reasons.

ACOUPLE=(*alternatedsname,alternatevolume*)

Specifies the data set to use as an alternate sysplex couple data set. This data set must be defined and formatted with the XCF format utility and for each parameter specified in the utility, the parameter values must be equal to or greater than the parameter values that were used to format the current primary couple data set. See *z/OS MVS Setting Up a Sysplex* for additional information about planning for couple data sets.

Specify the volume only when the data set is not cataloged. You need to use parentheses only when you specify the volume. After the command completes, any previous alternate sysplex couple data set is removed from service. The specified alternate sysplex couple data set must be accessible and usable from all systems in the sysplex.

The data set named *alternatedsname* can have one or more name segments, separated by periods, and cannot exceed a total length of 44 characters. Each name segment is one to eight alphanumeric, hyphen (-), and national (\$,#,@) characters, but must begin with an alphabetic or national character.

The volume *alternatevolume* must be specified as one to six alphanumeric or national characters, and may begin with any of these characters.

INTERVAL=*timeinterval*

Specifies the length of the failure detection interval for the system. *timeinterval* is specified in seconds and ranges from 3 to 86400 (24 hours).

OPNOTIFY=*timeinterval*

Specifies how long a system must appear to be inoperative before XCF notifies the operator. *timeinterval* is specified in seconds and ranges from 3 to 86400 (24 hours). The value must be greater than or equal to the **INTERVAL** value. When this parameter is omitted, the default is the failure detection interval plus 3 seconds.

CLEANUP=*timeinterval*

Specifies the time interval that XCF waits for multisystem applications to complete cleanup functions. The interval begins after XCF sends notification to group members that the system on which they are running is being removed from the sysplex. *timeinterval* is specified in seconds and ranges from 0 to 86400 (24 hours).

MAXMSG=*defaultmaxmsgbuffers*

Specifies the default value used if the MAXMSG keyword is not specified on the SETXCF START command. The MAXMSG value must be a number from 1 to 999999. See *z/OS MVS Setting Up a Sysplex* for further information about determining message buffer space.

RETRY=*defaultretrylimit*

Specifies the default value used if the RETRY keyword is not specified on the SETXCF START command. It is the number of failures that XCF tolerates before it marks a path as inoperative. Specify a value between 3 and 255.

CLASSLEN=*defaultclasslength*

Specifies the default message length for the transport classes, used if the CLASSLEN keyword is not specified on the SETXCF START command. XCF uses this length to optimize its processing for messages sent in a transport class. Specify a value between 0 and 62464.

TYPE=*(name,name...)*

Specifies the type of data stored in this couple data set.

The supported names include:

- SYSPLEX for sysplex (XCF) types
- ARM for automatic restart management
- CFRM for coupling facility resource management
- LOGR for system logger
- SFM for sysplex failure management
- WLM for workload management
- BPXMCDS for UNIX System Services

Other TYPES might exist for other components. Please check the component documentation for information on what to specify for TYPE in this and other XCF operator commands and the XCF COUPLExx parmlib member.

The name or names specified must correspond to the name or names used when the couple data set was formatted with the couple data set format utility, IXCL1DSU. See *z/OS MVS Setting Up a Sysplex* for a description of the couple data set format utility.

The following keywords all refer to the couple data sets that support the service specified by the TYPE keyword.

SETXCF Command

PCOUPLE=(*primarydsname,primaryvolume*)

Specifies the data set to use as the primary couple data set for the type of service specified by TYPE=. Note that you cannot specify PCOUPLE to identify the sysplex couple data set (which is initially specified in the COUPLExx parmlib member).

The data set must be defined and formatted with the XCF format utility.

If the service is already operational in the sysplex, the system ignores the data set specified by PCOUPLE. Instead, the system attempts to make the service available to the system by using the couple data set that is currently supporting the service on other systems in the sysplex.

If the service is not already operational in the sysplex, the system attempts to use the specified data set as the primary couple data set for the service specified.

When TYPE=(CFRM) is specified and the CFRM couple data set is added to the sysplex, it MUST NOT BE REMOVED OR DELETED. If the CFRM couple data set is removed from the sysplex, the sysplex system will enter a non-restartable WAIT STATE. This is true even if no CFRM policies were activated.

ACOUPLE=(*alternatedsname,alternatevolume*)

Specifies the data set to use as the alternate couple data set for the type of service specified by TYPE=. The data set must already be allocated on the volume specified. The data set must be defined and formatted with the XCF format utility.

For each parameter specified in the format utility, the parameter values must be equal to or greater than the parameter values that were used to format the current primary couple data set. See "Planning the Couple Data Sets" in *z/OS MVS Setting Up a Sysplex* for additional information.

When you add a new alternate couple data set to the sysplex, the system copies the policies on the primary couple data set to that new alternate couple data set. The policies it copies from the primary couple data set replace any policies that existed on the alternate couple data set.

You do not need to specify the *alternatevolume*. If the volume is not specified, the data set must be cataloged. If the volume is specified, the system does not use the catalog to locate the data set.

If the system can use the specified couple data set for the service specified, then the data set becomes the alternate couple data set for that service.

The system deallocates the data set that the system had been using as the alternate couple data set for the service at the time the SETXCF command was issued if the data set is no longer in use for any service in the sysplex.

PSWITCH

Specifies that the current alternate couple data set for the type of service defined is to become the primary couple data set. The system stops using the current primary couple data set and deallocates it if it is no longer in use by any service in the sysplex.

Using the TYPE Keyword

Remember the following when using the TYPE keyword:

- If you use PSWITCH with the TYPE keyword, the couple data set switched is the one containing the type of data specified (ARM, BPXMCDS, CFRM, LOGR, SFM or WLM).
- If you use PSWITCH without the TYPE keyword, the sysplex couple data set is switched.
- If you use ACOUPLE with the TYPE keyword, the alternate couple data set is the one containing the type of data specified (ARM, BPXMCDS, CFRM, LOGR, SFM or WLM). The specified alternate couple data set must be accessible and usable from all systems in the sysplex that are using the policy for that service.
- If you use ACOUPLE without the TYPE keyword, the alternate couple data set is the sysplex couple data set and, as such, must be accessible and usable from all systems in the sysplex.

SETXCF FORCE Command

Use the SETXCF FORCE command to clean up coupling facility structures without connections, persistent connections to coupling facility structures, coupling facility structure dumps, coupling facility structure dump serialization, or Automatic Restart Management (ARM) elements.

A structure is identified by its structure name, which can be up to 16 characters long. The name can contain numeric characters, uppercase alphabetic characters, national characters (\$, @, #), or an underscore (_). The structure name must begin with an uppercase alphabetic character. Structure names provided by IBM begin with SYS or the letters A through I.

Note that if you issue the SETXCF FORCE command for a coupling facility resource from a system that does not have connectivity to the coupling facility that contains the resource, the system accepts the command; however, the command remains pending until either the system establishes connectivity with the coupling facility or another system processes the deletion.

When forcing deregistration of an ARM element, you must issue the command from the system on which the element registered. If the system on which the element registered is no longer active, issue the command from any system in the sysplex. The system will respond to a valid ARMDEREGISTER command with a message indicating whether the command was completed or rejected — IXC393I. If the element might be in use by the system, you must reply to the IXC394A message issued before getting the IXC393I.

Attention: To reduce the risk of losing data, do not force the deletion of structures, connections, or ARM elements unless you understand their use in the sysplex by applications or subsystems.

For additional information about the circumstances under which to issue the SETXCF FORCE command, see "Coupling Facility Replacement and Reconfiguration Guidelines" in *z/OS MVS Setting Up a Sysplex*.

For more information about the SETXCF FORCE parameters see both *z/OS MVS Setting Up a Sysplex* and *z/OS MVS Initialization and Tuning Reference*.

SETXCF Command

```
SETXCF FORCE,  
  {STRUCTURE,STRNAME=(strname[,strname]...)}  
  {CONNECTION,STRNAME=strname,CONNNAME={(connname[,connname]...)|ALL}}  
  {STRDUMP,STRNAME=strname[,STRDUMPID=strdumpid]}  
  {STRDUMPSERIAL,STRNAME=strname[,STRDUMPID=strdumpid]}  
  {ARMDEGERISTER,ELEMENT=element}
```

STRUCTURE or STR

Directs the system to force the deletion of a named coupling facility structure. If a dump for the specified structure is in progress, deallocation of the structure remains pending until either the dump is complete or the dump is forced using the STRDUMP option. The system notifies the operator that the command is accepted but that the structure is pending deallocation.

STRNAME or STRNM=(*strname*[,*strname*]...)

Identifies one or more persistent coupling facility structures to delete. In order to delete the structure with the SETXCF FORCE command, the structure must have no active connections. If only failed-persistent connections to the structure exist, the structure can be deleted with the SETXCF FORCE command.

CONNECTION or CON

Directs the system to force the deletion of a failed-persistent connection. The system will not force failed-persistent connections to a persistent lock or serialized list structure because of the possibility of an undetected loss of data. The system will reject such a FORCE,CONNECTION request with message IXC354I or IXC363I.

STRNAME or STRNM=*strname*

Specifies the name of an active structure that contains the connection to be deleted. *strname* can be up to 16 alphanumeric characters long and must begin with an uppercase alphabetic character.

CONNNAME= or CONNM=(*connname*[,*connname*]...) or ALL

Directs the system to delete one or more connections to the named structure. You can use SETXCF FORCE only to delete a failed-persistent connection.

When you specify CONNAME=ALL, you request the system to delete all failed-persistent connections to the specified structure.

STRDUMP or STRD

Specifies the system is to force the deletion of a structure dump. The structure is either actively in use by the sysplex or pending deallocation. Under normal operating conditions when a structure dump is associated with a structure in the coupling facility, the structure dump identifier is recorded in the active CFRM policy. However, if a situation arises where the structure dump identifier does not get recorded in the active CFRM policy, the SETXCF FORCE,STRDUMP command can still be used to delete the structure dump. To force the structure dump in this case, omit the STRDUMPID= keyword and the system will determine the structure dump identifier.

STRNAME= or STRNM=*strname*

Specifies the name of the structure for which to delete the structure dump.

STRDUMPID= or STRDID=*strdumpid*

Identifies the structure dump to be deleted. The structure dump identifier uniquely differentiates between a structure dump associated with a structure actively in use in the sysplex and a structure dump associated with a structure pending deallocation. Use the DISPLAY XCF,STRUCTURE command to determine the structure dump identifier.

The structure dump identifier can be up to four hexadecimal digits. If you omit this keyword, the structure dump associated with the named structure actively in use in the sysplex is the one that is deleted.

STRDUMPSERIAL or STRDSER

Specifies that the system is to release its dumping serialization for a coupling facility structure. The structure must be actively in use in the sysplex. Under normal operating conditions when a structure dump is associated with a structure in the coupling facility, the structure dump identifier is recorded in the active CFRM policy. However, if a situation arises where the structure dump identifier does not get recorded in the active CFRM policy, the SETXCF FORCE,STRDUMPSERIAL command can still be used to release the structure dump serialization. To force the structure dump serialization in this case, omit the STRDUMPID= keyword and the system will determine the structure dump identifier.

STRNAME= or STRNM=*strname*

Identifies the structure for which to release dump serialization.

STRDUMPID= or STRDID=*strdumpid*

Identifies the structure dump associated with the structure for which to release dump serialization.

ARMDREGISTER

Directs the system to force deregistration of an ARM element.

ELEMENT

Identifies the name of the ARM element to be deregistered.

SETXCF MODIFY Command

Use the SETXCF MODIFY command to change current XCF parameters. The system changes only those parameters explicitly provided on the SETXCF MODIFY command; all other parameters associated with the resource remain the same.

- Modify inbound paths.
- Modify outbound paths.
- Modify local message space.
- Modify transport classes.

SETXCF Command

The complete syntax for the SETXCF MODIFY command is:

```
SETXCF MODIFY,{PATHIN,{DEVICE=([/]indevnum[,/]indevnum]...)}    }
               {STRNAME=(strname[,strname]...)}      }
               [,MAXMSG=maxmsgbuffers]
               [,RETRY=retrylimit]

               {PATHOUT,{DEVICE=([/]outdevnum[,/]outdevnum]...)}  }
               {STRNAME=(strname[,strname]...)}      }
               [,CLASS=classname]
               [,MAXMSG=maxmsgbuffers]
               [,RETRY=retrylimit]

               {LOCALMSG,MAXMSG=maxmsgbuffers          }
               [,CLASS=class-name]

               {CLASSDEF,CLASS=classname                }
               [,CLASSLEN=classlength]
               [,MAXMSG=defaultmaxmsgbuffers]
               [,ADDGROUP=(groupname[,groupname]...)]
               [,DELGROUP=(groupname[,groupname]...)]
```

The parameters are:

PATHIN or PI,DEVICE= or DEV=([/]indevnum,[/]indevnum) [,RETRY=retrylimit]

Specifies the device number of one or more inbound signalling paths. A device number, *indevnum*, is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). If you specify only one device, you do not need to enter the parentheses.

Use the RETRY keyword to modify the retry limit and the MAXMSG keyword to modify the amount of message buffer space. You must specify at least one of the MAXMSG or RETRY keywords. The MAXMSG value must be a number from 1 to 999999.

PATHIN or PI,STRNAME or STRNM=(strname[,strname]...)

Specifies the name of one or more coupling facility structures that are associated with defined XCF inbound signalling paths and that are to be modified.

The structure name can be up to 16 alphanumeric characters long, and must begin with an uppercase alphabetic character. If you specify only one structure name, you do not need to enter the parentheses.

**PATHOUT or PO,DEVICE or DEV=([/]outdevnum,[/]outdevnum)
 [,MAXMSG=maxmsgbuffers][,RETRY=retrylimit]**

Specifies the device number of one or more outbound signalling paths. A device number, *outdevnum*, is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). If you specify only one device, you do not need to enter the parentheses.

Use the RETRY keyword to modify the retry limit, the CLASS keyword to modify the transport class assignment, and the MAXMSG keyword to modify the amount of message buffer space. You must specify at least one of the MAXMSG, CLASS, or RETRY keywords. The MAXMSG value must be a number from 1 to 999999.

PATHOUT or PO,STRNAME or STRNM=(strname[,strname]...)

Specifies the name of one or more coupling facility structures that are associated with defined XCF outbound signalling paths and that are to be modified.

The structure name can be up to 16 alphanumeric characters long, and must begin with an uppercase alphabetic character. If you specify only one structure name, you do not need to enter the parentheses.

LOCALMSG or LM,MAXMSG=*maxmsgbuffers* [,CLASS=*classname*]

Within a particular transport class, you can modify the amount of message buffer space made available for local message traffic after the creation of the class definition. The total amount of buffer space for the transport class (indicated by the optional keyword CLASS) is *maxmsgbuffers* plus the value specified or defaulted to on the MAXMSG parameter on the CLASSDEF statement of the COUPLExx parmlib member. The MAXMSG value must be a number from 1 to 999999.

If you omit CLASS, the change affects the default transport class, named DEFAULT.

CLASSDEF or CD,CLASS=*classname*

Specifies the transport class to be modified. You must specify at least one of the CLASSLEN, MAXMSG, ADDGROUP, or DELGROUP keywords.

[,CLASSLEN=*classlength*]

Use the CLASSLEN keyword to modify the message length.

MAXMSG=*defaultmaxmsgbuffers*

Use the MAXMSG keyword to modify the message buffer space. By changing the MAXMSG value, you change the amount of message buffer space initially allotted to each system for the indicated transport class. However, changing the MAXMSG value does not affect it for any existing resource — such as an outbound XCF path — in the class. The MAXMSG value must be a number from 2 to 999999. See *z/OS MVS Setting Up a Sysplex* for further information about determining message buffer space.

ADDGROUP=(*groupname*[,*groupname*]...)

Specifies one or more groups to be added to the set of groups assigned to the transport class. Groups, possibly including undesignated groups, were assigned to this class when it was created. Explicitly assigning a group to this class does not delete the UNDESIG group from the class.

The group name *groupname* must be specified as one to eight alphanumeric and national (\$,#,@) characters, and may begin with any of these characters.

DELGROUP=(*groupname*[,*groupname*]...)

One or more groups to be deleted from the set of groups assigned to the transport class. If this command deletes the last assigned group from a transport class, XCF automatically assigns the undesignated groups to the class. If a transport class has no groups explicitly assigned to it, undesignated groups cannot be deleted from the class.

The group name *groupname* must be specified as one to eight alphanumeric and national (\$,#,@) characters, and may begin with any of these characters.

SETXCF PRSMOPOLY Command

Use the SETXCF PRSMOPOLY (or PRSMOPOL) command to either:

- Activate an XCF PR/SM policy, or
- Deactivate a current active XCF PR/SM policy.

In a multisystem sysplex on PR/SM, the XCF PR/SM policy provides a way for the installation to obtain high availability for multisystem applications on the MVS

SETXCF Command

systems in the sysplex. See *z/OS MVS Initialization and Tuning Reference* and *z/OS MVS Setting Up a Sysplex* for information about the XCF PR/SM policy.

Note: When an active Sysplex Failure Management policy is in effect in the sysplex, the system rejects the SETXCF PRSMPOLICY command.

```
SETXCF PRSMPOLICY,{DEACTIVATE|ACTIVATE=memname}
```

The parameters are:

ACTIVATE=memname

Specifies the member in SYS1.PARMLIB that contains the XCF PR/SM policy to be activated. If an error occurs while processing the member, the current XCF PR/SM policy, if any, remains in effect.

DEACTIVATE

Specifies that all XCF PR/SM policy processing is to be stopped.

Example

To activate the XCF PR/SM policy contained in member XCFPOL03 parmlib member, enter:

```
SETXCF PRSMPOLICY,ACTIVATE=XCFPOL03
```

SETXCF START Command

Use the SETXCF START command to:

- Start new inbound signalling paths or restart inoperative inbound signalling paths
- Start outbound signalling paths or restart inoperative outbound signalling paths
- Define transport classes
- Start using a new administrative policy as an active policy
- Start rebuilding one or more coupling facility structures either in the same coupling facility or in another coupling facility
- Start populating a coupling facility that has been newly brought into service or returned to service in a sysplex with structures selected from the set of those defined in the active CFRM policy. The structures selected are those that list the coupling facility to be populated as higher in the structure's preference list than the coupling facility in which the structure already is allocated
- Start user-managed duplexing of one or more structures in a coupling facility into another coupling facility
- Start altering the size of a coupling facility structure
- Start the REALLOCATE process. The REALLOCATE process uses the XCF structure allocation algorithm and the structure definition in the active or pending CFRM policy to evaluate each allocated structure. When this evaluation indicates that adjustments can be made for an allocated structure, then REALLOCATE processing makes the appropriate adjustments.
- Transition from a policy-based event and confirmation management protocol to a message-based event and confirmation management protocol.

An inoperative signalling path remains defined to XCF but is not in use. DISPLAY XCF lists inoperative as well as active signalling paths. When you restart an

inoperative signalling path with SETXCF START, you can alter MAXMSG, RETRY, and CLASS. These values take effect when XCF restarts the signalling path.

The complete syntax for the SETXCF START command is:

```

SETXCF START,{CLASSDEF,CLASS=classname} }
[,CLASSLEN=classlength]
[,MAXMSG=maxmsgbuffers]
[,GROUP=(groupname[,groupname]...)] }

| {MSGBASED}
{PATHIN,{DEVICE=([/]indevnum[,/]indevnum]...) } }
{STRNAME=(strname[,strname]...) } }
[,MAXMSG=maxmsgbuffers]
[,RETRY=retrylimit]

{PATHOUT,{DEVICE=([/]outdevnum[,/]outdevnum]...) } }
{STRNAME=(strname[,strname]...) } }
[,MAXMSG=maxmsgbuffers]
[,RETRY=retrylimit]
[,CLASS=classname]

{POLICY,TYPE=name,POLNAME=polname} }

{REBUILD,{POPULATECF=cfname} }
{DUPLEX,}
{STRNAME=(strname[,strname]...) } }
{CFNAME=(cfname[,cfname]...) } }
[,LOCATION={NORMAL|OTHER}]
[,LESSCONN={TERMINATE|CONTINUE}] }

{ALTER,STRNAME=strname,SIZE=size} }
{REALLOCATE} }
```

CLASSDEF or CD,CLASS=classname

Specifies a definition for a new transport class. You must specify the *classname* as one to eight alphanumeric and national (\$#, @) characters, and may begin with any of these characters. The DEFAULT transport class always exists. If you specify only one class, you do not need to enter the parentheses.

CLASSLEN=classlength

Specifies the message length for this transport class, where *classlength* must be a number between 0 and 62464. If you omit this keyword, the system uses the current CLASSLEN value (specified either on the SETXCF COUPLE command or in the COUPLExx parmlib member).

MAXMSG=maxmsgbuffers

Specifies the default amount of message buffer space allotted for messages sent in this transport class, where MAXMSG value must be a number between 2 and 999999. See *z/OS MVS Setting Up a Sysplex* for further information about determining message buffer space.

If you omit this keyword, the system uses the MAXMSG value specified on either the SETXCF COUPLE command or in the COUPLE statement in the COUPLExx parmlib member).

GROUP or GRP =(groupname[,groupname]..)

Assigns one or more groups to the transport class. The order in which groups are specified is unimportant; all groups have equal access to the

SETXCF Command

signalling services. If you omit this keyword, XCF assigns all groups not currently assigned (the UNDESIG groups), to the classname specified on the CLASSDEF parameter.

You must specify the group name *groupname* as one to eight alphanumeric and national (\$, #, @) characters, and may begin with any of these characters.

MSGBASED

Switch to a message-based event and confirmation management protocol from a policy-based event and confirmation protocol. The sysplex must be running with a version of the CFRM couple data set that supports message-based event and confirmation processing in order for this command to execute successfully. The system chooses the system where the SETXCF START,MSGBASED command was issued as the new event managing system.

PATHIN or PI,DEVICE or DEV=([/]indevnum,[/]indevnum)...

Specifies the device number of one or more inbound signalling paths that XCF can use. A device number, *indevnum*, is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). If you specify only one device, you do not need to enter the parentheses.

The specified device must be unallocated. The system at the other end of a signalling path must be inactive or part of the sysplex, and the other end of the path must be defined to XCF as an outbound path.

Once the command completes, the other systems in the sysplex can begin to send signalling traffic on the path to this system. For a signalling path device that is not online, MVS can vary the device online when you use this command to start the signalling path. The device remains defined to XCF until the definition is deleted through the SETXCF STOP command.

PATHIN or PI,STRNAME or STRNM=(strname[,strname]...)

Specifies the name of one or more coupling facility structures that XCF can use as inbound signalling paths. The designated structure is defined as a signalling path only to this system and not to every system in the sysplex. Other systems that are connected to the structure must also define the structure for use as a signalling path.

The structure name *strname* can be up to 16 alphanumeric characters long and must begin with 'IXC'. The remaining characters can be numeric, uppercase alphabetic, national characters (\$, @, #), or an underscore (_). If you specify only one structure name, you do not need to enter the parentheses.

MAXMSG=maxmsgbuffers

Specifies the maximum amount of message buffer space, in kilobytes, that XCF can use to receive messages through the inbound signalling path. If you omit this keyword, the system uses the current MAXMSG value (specified on either the SETXCF COUPLE command or the COUPLE statement of the COUPLExx parmlib member). The MAXMSG value must be a number from 1 to 999999.

RETRY=retrylimit

Specifies the *retrylimit* for the inbound signalling paths. If you omit this keyword, the system uses the RETRY value, a number between 3 and 255 specified on either the SETXCF COUPLE command or the COUPLE statement of the COUPLExx parmlib member.

PATHOUT or PO,DEVICE or DEV=([/]outdevnum,[/]outdevnum)..

Specifies the device number of one or more outbound signalling paths that XCF

can use. A device number, *outdevnum*, is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). If you specify only one device, you do not need to enter the parentheses.

The specified device must be unallocated. The system at the other end of the path must be inactive or part of the sysplex, and the other end of the path must be defined to XCF as an inbound path. Once the command completes successfully, the other system in the sysplex can begin to receive signalling traffic on the path from this system. For a signalling path device that is not online, MVS can vary the device online when you use this command to start the signalling path. The device remains defined to XCF until the definition is deleted through the SETXCF STOP command.

PATHOUT or PO,STRNAME or STRNM=(*strname*[,*strname*]...)

Specifies the name of one or more coupling facility structures XCF can use as outbound signalling paths. The designated structure is defined only to this system, not to every system in the sysplex. Other systems connected to the structure must also define the structure for use as a signalling path.

The structure name *strname* can be up to 16 alphanumeric characters long and must begin with 'IXC'. The remaining characters can be numeric, uppercase alphabetic, national characters (\$, @, #), or an underscore (_). If you specify only one structure name, you do not need to enter the parentheses.

MAXMSG=*maxmsgbuffers*

Specifies the amount of message buffer space contributed by this signalling path. If you omit this keyword, the system uses the current MAXMSG value for the transport class that this path is assigned. The MAXMSG value must be a number between 1 and 999999.

RETRY=*retrylimit*

Specifies the *retrylimit* for the outbound signalling path. If you omit this keyword, the system uses the current RETRY value, a number between 3 and 255 specified on either the SETXCF COUPLE command or the COUPLE statement of the COUPLExx parmlib member.

CLASS=*classname*

Specifies the name of the transport class to which the outbound signalling paths are assigned. If you omit this keyword, the system uses the DEFAULT transport class. The classname must be previously defined. You must specify the *classname* as one to eight alphanumeric and national (\$,#,@) characters, and may begin with any of these characters.

POLICY or POL,TYPE=*name*,POLNAME or POLNM=*polname*

Specifies that the administrative policy named is to be made active in the sysplex or that the active policy is to be changed. The system from which the SETXCF command is issued must have access to the couple data set supporting the service.

See *z/OS MVS Setting Up a Sysplex* for information about transitioning to a new administrative policy.

TYPE=*name*

Specifies the *name* of the service that is using the couple data set for policy data. The supported services are:

- ARM for automatic restart management
- CFRM for coupling facility resource management
- SFM for sysplex failure management

POLNAME or POLNM=*polname*

Specifies the name of the administrative policy to be made active.

SETXCF Command

Note: When TYPE=ARM is specified, POLNAME is optional. If no policy name is specified, the policy defaults will be used.

REBUILD or RB,POPULATECF or POPCF=*cfname*

Specifies the name of the coupling facility that is to be populated with structures selected from the set of allocated structures in the active CFRM policy.

A structure rebuild will be attempted for each allocated structure in the policy that contains the specified coupling facility in its preference list, if the specified coupling facility is at a higher position in the preference list than the coupling facility in which the structure currently is allocated. If the structure is allocated in a more preferable coupling facility already, the rebuild will not continue.

POPULATECF rebuild processing assumes LOCATION=OTHER. LOCATION and LESSCONN options cannot be specified.

Each structure that contains the specified coupling facility at a higher position in its preference list will be processed serially to completion (either stopped or completed) before the next structure is selected. The serial nature of this processing allows even XCF signalling structures to be selected for coupling facility population.

The coupling facility name can be up to 8 alphanumeric characters long and must begin with an uppercase alphabetic character. The name can contain numeric characters, uppercase alphabetic characters, national characters (\$, @, #), or an underscore (_).

REBUILD or RB,DUPLEX,STRNAME or STRNM=(*strname*[,*strname*]...)

Specifies the name of one or more coupling facility structures that are to be duplexed in another coupling facility.

If user-managed structure duplexing is not supported for the target structure, the duplexing operation will not be started and the system issues a message to the operator.

Duplexing rebuild processing assumes LOCATION=OTHER and LESSCONN=TERMINATE. Other LOCATION and LESSCONN options cannot be specified.

The structure name can be up to 16 characters long and can contain numeric characters, uppercase alphabetic characters, national characters (\$, @, #), or an underscore (_). The name must begin with an uppercase alphabetic character. IBM names begin with SYS, or letters A-I.

If you specify only one structure name, you do not need to enter the parentheses.

REBUILD or RB,DUPLEX,CFNAME=(*cfname*[,*cfnane*]...)

Specifies the name of one or more coupling facilities for which all structures are to be duplexed in a different coupling facility.

The system attempts to start a duplexing operation for each structure that is currently allocated in the specified coupling facility.

If structure duplexing is not supported for a particular structure, the system issues a message to the operator.

The coupling facility name can be up to 8 alphanumeric characters long and can contain numeric characters, uppercase alphabetic characters, national characters (\$, @, #), or an underscore (_). It must begin with an uppercase alphabetic character.

If you specify only one coupling facility name, you do not need to enter the parentheses.

REBUILD or RB,STRNAME or STRNM=(strname[,strname]...)

Specifies the name of one or more coupling facility structures that are to be rebuilt in the same coupling facility or another coupling facility. The structure name can be up to 16 alphanumeric characters long and must begin with an uppercase alphabetic character. IBM names begin with SYS, or letters A-I. If you specify only one structure name, you do not need to enter the parentheses.

REBUILD or RB,CFNAME or CFNM=(cfname[,cfname]...)

Specifies the name of one or more coupling facilities for which all structures other than XCF signalling structures are to be rebuilt. The coupling facility name can be up to 8 alphanumeric characters long and must begin with an uppercase alphabetic character. If you specify only one coupling facility name, you do not need to enter the parentheses.

For any given structure, the system might not start rebuild. *z/OS MVS Programming: Sysplex Services Guide* lists the requirements for rebuild initiation. For example, if the named coupling facility contains one or more XCF signalling structures, the system does not start rebuild for them. To rebuild an XCF-signalling structure, issue the SETXCF START,REBUILD,STRNAME=... command for one structure at a time.

LOCATION=NORMAL or OTHER

Specifies the location where the new structure or structures can be rebuilt.

If you specify LOCATION=NORMAL, the new structure can be allocated in any coupling facility in the preference list, following the normal allocation rules. If you specify LOCATION=OTHER, the new structure cannot be allocated for rebuild in the same coupling facility as the original structure. The new structure can be allocated in any other coupling facility in the preference list, following the normal allocation rules.

Duplexing rebuild and POPULATECF processing assume LOCATION=OTHER.

Note that before the rebuild process begins, you might need to change the administrative policy to specify where the structure can reside and then activate the policy. The CFRM administrative policy contains the preference list that specifies coupling facilities where a structure can reside.

LESSCONN or LC=TERMINATE or CONTINUE

Specifies the action the system is to take when rebuilding the structure results in a new structure that has poorer connectivity relative to the set of active structure connectors than the old structure does.

- With LESSCONN=TERMINATE, the system stops the rebuild processing for the new structure if connectivity relative to the set of active connectors to the structure is not equal or better than it was to the current structure.

LESSCONN=TERMINATE is the default system action. This protects active connectors against inadvertently losing connectivity to the structure as a result of rebuilding the structure.

Duplexing rebuild processing assumes LESSCONN=TERMINATE.

- With LESSCONN=CONTINUE, the system allows the rebuild processing for the new structure even if connectivity relative to the set of active connectors to the structure is poorer than it was to the current structure.

SETXCF Command

Attention: Because this might cause active connectors to lose connectivity to the structure, do not use this keyword unless you understand the impact to the application or subsystem.

Some connectors stop the rebuild if a loss of connectivity is observed, but most connectors disconnect from the structure to allow the rebuild to complete. For many exploiters, disconnecting from the structure is likely to result in losing the sysplex-related functionality (for example, loss of data sharing capability) on that system. For critical system exploiters, this may result in a system wait state. See the application or subsystem documentation for recommendations.

ALTER,STRNAME or STRNM=*strname*,**SIZE=***size*

Specifies that structure alter processing is to be initiated.

STRNAME or STRNM=*strname*

Specifies the name of the coupling facility structure to be altered. You may specify only one structure name. The name can be up to 16 alphanumeric characters long and must begin with an uppercase alphabetic character. The name can contain numeric characters, uppercase alphabetic characters, national characters (\$,@,#) or an underscore(_). IBM names begin with SYS, or letters A-I.

SIZE=*size*

Specifies the target size of the structure to be altered. Specify *size* in units of 1K. Size is bounded by the minimum and maximum sizes determined when the structure was allocated. The minimum size of a structure is determined by the coupling facility; the maximum size of a structure is established by the installation in the CFRM policy. (Use the DISPLAY XCF,STRUCTURE,STRNAME=*strname* command to determine the maximum structure size allowed.)

REALLOCATE or REALLOC

Specifies that the REALLOCATE process is to be initiated.

The REALLOCATE process evaluates each allocated structure to recognize the need to make the following adjustments:

- Relocate the structure instance or instances
- Complete a pending policy change
- Trigger MVS-initiated duplexing

The evaluation of each allocated structure uses the XCF structure allocation algorithm and either the active or pending CFRM policy definition for the structure. Message IXC574I is written to the hardcopy log to show the current location of instances allocated in coupling facilities, the policy information used, and the evaluation result. Then it compares the current location with the location identified by evaluation to determine what if any adjustments are needed.

By evaluating the allocated structure, the REALLOCATE process can recognize that the structure is optimally located and immediately complete a change to the policy definition for the structure when the change does not affect the size. In addition, REALLOCATE processing is a triggering event for MVS-initiated duplexing rebuild when it finds the structure is not duplexed and DUPLEX(ENABLED) is specified for the structure in the CFRM policy.

The REALLOCATE process recognizes the need to relocate structure instances when one of the following conditions is detected:

- There is a change to the policy definition for the structure affecting the structure size or location.
- The structure is not optimally located.

To control the relocation of a specific structure by the REALLOCATE process, you can specify the ALLOWREALLOCATE(YESNO) keyword for the structure in the CFRM policy.

- If you specify or default to ALLOWREALLOCATE(YES) for the structure in the CFRM policy:

When the REALLOCATE process recognizes the need to relocate structure instances (see the preceding description of such conditions), the structure is selected as the target of the REALLOCATE process and the structure rebuild process is used to make the adjustments. Structure rebuild process supports the following:

- User-managed rebuild
- User-managed duplexing rebuild
- System-managed rebuild
- System-managed duplexing rebuild

Multiple steps may need to be taken to complete the relocation of a selected structure. The steps are accomplished using structure rebuild processing (for example, user-managed rebuild) to adjust the location or activate a pending policy change for the structure that is the target of the REALLOCATE process. Messages to the operator document the steps being taken for each structure that is examined.

- For a simplex structure, one step (rebuild) is used to adjust the location or to activate a pending policy change.
- For a duplexed structure, two or three steps are used. The first step stops duplexing and one or more subsequent steps are used as needed to adjust the location, activate a pending policy change, and to reduplex the structure. If a subsequent step cannot be started, the system issues message IXC546I with an explanation.
- If you specify ALLOWREALLOCATE(NO) for the structure in the CFRM policy:

The REALLOCATE process evaluates the allocated structure but does not select the structure as the target of the REALLOCATE process. However, when NO is specified it is still possible for the REALLOCATE process to do the following adjustments if applicable:

- Complete a pending policy update when the pending change does not affect the size or location.
- Trigger an MVS-initiated duplexing rebuild when DUPLEX(ENABLED) is specified for the structure and the structure is not duplexed.

When the REALLOCATE process does not select an allocated structure, message IXC544I is issued with an explanation.

When the start request is accepted, the DISPLAY XCF,STR or the DISPLAY XCF,CF command shows THE REALLOCATE PROCESS IS IN PROGRESS. For a summary of allocated structure status, use the DISPLAY XCF,STR,STATUS=ALLOCATED command:

- The structure that is the current target indicates TARGET OF REALLOCATE PROCESS.

SETXCF Command

- | • Allocated structures which have not been evaluated indicate REALLOCATE EVALUATION PENDING.
- | • Structures which have been processed do not have additional status indicators displayed but the log can be examined to determine the action taken.

When the entire process completes for all structures, the processing provides a report (message IXC545I) summarizing the actions that were taken as a whole. The REALLOCATE process evaluates all allocated structures, in a serial (one structure at a time) fashion. Each selected structure is processed to completion before the next structure is evaluated. The serial nature of this processing allows even XCF signalling structures to be selected for relocation.

REALLOCATE processing evaluates a structure based on the CFRM policy and on the current conditions (for example, available coupling facilities, coupling facility attributes, and connection attributes), and for each structure selected for processing, takes the necessary steps to adjust the location of the structure's allocated instances. From the time a structure is evaluated to the time when the steps using structure rebuild processing cause a new instance to be allocated, it is possible for the conditions to have changed. The result is that the current conditions are used when the structure allocation algorithm is applied. The REALLOCATE process does not validate the resulting location of the allocated instances but relies on the result of applying the XCF allocation criteria. Because of this, it is possible that the coupling facilities shown as preferred when message IXC574I was written to the hardcopy log with the evaluation information are not the coupling facilities containing the allocated instances when the necessary steps finish. Where REALLOCATE processing intersects with other environmental changes (for example, starting or stopping a structure rebuild process due to a policy change, a coupling facility failure, or loss of connectivity to a coupling facility), the other ongoing process will take precedence with REALLOCATE processing issuing messages IXC544I or IXC546I as appropriate. For some environmental changes (for example, a coupling facility failure), the installation may choose to stop the REALLOCATE process.

Consider the following when you use the SETXCF START,REALLOCATE command:

- Move structures out of a coupling facility following a CFRM policy change that deletes/changes that coupling facility (for example, in preparation for a coupling facility upgrade).
- Move structures back into a coupling facility following a CFRM policy change that adds/restores the coupling facility (for example, following a coupling facility upgrade/add).
- Clean up pending CFRM policy changes that may have accumulated for whatever reason, even in the absence of any need for structure "relocation" per se.
- Clean up simplex or duplexed structures that were allocated in or moved into the "wrong" coupling facilities, for whatever reason (for example, the "right" coupling facility was inaccessible at the time of allocation).
- Clean up duplexed structures that have primary and secondary "reversed" because of a prior condition which resulted in having duplexing stopped with KEEP=NEW and the structure reduplicated.

Notes:

1. The REALLOCATE process is mutually exclusive with the POPULATECF function, which can be started either by the SETXCF operator command or the IXLREBLD programming interface.
2. The REALLOCATE process can only be started or stopped using the SETXCF command.
3. Support for the REALLOCATE process is provided by APAR OA03481.
 - The REALLOCATE process cannot be started if there exists an active system in the sysplex that does NOT have the APAR installed. The SETXCF START,REALLOCATE command is rejected.
 - An in-progress REALLOCATE process is stopped immediately when an active system without the APAR installed is discovered in the sysplex. The SETXCF START,REALLOCATE command was accepted but subsequently an active system without the APAR installed was discovered by an up-level system which immediately stopped the process.

In both cases, message IXC543I is issued with explanatory text.
4. Enhancements for the REALLOCATE process are provided by APAR OA08688.
 - A structure-level control is provided to prevent REALLOCATE processing from selecting particular structures while allowing it to initiate structure rebuilds for others. This control is achieved by providing an option on the CFRM STRUCTURE parameter to the XCF Administrative Policy Utility (IXCMIAPU). The option is ALLOWREALLOCATE. See *z/OS MVS Setting Up a Sysplex*, Appendix C under section Coding the Administrative Data Utility for topic CFRM Parameters for Administrative Data Utility.
 - For a simplex structure with DUPLEX(ENABLED) specified, the REALLOCATE process is a triggering event. Message IXC536I is issued when MVS is able to initiate a duplexing rebuild for the structure identified by REALLOCATE processing.
 - When a pending policy change does not affect size, REALLOCATE processing avoids structure rebuild processing when evaluation with the pending policy shows that the structure does NOT need relocating. It completes the pending change. Message IXC544I is issued indicating ALLOCATED IN PREFERRED CF AND POLICY CHANGE MADE.
 - With this enhancement, REALLOCATE processing will ignore a specified exclusion list in some cases, so as to avoid anomalies when honoring the exclusion list would have precluded structures from relocating to the optimal location.

SETXCF STOP Command

Use the SETXCF STOP command to:

- Stop one or more inbound signalling paths
- Stop one or more outbound signalling paths
- Delete the definition of a transport class
- Stop using an administrative policy
- Stop rebuilding one or more coupling facility structures
- Stop populating a coupling facility that had been newly brought into service in a sysplex with structures selected from the set of those defined in the active CFRM policy.

SETXCF Command

- Stop user-managed duplexing of one or more structures in a coupling facility and specify the structure that is to remain in use.
- Stop altering a coupling facility structure.
- Stop an in-progress REALLOCATE process.
- Transition from a message-based event and confirmation management protocol to a policy-based event and confirmation management protocol.

The complete syntax for the SETXCF STOP command is:

```
SETXCF STOP,{PATHIN,{DEVICE=([/]indevnum[,/]indevnum]...) }      }
              {STRNAME=(strname[,strname]...) }                         }
              [,UNCOND=NO|YES]

{PATHOUT,{DEVICE=([/]outdevnum[,/]outdevnum]...) }   }
              {STRNAME=(strname[,strname]...) }                         }
              [,UNCOND=NO|YES]

{CLASSDEF,CLASS=classname}                                }

{MSGBASED}
{POLICY,TYPE=name  }

{REBUILD,{POPULATECF=cfname}                           }
              {DUPLEX,}
              {STRNAME=(strname[,strname]...) }
              {,KEEP=NEW|OLD}
              {CFNAME=(cfname[,cfname]...) }                         }

{ALTER,STRNAME=strname}                               }
{REALLOCATE,[,FORCE]}                                }
```

ALTER,STRNAME or STRNM=strname

Specifies that structure alter processing should be stopped for the named structure. (Use the DISPLAY XCF,STRUCTURE command to determine which structures are being altered.)

The SETXCF STOP,ALTER command can be used to stop an alter in progress that was initiated either with a SETXCF START,ALTER command or with the IXLALTER programming interface.

STRNAME or STRNM=strname

Identifies the structure for which to stop structure alter processing.

CLASSDEF or CD,CLASS=classname

Deletes the definition for the specified transport class and frees the space allocated for message buffers. The transport class must be defined to XCF. You can delete a transport class only when no signalling paths are currently assigned to the class. You cannot delete the DEFAULT transport class definition.

MSGBASED

Switch to a policy-based protocol from the current message-based event and confirmation management protocol.

PATHIN or PI,DEVICE=([/]indevnum[,/]indevnum]...)

Specifies the device number of one or more inbound signalling paths that XCF should no longer use. A device number, *indevnum*, is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). If you specify only one device, you do not need to enter the parentheses.

A specified device must be defined to XCF as an inbound path.

Once the command completes successfully, XCF stops receiving signalling traffic along these paths and deallocates the device. The device is no longer defined to XCF as a signalling path. If a specified signalling path is the last path from another system in the sysplex, XCF rejects the command.

PATHIN or PI,STRNAME or STRNM=(*strname*[,*strname*]...)

Specifies the name of one or more coupling facility structures that XCF should no longer use. The specified structure must be defined to XCF as an inbound path.

The structure name can be up to 16 alphanumeric characters long and must begin with an uppercase alphabetic character. If you specify only one structure name, you do not need to enter the parentheses.

UNCOND=NO or YES

UNCOND=NO stops a signalling path when no other stop commands are active against that path. UNCOND=YES stops a signalling path even when another stop command is active against that path. This operand stops an outstanding stop request and initiates a new one, recovering path resources so they are not lost for the duration of the IPL. Because a STOP command with the UNCOND=YES option might cause loss of signals, the system accepts the command only when an outstanding stop is active against a path.

Note: Use UNCOND=YES only at the direction of the system programmer.

PATHOUT or PO,DEVICE=([/]*outdevnum*[,(/]*outdevnum*]...)

Specifies the device number of one or more outbound signalling paths that XCF should no longer use to send messages. A device number, *outdevnum*, is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). If you specify only one device, you do not need to enter the parentheses.

XCF stops sending signalling traffic along these paths, leaving the devices unallocated. The device is no longer defined to XCF as a signalling path. If a specified path is the last path to another system in the sysplex, XCF rejects the command.

PATHOUT or PO,STRNAME or STRNM=(*strname*[,*strname*]...)

Identifies one or more coupling facility structures that XCF should no longer use. The specified structure must be defined to XCF as an outbound path.

The structure name can be up to 16 alphanumeric characters long and must begin with an uppercase alphabetic character. If you specify only one structure name, you do not need to enter the parentheses.

UNCOND=NO or YES

UNCOND=NO stops a signalling path when no other stop commands are active against that path. UNCOND=YES stops a signalling path even when another stop command is active against that path. This operand stops an outstanding stop request and initiates a new one, recovering path resources so they are not lost for the duration of the IPL. Because a STOP command with the UNCOND=YES option could cause loss of signals, the system accepts the command only when an outstanding stop is active against a path.

Note: Use UNCOND=YES only at the direction of the system programmer.

POLICY or POL,TYPE=*name*

Directs the system to deactivate the active policy for the type of service named.

SETXCF Command

TYPE=name

Specifies the name of the service that is using the couple data set for policy data. The supported service names are:

- ARM for automatic restart management
- CFRM for coupling facility resource management
- SFM for sysplex failure management

REBUILD or RB,POPULATECF or POPCF=cfname

Specifies the name of the coupling facility in which structure population is to stop. All structure rebuilds that were initiated by a SETXCF START,REBUILD,POPULATECF command will be stopped.

Note that you also can use the SETXCF STOP,REBUILD,CFNAME or SETXCF STOP,REBUILD,STRNAME to stop structure rebuilds that were initiated by a SETXCF START,REBUILD,POPULATECF command.

REBUILD or RB,DUPLEX,STRNAME or STRNM=(strname[,strname]...)

Specifies the name of one or more coupling facility structures for which duplexing is to be stopped. You must also specify with the KEEP keyword which of the duplexed structures should remain after the duplexing operation has stopped.

The structure name can be up to 16 alphanumeric characters long and must begin with an uppercase alphabetic character. IBM names begin with SYS, or letters A-I. If you specify only one structure name, you do not need to enter the parentheses.

KEEP=NEWOLD

Specifies which of the duplexed structures should remain after duplexing has stopped.

KEEP=NEW specifies that processing should switch to the new structure.

KEEP=OLD specifies that processing should fall back to the old structure.

Note: If the CFRM active policy specifies that the structure is DUPLEX(ENABLED), it is possible that the system will attempt to duplex the structure again immediately after the completion of the SETXCF STOP processing. To avoid this, change the structure's DUPLEX setting in the CFRM policy to DUPLEX(ALLOWED) before initiating the SETXCF STOP or to DUPLEX(DISABLED) which will cause XES to initiate the stop processing. The structure's DUPLEX setting can be changed back to DUPLEX(ENABLED) when required.

REBUILD or RB,DUPLEX,CFNAME=(cfname[,cfname]...)

Specifies the name of one or more coupling facilities in which structure duplexing is to stop.

Duplexing will be stopped for each structure in each specified coupling facility so that no structures involved in structure duplexing processing remain in the coupling facility. The structures are processed serially.

- If the specified coupling facility contains the new structure in the duplexed pair of structures, the system will fall back to the old structure.
- If the specified coupling facility contains the old structure in the duplexed pair of structures, the system will switch to the new structure.

The coupling facility name can be up to 8 alphanumeric characters long and must begin with an uppercase alphabetic character. If you specify only one coupling facility name, you do not need to enter the parentheses.

Note: If the CFRM active policy specifies that a structure in the coupling facility for which SETXCF STOP processing is requested is DUPLEX(ENABLED), it is possible that the system will attempt to duplex the structure again immediately after the completion of the SETXCF STOP processing. To avoid this, change the structure's DUPLEX setting in the CFRM policy to DUPLEX(ALLOWED) before initiating the SETXCF STOP processing or to DUPLEX(DISABLED) which will cause XES to initiate the stop processing. The structure's DUPLEX setting can be changed back to DUPLEX(ENABLED) when required.

REBUILD or RB,STRNAME or STRNM=(*strname*[,*strname*]...)

Specifies the name of one or more coupling facility structures for which rebuild processing is to stop. The structure name can be up to 16 alphanumeric characters long and must begin with an uppercase alphabetic character. IBM names begin with SYS, or letters A-I. If you specify only one structure name, you do not need to enter the parentheses.

REBUILD or RB,CFNAME or CFNM=(*cfname*[,*cfname*]...)

Specifies the name of one or more coupling facilities for which rebuild processing is to stop for all structures. The coupling facility name can be up to 8 alphanumeric characters long and must begin with an uppercase alphabetic character. If you specify only one coupling facility name, you do not need to enter the parentheses.

ALTER,STRNAME or STRNM=*strname*

Specifies that structure alter processing should be stopped for the named structure. (Use the DISPLAY XCF,STRUCTURE command to determine which structures are being altered.)

The SETXCF STOP,ALTER command can be used to stop an alter in progress that was initiated either with a SETXCF START,ALTER command or with the IXALTER programming interface.

STRNAME or STRNM=*strname*

Identifies the structure for which to stop structure alter processing.

REALLOCATE[,FORCE] or REALLOC[,FORCE]

Specifies that an in-progress REALLOCATE process is to be stopped.

When stopping without specifying FORCE, REALLOCATE processing completes the steps for the current target structure then finishes. The status of the REALLOCATE processing will be "STOPPING" as shown by either the DISPLAY XCF,STR or the DISPLAY XCF,CF operator command.

When stopping with FORCE specified, REALLOCATE processing finishes immediately AND the step(s) for the current target structure might NOT be completed. Use the FORCE option when structure rebuild processing for the structure which is the target of the REALLOCATE process is not making progress.

When the process finishes, for the structures selected prior to the operator stopping the process, the processing provides a report (message IXC545I) summarizing the actions that were taken up to the time that processing was stopped.

To stop the REALLOCATE process does NOT require issuing the command without FORCE specified before issuing with FORCE specified.

SLIP Command

The SLIP command controls SLIP (serviceability level indication processing), a diagnostic aid that intercepts or traps certain system events and specifies what action to take. Using the SLIP command, you can set, modify, and delete SLIP traps.

Table 4-38. Summary of the SLIP Command

Command:	Topic:
SLIP SET	"Setting a SLIP Trap" on page 4-513
SLIP MOD	"Modifying an Existing SLIP Trap" on page 4-573
SLIP DEL	"Deleting an Existing SLIP Trap" on page 4-574

General information about the SLIP command is in the following topics:

- "Syntax"
- "Using SLIP Commands"
- "Processing of SLIP Commands" on page 4-507
- "Coding SLIP Command Parameters" on page 4-508

Syntax

The following introductory syntax gives you an overview of the entire command.

SLIP SET[,options],END	Command for an error event trap (non-PER)
SLIP SET,IF[,options],END	Command for an instruction fetch PER trap
SLIP SET,SBT[,options],END	Command for a successful branch PER trap
SLIP SET,SA SAS[,options],END	Commands for a storage alteration PER trap
SLIP MOD[,options]	Command to modify an existing trap
SLIP DEL[,options]	Command to delete an existing trap

Notes:

- You must specify SET, MOD, or DEL immediately following SLIP.
- If you specify IF, SBT, SA, or SAS, it must immediately follow SET.
- You must specify END at the end of all SLIP SET commands.

More detailed syntax is presented in the following:

- "Syntax for an Error Event SLIP SET Command" on page 4-519
- "Syntax for an Instruction Fetch or Successful Branch SLIP SET PER Command" on page 4-520
- "Syntax for a Storage Alteration SLIP SET PER Command" on page 4-521
- "Syntax for the ACTION Parameters for the SLIP SET Command" on page 4-522
- "Modifying an Existing SLIP Trap" on page 4-573
- "Deleting an Existing SLIP Trap" on page 4-574

Using SLIP Commands

Use a SLIP command only at the direction of the system programmer. You can enter a SLIP command:

- On a console with MVS master authority.
- On a TSO terminal in OPERATOR mode.
- In a TSO CLIST.

In the CLIST, use the line continuation character at the end of each line and the END parameter at the end of the last line.

- In an IEACMD00, COMMNDxx, or IEASLPxx parmlib member.

While you can enter a SLIP command in any of these members, IBM recommends that you place your SLIP commands in IEASLPxx and enter a SET SLIP=xx command to activate the member. IEACMD00 and COMMNDxx require that a command be on a single line. Also, SLIP may process commands in IEACMD00 and COMMNDxx in any order, but processes commands in IEASLPxx in the order in which they appear.

SLIP Traps in Systems in a Sysplex

For a sysplex containing similar systems, certain problems might require identical SLIP traps on those similar systems. To set up these traps, do the following:

1. Assign similar names to identical jobs on different systems. The names should form a pattern, such as JOB1, JOB2, JOB3, and so on.
2. Create one IEASLPxx member containing the trap you need for the problem.

Use a REMOTE parameter in the SLIP command so that, the first time a trap matches on a system, the action will also be taken on other systems in the sysplex. For example, the SLIP command could request a dump on its system and, through REMOTE, on all the similar systems.

Use an IDGROUP parameter so that, after the match, the identical traps on the other systems will be disabled.

Use wildcards in parameters so that the command will process in all systems in the sysplex. For example, JOB? would indicate JOB1, JOB2, JOB3, and so on.

3. Place the member in the shared parmlib data set or in the parmlib data set for each of the similar systems.
4. In systems using JES2 or JES3, activate the member or members with the following command entered on one of the systems:

```
ROUTE *ALL,SET SLIP=xx
```

If only some systems in the sysplex are similar, use a ROUTE command specifying a named subset of systems; see "ROUTE Command" on page 4-367 for details.

When a SLIP trap results in SVC dumps from multiple systems, each dump contains the same incident token. You can use the incident token to correlate the multiple dumps to one problem.

Processing of SLIP Commands

For an error event, the system gives control to SLIP before giving control to ESTAE or FRR recovery routines. This sequence allows SLIP to capture information before recovery routines change it, thus providing the advantage of working with the original problem data.

When you have defined more than one SLIP trap and SLIP gets control, SLIP first examines the most-recently defined trap. If SLIP does not find a match condition, it proceeds to the previously defined trap.

Any SLIP trap affects system performance, but PER traps can have a measurable effect on performance. Therefore, use conditions to filter the events being checked for matches, especially for PER traps. Improper use of PER traps can cause severe performance problems. See "Setting Effective SLIP PER Traps" on page 4-515.

Coding SLIP Command Parameters

The parameters can contain:

- Wildcard characters. See “Using Wildcards in Commands” on page 1-17.
- Indirect addresses.

Indirect Addresses

An indirect address is the address of a location or a general purpose register that contains another address. You can use indirect addressing with the following SLIP command parameters: DATA, LIST, REFAFTER, REFBEOF, RANGE, SUMLIST and TRDATA.

Note: Indirect addresses used with SLIP are similar to those used with the TEST command in TSO except that:

- Unlimited levels of indirect addressing are permitted.
- Symbols are not used.
- Absolute addresses are not followed by a period.
- Address modifiers must be hexadecimal.

The elements of an indirect address used by SLIP are:

1. A **direct address**, which consists of 1 to 8 hexadecimal digits optionally followed by one or more displacements.
2. A **32-bit register** (or the low half of a 64-bit register), in the form nR, where **n** is a decimal number from 0 to 15.
3. A **64-bit register**, in the form nG, where **n** is a decimal number from 0 to 15. Note, if the G suffix is used on an ESA/390 system, it is translated to R.
4. **Symbolics**, which include the following forms:
 - BEAR (breaking event address register). Use the BEAR symbolic to access the address from where the last successful branch occurred, before the event that caused the SLIP action processor to receive control. When SLIP receives control for a MEMTERM, the value of the BEAR symbolic is 0.
 - BPER (beginning PER range). Use the BPER symbolic wherever an indirect address is used to refer to the beginning PER range. The BPER symbolic is most useful when using dynamic PER traps where the PER range of a subsequently activated trap is unknown when the traps are set.

Note: When using REFBEOF or REFAFTER, the value of BPER cannot be changed. For example, REFBEOF (BPER,EQ,01) is not allowed.

5. An **indirection indicator**, which can be a percent sign (%), a question mark (?), or an exclamation point (!). The indirection indicator says that the information at the direct address or in the register is a fullword pointer to the data. A percent sign means that the pointer is a 24-bit address. A question mark means that the pointer is a 31-bit address. An exclamation point means that the pointer is a 64-bit address.
6. A **displacement**, which begins with a plus or minus sign and consists of 1 to 4 hexadecimal digits. The maximum displacement allowed is 7FFF.

In the expression

128%+4%+8%+C

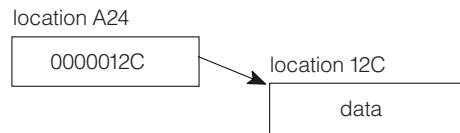
128 is a direct address.
% signs indicate 24-bit indirect addressing.
+4, +8, and +C are displacements.

In the expression

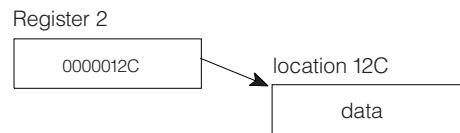
2R??+4?+8?+C

2R is the register (general purpose register 2).
 ?? and ? indicate 31-bit indirect addressing.
 2R?? is equivalent to 2R?+0?.
 +4, +8, and +C are displacements.

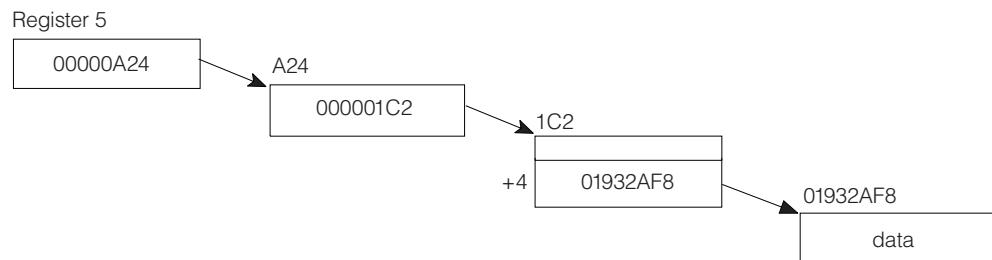
To refer to data when the address of the data is located at A24, specify A24?. Graphically:



To refer to data when the address of the data is in general purpose register 2, specify 2R%. Graphically:



You can indicate as many levels of indirect addressing as necessary by following the initial indirect address with a corresponding number of percent signs or question marks. You can also include plus or minus displacement values. For instance, you can specify 5R%?+4?. Graphically:

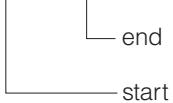


Shorthand Method for Specifying Addresses: When specifying more than one address in a SLIP parameter, you can use a shorthand method to specify any address after the first. The first address is written out completely and defines the base. The base consists of everything in the first address **except the last displacement**. (If no displacement is specified, SLIP assumes a displacement of zero.) When you specify subsequent addresses, you can omit the base.

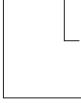
The following example shows two ways of specifying a range from 2R%+4 through 2R%+7. In the shorthand method, the base is 2R% and SLIP adds the base to the displacement you specify.

SLIP Command

Regular Method

2R%+4,2R%+7

start
end

Shorthand Method

2R%+4,+7

end (shorthand method omits the base and specifies only the last displacement)
start

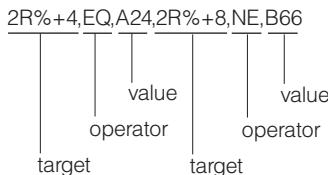
When you are using shorthand to specify an address range, make sure that the ending address is greater than the starting address. Because SLIP does not consider the base to include the last displacement, the displacement you specify for the ending address must be greater than the displacement you specify for the starting address.

For example, **it would be incorrect** to specify a range as $2R\%+4,+3$. An error condition exists because the ending address is less than the starting address.

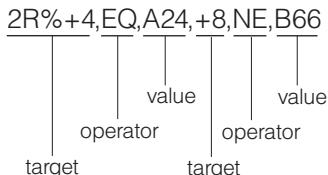
Indirect Addressing with the DATA, REFBEFOR, and REFAFTER Parameters:

The following discussion applies to the DATA, REFBEFOR and REFAFTER parameters when you specify multiple targets and use base/displacement shorthand.

The first direct or indirect address is written out completely and establishes the base. Subsequent addresses are written as plus or minus displacements from the base. For example,

2R%+4,EQ,A24,2R%+8,NE,B66

target value target value
operator operator

is written using shorthand form as

2R%+4,EQ,A24,+8,NE,B66

target value target value
operator operator

You must establish a direct or indirect base address before using displacements. In the following example, 2R specifies the contents of general purpose register 2, not an address. Therefore, the example using a +6 displacement is incorrect.

2R,EQ,C12,+6,NE,D01

Indirect Addressing with the LIST, RANGE, SUMLIST and TRDATA

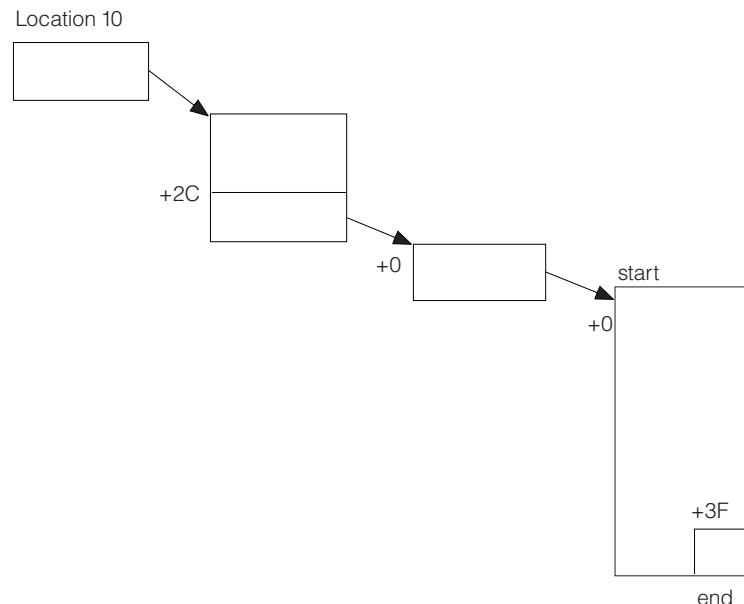
Parameters: When indirect addressing is used with the LIST, SUMLIST, RANGE or TRDATA parameters, the indirect addresses point to the beginning and end of a field of data. The following example shows a starting and ending address:

10%+2C%%,10%+2C%%+3F
 start end

The same example in shorthand is:

10%+2C%%,+3F
 start end

Graphically:



Qualifying Direct or Indirect Addresses to Address Spaces: You can qualify direct or indirect addresses in the DATA, REFBEOF, REFAFTER, LIST, RANGE, SUMLIST, and TRDATA parameters of the SLIP command. The qualifier can be the address space name or the job name of the job associated with the address space. If you omit an address space or job name qualifier, SLIP processing uses the current address space. The formats of the qualifiers are:

asid.addr
 'jobname'.addr

asid

Is an explicit or symbolic address space qualifier. An explicit *asid* is a 1- to 4-digit hexadecimal ASID number. A symbolic *asid* is one of the following:
 CURRENT or CU Current address space

SLIP Command

HASID or H	Home address space
I	Address space where the instruction executed
PASID or P	Primary address space
SA	Current alteration space used by an SA or SAS trap
SASID or S	Secondary address space

jobname

Is the job name that is associated with the address space. A *jobname* can be 1 to 8 alphanumeric and national (\$, #, @) characters and is enclosed in single quotes. You can specify wildcards in *jobname* with the following exception: an * must be a suffix and cannot appear alone.

See “Using Wildcards in Commands” on page 1-17. When a *jobname* with wildcards qualifies an address, the system selects one job, whose corresponding address space has the lowest address space identifier (ASID).

addr

Is either a direct address of 1 to 8 hexadecimal digits or an indirect address.

Example: To list 8 bytes of data from address space 3 and 32 bytes of data from the primary address space at the time of interrupt or error, enter:

LIST=(3.FC210,3FC217,P.3R%,+1F)

Indirect Addressing Using a Register: When you specify a register, SLIP normally uses the contents of the general purpose register to calculate an address. It uses the address space or data space associated with the related access register when all the following conditions are true:

- CURRENT is specified, or defaulted to
- The processor is in access register (AR) ASC mode
- The indirect address starts with a register indirection.

SLIP will continue to use that space until changed explicitly by a qualifier.

CURRENT can reset the space by negating the space previously found through an access register. The LIST value shown in the following example provides three storage ranges, in pairs. In AR ASC mode, the first two ranges are associated with access register 3. The third range is associated with location 8000 in the primary space.

When no register indirection is specified, SLIP will use the current address space.

LIST=(CU.3R%,+3,+6,+9,CU.8000,+4)

The LIST parameter value shown in the following example, also provides three storage ranges. In AR ASC mode the first range, 3R% to 4R%, is associated with the space indicated by access register 3. The second range, 5R% to 6R%, is also associated with the space indicated by access register 3.

The associated space changes only when the symbolic CURRENT is explicitly specified for the third pair, 7R% to 8R%. The symbolic CURRENT in this example changes the associated space to that indicated by access register 7.

LIST=(CU.3R%,4R%,5R%,6R%,CU.7R%,8R%)

You can use the BEAR symbolic to capture data about a wild branch as follows:

SLIP SET,C=0C1,DATA=(BEAR,EQ,nn),A=SVCD,E

In this example, if SLIP were entered to process an ABEND 0C1 and if the last successful branch occurred from *nn* (for example a possible branch to low storage from *nn*), an SVC dump is to be taken. The value of the BEAR symbolic is contained within SDWABEA in the dump.

You can use the BPER symbolic in conjunction with a dynamic PER trap where the range for an activated trap can be unknown.

```
SLIP SET,IF,LPAMOD=(MYMOD,60),A=TARGETID,TI=TRP2,ID=TRP1,E  
SLIP SET,SA,A=SVCD,RA=(1R?),DATA=(BPER?,EQ,00),ID=TRP2,E
```

In this example, these two traps constitute a dynamic PER activation chain. The author is interested in taking an SVC dump when the first byte of the storage buffer that register 1 points to at offset 60 into MYMOD becomes zero. Without the use of the symbolic, you can not reference the address of this buffer in the trap.

Setting a SLIP Trap

The SLIP commands used to set SLIP traps are described in the following topics:

- “Structure of a SLIP SET Command”
 - Events
 - Conditions
 - Actions
 - Types of SLIP SET Parameters
- “Setting Effective SLIP PER Traps” on page 4-515
 - Keeping PER Traps from Slowing System Performance
 - PSWs Disabled for PER
 - Ignored PER Traps
 - Recursive Traps
 - PER Monitoring and Checkpoint/Restart
 - Dynamic PER Traps
- “Syntax for an Error Event SLIP SET Command” on page 4-519
- “Syntax for an Instruction Fetch or Successful Branch SLIP SET PER Command” on page 4-520
- “Syntax for a Storage Alteration SLIP SET PER Command” on page 4-521
- “Syntax for the ACTION Parameters for the SLIP SET Command” on page 4-522
- “SLIP SET Parameters” on page 4-528

Structure of a SLIP SET Command

In SLIP SET traps, you can indicate what kinds of **events** you want trapped and the system **conditions** for the trap, then specify what **action** the system is to take when the event occurs during the specified conditions.

Events: The kinds of events you can intercept are:

- **Error event:** This is also called a *non-PER* event. The trap is set by the command:

```
SLIP SET[,options],END
```

The error events are:

- An ABEND macro issued by a task
- Dynamic address translation error
- Software error caused by a machine check
- Abnormal end of an address space
- Paging I/O error
- Program check interruption
- Restart interruption

SLIP Command

- SVC error

Note: SLIP does not trap errors that are intercepted by SPIE or ESPIE routines.

- **Program event recording (PER) event:** The PER events are:

- **Instruction fetch:** The trap is set by the command:

SLIP SET,IF[,options],END

- **Successful branch:** The trap is set by the command:

SLIP SET,SBT[,options],END

- **Storage alteration:** The trap is set by one of these commands:

SET SET,SA[,options],END

SET SET,SAS[,options],END

Notes:

1. The parameters IF, SA, SAS, and SBT are positional. If you specify any one of them, it must directly follow a comma immediately after SLIP SET.
2. It is not possible to set a SLIP trap for the storage alteration of a hyperspace.

Conditions: The error and PER events you can trap are quite general, and you probably would not want to take an action each time such an event occurs. To narrow the scope of SLIP processing, you can qualify the event by requesting exactly what condition the system must be in when the error or PER event happens in order for the action to occur. The system checks each specified condition to see if it corresponds to the system condition at the time of the error or PER interruption.

The conditions you specify serve as filters to screen out those events in which you are not interested. A **match** for the trap occurs when the specified conditions are the same as the system conditions. A **no-match** occurs when the specified conditions are not the same as the system conditions. Only when all the conditions you specify match the system conditions will your action be taken.

Among the conditions you can specify are:

- The system mode at the time of the error or PER interruption
- A user or system completion code and reason code associated with an error
- The name of a job that must be in control at the time of the error or PER interruption
- The name of the job step program that must be in control at the time of the error or PER interruption
- The module name, entry point name, or address range where the error or PER interruption must occur
- The address space that must be in control at the time of the error or PER interruption
- The contents of specific storage locations and/or registers at the time of the error or PER interruption

If you omit a particular condition, the system does not check for that condition.

Actions: When one of these events occurs, you can take one of the following actions:

- Request an SVC dump tailored specifically to your needs
- Cause a system trace record to be written (PER only)
- Cause a generalized trace facility (GTF) trace record to be written
- Cause a logrec record to be written

- Put the system in a wait state
- Suppress system or problem program dumps (for error events only)
- Cause the recovery routines of the interrupted program to get control (PER only)
- Ignore the event

You can also request an additional action to be taken before or after the main action.

Types of SLIP SET Parameters: SLIP SET parameters are in five functional groups: event, condition, action, trap control, and specialized.

Table 4-39. Summary of the SLIP SET parameters

Function:	Parameters:		
Event parameters: Indicate the event to be monitored and trapped.	IF SAS SBT		
Condition parameters: To narrow the scope of the trap, the condition parameters specify system conditions that qualify the event.	ADDRESS ASID ASIDSA COMP DATA DSSA ERRTYP	JOBNAME JSPGM LPAEP LPAMOD MODE NUCEP NUCMOD	PSWASC PVTEP PVTMOD RANGE REASON
Action parameters: These parameters specify what the system is to do when the trap matches.	IGNORE NODUMP NOSUP NOSVCD NOSYSA NOSYSM NOSYSU	RECORD RECOVERY REFAFTER REFBEFOR STDUMP STRACE SUBTRAP	SVCD SYNCVCD TARGETID TRACE TRDUMP WAIT
	The following parameters are options to tailor the action that the system is to do.		
	ASIDLST DSPNAME GTFID JOBLIST	LIST REMOTE REFAFTER REFBEFOR	SDATA STRLIST SUMLIST TARGETID
Trap control parameters: These parameters control the operation of the trap.	DISABLE ENABLE	IDGROUP MATCHLIM	PRCNTLIM
Specialized parameters:	DEBUG END	ID OK	RBLEVEL

Setting Effective SLIP PER Traps

This topic describes where to place SLIP PER traps and how to keep SLIP PER traps from affecting system performance.

Note: Only one PER trap with an action other than IGNORE can be eligible for checking at any one time. *But see “Multiple PER Traps” on page 4-516.*

Keeping PER Traps from Slowing System Performance: For PER traps, limit PER monitoring to minimize slowing of the system:

- To reduce the range of storage monitored by the PER hardware:
 - For instruction fetch or successful branch PER traps, use the value in the LPAEP, LPAMOD, NUCEP, NUCMOD, PVTEP, PVTMOD, or RANGE parameter
 - For storage alteration PER traps, use the RANGE value

Limiting the range avoids processing unnecessary PER interrupts.

SLIP Command

- For non-IGNORE PER traps, use the JOBNAME parameter to limit PER monitoring to the address spaces in which the specified job runs. Use JOBNAME rather than ASID so that SLIP does PER monitoring for the job, even if some of the work runs in an address space other than the one in which the job was dispatched.

But, if a non-IGNORE PER trap might produce PER interrupts in an undesired address space, do the following:

- Use the ASID parameter to limit PER monitoring to the address space(s) identified on the parameter.
- Use MODE=HOME to request PER monitoring only when the unit of work runs in the address space in which it was dispatched.
- If a PER trap will produce PER interrupts for only one job or in only the desired address spaces, even if PER is active in all address spaces, perhaps because of the range specified, then do **not** specify an ASID parameter or MODE=HOME.

If you do not take measures to limit SLIP's use of system resources, your system may encounter performance problems. Use a monitoring product, such as RMF, to monitor the amount of system resources SLIP is using.

Note: The SLIP PER support is designed not to disrupt processing, even though this design means that a trap might fail to collect data or might not cause a requested action.

PSWs Disabled for PER: Certain processing cannot tolerate PER interrupts. For that processing, the PSW PER bit is set off to prevent interrupts. PER is disabled in the new PSWs for:

- Program check
- Machine check
- Restart

PER remains disabled in such critical paths until processing reaches a point where a PER interrupt can be accepted.

Ignored PER Traps: SLIP ignores — that is, does not process — PER events if:

- The PER interrupt occurred while DAT was off. SLIP PER support applies only to virtual addresses.
- The PER interrupt is redundant. See *Principles of Operation* for a description of redundant PER interrupts.
- The PER interrupt occurred, but an enabled non-IGNORE PER trap does not exist. SLIP does the following:
 - Ignores a PER interrupt caused by a non-SLIP tool that set up the PER control registers.
 - Turns off the PER bit in the resume PSW before returning to the first level interrupt handler (FLIH) for program checks. When the PER bit is off, the SLIP trap will not match.

Multiple PER Traps: It is not necessary to set SLIP traps individually and run a failing job multiple times, using one trap for each execution until a dump is taken. You can set SLIP PER traps at multiple points in a load module as follows: use a non-IGNORE PER trap to monitor the range that encompasses all of the points in which you are interested, followed by several IGNORE PER traps to prevent the SLIP action from being taken on the intervening instructions in which you are not interested. For example:

```

SLIP SET,IF,DISABLE,ACTION=SYNCSVCD,LPAMOD=(load_module_name,58CA4,85440),ID=JW01,
      JOBNAME=jjjjjjjj,END
SLIP SET,IF,DISABLE,ACTION=IGNORE,LPAMOD=(load_module_name,58CA5,5C80B),ID=JW02,END
SLIP SET,IF,DISABLE,ACTION=IGNORE,LPAMOD=(load_module_name,5C80D,5D0B3),ID=JW03,END
SLIP SET,IF,DISABLE,ACTION=IGNORE,LPAMOD=(load_module_name,5D0B5,5DD9D),ID=JW04,END
SLIP SET,IF,DISABLE,ACTION=IGNORE,LPAMOD=(load_module_name,5DD9F,8543F),ID=JW05,END
SLIP MOD,EN,ID=JW0*

```

In the above example, SLIP ID=JW01 would be set (disabled) first, followed immediately by SLIP IDs JW02, JW03, JW04, and JW05, all of which would also be set disabled. The final SLIP command (SL MOD,EN, ID=JW0*) would then enable all five of the SLIPs, but in reverse order, which is exactly what is required. Thus, the messages issued would be:

```

IEE727I SLIP TRAP ID=JW01 SET
IEE727I SLIP TRAP ID=JW02 SET
IEE727I SLIP TRAP ID=JW03 SET
IEE727I SLIP TRAP ID=JW04 SET
IEE727I SLIP TRAP ID=JW05 SET

IEE727I SLIP TRAP ID=JW05 ENABLED
IEE727I SLIP TRAP ID=JW04 ENABLED
IEE727I SLIP TRAP ID=JW03 ENABLED
IEE727I SLIP TRAP ID=JW02 ENABLED
IEE727I SLIP TRAP ID=JW01 ENABLED

```

When the failing job was then executed (only one execution would be necessary) a PER (hardware) interruption would be taken on the execution of every instruction in the specified range (58CA4,85440). At each of those interruptions, the software PER routines would gain control; they would run the chain of enabled SLIP traps to see if there were any IGNORE traps encompassing the specific address at which that particular interruption had occurred.

If the system *does* find an IGNORE trap, PER processing for that interruption would then be complete and control would return to the application program to continue executing. However, if the system *does not* find an IGNORE trap, it will take the action specified in the non-IGNORE trap (JW01). This is a dump at exactly the desired point, that is, at the instruction beginning at one of the locations 58CA4, 5C80C, 5D0B4, 5DD9E, or 85440.

For convenience you can enter all of these SLIP traps in an IEASLPxx member and then set (SET SLIP=xx) to that member. That way, if you've made an error in one of the SLIPs you need only correct that one error, add five **SLIP DEL, ID=JWxx** statements to the beginning of the IEASLPxx member, and then reset to that member. This process is easier than reentering every SLIP trap from the console. If you do this, the above IEE727I message would be preceded by

```
SET SLIP=xx
```

```
IEE252I MEMBER IEASLPxx FOUND IN PARMLIB
```

and followed by

```
IEE536I SLIP      VALUE xx NOW IN EFFECT
```

Recursive Traps: Do not specify a recursive trap, that is, do not place a PER trap in a function and then specify an action that causes SLIP to use the function. Some system services that SLIP uses check for recursion and prevent it.

For example, suppose a SLIP trap is placed in generalized trace facility (GTF) entry code and the trap specifies ACTION=TRACE. When the trap matches, GTF does not write a trace record because of the recursive checks within GTF.

SLIP Command

A similar situation exists with other trace actions, dump actions, and wait. In general, recursions result in the action not being taken. Avoid recursions by choosing an appropriate SLIP action.

PER Monitoring and Checkpoint/Restart: Checkpoint/restart does not include support for SLIP PER monitoring. The effects of PER on restarting a checkpointed program follow:

- **No PER monitoring before checkpoint, no PER monitoring after restart:** A program is running in an address space not monitored for PER interrupts; the program is checkpointed. If the program is restarted in an address space monitored for PER interrupts, SLIP does not monitor the restarted program.
- **PER monitoring before checkpoint, no PER monitoring after restart:** A program is running in an address space monitored for PER interrupts; the program is checkpointed. If it is restarted in an address space not monitored for PER interrupts, but other address spaces are being monitored, unwanted PER interrupts may occur, depending on the PER control register settings.

If unwanted PER interrupts occur in the restarted program, SLIP disables the PSW PER bit in the restarted program. This action can eventually remove all performance problems because of the unwanted PER interrupts from the restarted program.

- **PER monitoring before checkpoint, no PER monitoring after restart:** A program is running in an address space monitored for PER interrupts; the program is checkpointed. If the program is restarted and PER monitoring is not active in the system, the system performance might slow down because the PSW PER bit is enabled in the restarted program.

Dynamic PER Traps: TARGETID, along with indirect addressing on the RANGE parameter, allows you to activate a new PER trap dynamically once the previous trap has been deactivated as a result of MATCHLIM. In the next example, specifying TARGETID=TR2 on the first PER trap will cause the second PER trap, ID=TR2, to activate when the first trap deactivates. The address range of the second PER trap is determined by the contents of register 1 when the MATCHLIM occurs.

First PER trap:

```
SLIP SET,IF,RANGE=10000,TARGETID=TR2,ACTION=TARGETID,END
```

Second PER trap:

```
SLIP SET,SA,DISABLE,RANGE=1R?,ID=TR2,ACTION=SVCD,END
```

The second PER trap can specify a third trap and so on. There is no limit to the number of traps in a chain of dynamic traps, which is called a dynamic PER activation chain.

Each PER trap can be of any kind and have its own unique set of matching criteria or filters. However, PER traps will be active only for address spaces specified in the initial trap by the JOBNAME, ASID, and MODE=HOME parameters. Therefore IBM recommends that an ASID parameter specified in the initial trap must include all address spaces for the subsequent traps. For example, it makes sense to specify ASID=(1,2,3) on the initial trap and ASID=(1,2) on the second trap, but not the reverse, because no units of work would be monitored in ASID=3.

Rules for dynamic PER traps:

- TARGETID can be used with all PER traps except IGNORE.

- When TARGETID is one of the parameters, you must also specify it as one of the actions. Otherwise the TARGETID parameter will be ignored.
- Each of the PER traps can have its own independent action.
- When TARGETID is specified the default value for MATCHLIM is 1.
- The RANGE parameter on all PER traps support indirect addressing.
- A dynamic PER activation chain is defined when every TARGETID can be associated with some PER trap. SLIP does not allow the traps of the chain to be enabled unless the definition is complete.

A PER trap within a dynamic PER activation chain cannot target itself or a preceding trap in the chain. In other words, a dynamic PER activation chain cannot be a circular chain.

Members of the chain cannot be deleted. Any attempt to delete a member will result in message IEE408I being issued.

You can disable any trap in the chain in order to delete the chain. This can aid in tracking down the currently active trap.

- SLIP allows specification of NUCMOD, PVTMOD, and LPAMOD on the trap. However for PVTMOD, the cross memory lock (CML) of the primary address space that existed at the time of the PER interrupt must be immediately obtainable in order to search for the specified load module. If the CML is obtainable but the system does not find the specified load module, it puts the target trap into an enabled but inactive state.

Syntax for an Error Event SLIP SET Command

The following considerations apply:

- Do not enter a SLIP trap that uses all the defaults (SLIP SET,END). The defaults (ENABLE, RBLEVEL=ERROR, ACTION=SVCD, and MATCHLIM=1) cause the system to issue message IEA412I and request an SVC dump for a non-error condition.
- If you specify only one *nodump* value on the ACTION parameter, you do not need to enclose it in parentheses.

```
SLIP SET

[,ADDRESS=(start[,end])          ]
[,LPAEP=(name[,start[,end]])    ]
[,LPAMOD=(name[,start[,end]])   ]
[,NUCEP=(name[,start[,end]])    ]
[,NUCMOD=(name[,start[,end]])   ]
[,PVTEP=(name[,start[,end]])    ]
[,PVTMOD=(name[,start[,end]])   ]

[,ASID=(asid[,asid]...)         ]
[,COMP=code[,REASON=code]        ]
[,DATA=(comparison[,comparison]...) ]
[,ERRTYP=(type,[,type]...)      ]
[,JOBNAME={userid | jobname}    ]
[,JSPGM=name                     ]
[,MSGID=message-id              ]
[,MODE= (mode[,mode]...[,ANY | EVERY]) ]
[,PSWASC=(mode[,mode]...)        ]
```

SLIP Command

```
[,ACTION=[IGNORE[,option]]]
  [(nodump[,nodump]...)[,option]]
  [NOSUP[,option]]
  [RECORD]
  [SVCD[,options]]
  [TRACE[,options]]
  [TRDUMP[,options]]
  [STOPGTF[,options]]
  [WAIT[,options]]
]

[,ENABLE | ,DISABLE]

[,IDGROUP=idgroup]

[,MATCHLIM=m]
| [,MATCHLIM=1]   for ACTION=SVCD or ACTION=SYNCSVCD

[,DEBUG]

[,ID=trapid]

[,OK]

[,RBLEVEL={ERROR | NOTSVRB | PREVIOUS}]

,END
```

Syntax for an Instruction Fetch or Successful Branch SLIP SET PER Command

The following considerations apply:

- Only one PER trap with an action other than IGNORE can be eligible for checking at any one time.
- Except when ACTION=IGNORE is specified, one of the following parameters is required: LPAEP, LPAMOD, NUCEP, NUCMOD, PVTEP, PVTMOD, or RANGE. With ACTION=IGNORE, these parameters are optional.
- If you specify only one value in the ACTION parameter, you do not need to enclose it in parentheses.

```
SLIP SET,{IF|SBT}

[,LPAEP=(name[,start[,end]])]
| [,LPAMOD=(name[,start[,end]])]
| [,NUCEP=(name[,start[,end]])]
| [,NUCMOD=(name[,start[,end]])]
| [,PVTEP=(name[,start[,end]])]
| [,PVTMOD=(name[,start[,end]])]
| [,RANGE=(start[,end])]

[,ASID=(asid[,asid]...)]
[,DATA=(comparison[,comparison]...)]
[,JOBNAME={userid | jobname}]
[,JSPGM=name]
[,MODE= (mode[,mode]...[,ANY | EVERY])]
[,PSWASC=(mode[,mode]...)]
```

```
[,ACTION=
  [(IGNORE[,RECOVERY])] ]
  [(RECOVERY[,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID)][,options] ]
  [(STDUMP[,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID)][,options] ]
  [(STRACE[,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID)][,options] ]
  [(SVCD[,RECOVERY][,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID)][,options] ]
  [(SYNCSVCD[,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID)][,options] ]
  [(STOPGTF[,REFAFTER][,REFBEFOR][,TARGETID)][,options] ]
  [(TRACE[,RECOVERY][,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID)][,options] ]
  [(TRDUMP[,RECOVERY][,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID)][,options] ]
  [(WAIT[,RECOVERY][,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID)][,options] ]]

[,ENABLE | ,DISABLE]

[,IDGROUP=idgroup]

[,MATCHLIM=m ]
|,MATCHLIM=1      for ACTION=SVCD or ACTION=SYNCSVCD
|,MATCHLIM=50     for ACTION=STDUMP or ACTION=STRACE

[,PRCNTLIM=p | ,PRCNTLIM=10]

[,DEBUG]

[,ID=trapid]

[,OK]

,END
```

Syntax for a Storage Alteration SLIP SET PER Command

Use the SLIP SET,SA command to set a PER storage alteration trap.

Use the SLIP SET,SAS command to set a PER storage alteration trap pertaining to the STURA assembler instruction. The command traps a store into the virtual address range as well as a store done by STURA. The trap can use other parameters to check changes to the virtual range.

The following considerations apply:

- Only one PER trap with an action other than IGNORE can be eligible for checking at any one time.
- Except when ACTION=IGNORE is specified, the RANGE parameter is required. With ACTION=IGNORE, RANGE must not be specified.
- If you specify only one value in the ACTION parameter, you do not need to enclose it in parentheses.

SLIP Command

```
SLIP SET,{SA|SAS}

[,ADDRESS=(start[,end])      ]
[,LPAEP=(name[,start[,end]])]
[,LPAMOD=(name[,start[,end]])]
[,NUCEP=(name[,start[,end]])]
[,NUCMOD=(name[,start[,end]])]
[,PVTEP=(name[,start[,end]])]
[,PVTMOD=(name[,start[,end]])]

[,RANGE=(start[,end])]
[,ASID=(asid[,asid]...)]
[,ASIDSA=(asid | 'jobname'[,asid | ,jobname']...)]
[,DATA=(comparison[,comparison]...)]
[,DSSA=(asid.name | 'jobname'.name[,asid.name | ,jobname'.name]...)]
[,JOBNAME={userid | jobname}]
[,JSPGM=name]
[,MODE= (mode[,mode]...[,ANY | EVERY])]
[,PSWASC=(mode[,mode]...)]]

[,ACTION=
  [(IGNORE[,RECOVERY])]
  [(RECOVERY[,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID])[,options]]
  [(STDUMP[,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID])[,options]]
  [(STOPGTF[,REFAFTER][,REFBEFOR][,TARGETID)][,options]]
  [(STRACE[,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID])[,options]]
  [(SVCD[,RECOVERY][,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID])[,options]]
  [(SYNCSVCD[,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID])[,options]]
  [(TRACE[,RECOVERY][,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID)][,options]]
  [(TRDUMP[,RECOVERY][,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID)][,options]]
  [(WAIT[,RECOVERY][,REFAFTER][,REFBEFOR][,STOPGTF][,TARGETID)][,options]]]

  [,ENABLE | ,DISABLE]

  [,IDGROUP=idgroup]

  [,MATCHLIM=m ]
    [,MATCHLIM=1      for ACTION=SVCD or ACTION=SYNCSVCD]
    [,MATCHLIM=50     for ACTION=STDUMP or ACTION=STRACE]

  [,PRCNTLIM=p | ,PRCNTLIM=10]
  [,DEBUG]
  [,ID=trapid]
  [,OK]

,END
```

Syntax for the ACTION Parameters for the SLIP SET Command

```
ACTION=IGNORE[,option]
ACTION=(IGNORE[,RECOVERY])

Where option is:

[,RECORD]
```

ACTION=(nodump[,nodump]...)[,option]

Where nodump is:

NODUMP
NOSVCD
NOSYSA
NOSYSM
NOSYSU

Where option is:

[,RECORD]

ACTION=NOSUP[,option]

Where option is:

[,RECORD]

ACTION=RECORD

ACTION=RECOVERY

ACTION=(RECOVERY[,REFAFTER][,REFBEFOR][,TARGETID])[,options]

Where the options are:

[,REFAFTER=(triplet[,triplet]...)]

[,REFBEFOR=(triplet[,triplet]...)]

[,TARGETID=(trapid)]

ACTION=REFAFTER,REFAFTER=(triplet[,triplet]...)

ACTION=REFBEFOR,REFBEFOR=(triplet[,triplet]...)

ACTION=STDUMP[,options]

ACTION=(STDUMP[,REFAFTER][,REFBEFOR],[STOPGTF],[TARGETID])[,options]

Where the options are:

[,ASIDLST=(asid[,asid]...)]

[,DSPNAME=(asid.name | 'jobname'.name[,asid.name | , 'jobname'.name]...)]

[,LIST=(start,end[,start,end]...)]

[,REFAFTER=(triplet[,triplet]...)]

[,REFBEFOR=(triplet[,triplet]...)]

[,SDATA=(area[,area]...)]
|,SDATA=(NOALLPSA, NOALLSQA, NOSUM, TRT)

[,SUMLIST=(start,end[,start,end]...)]

[,TARGETID=(trapid)]

ACTION=STOPGTF

SLIP Command

```
ACTION=STRACE  
ACTION=(STRACE[,REFAFTER][,REFBEFOR],[STOPGTF],[TARGETID])[,options]
```

Where the options are:

```
[,REFAFTER=(triplet[,triplet]...)]
```

```
[,REFBEFOR=(triplet[,triplet]...)]
```

```
[,TARGETID=(trapid)]
```

```
ACTION=SVCD[,options]
ACTION=(SVCD[,RECOVERY][,REFAFTER][,REFBEFOR],[STOPGTF],[TARGETID])[,options]
```

Where the options are:

```
[,ASIDLST=(asid[,asid]...)]
[,DSPNAME=(asid.name | 'jobname'.name[,asid.name | ,jobname'.name]...)]
[,JOBLIST=(jobname[,jobname]...)]
[,LIST=(start,end[,start,end]...)]
[,RECORD]
[,REFAFTER=(triplet[,triplet]...)]
[,REFBEFOR=(triplet[,triplet]...)]
[,REMOTE= { (UNCOND | COND,remote) }
           { (UNCOND | COND,(remote)[,(remote)]...) }
           { (remote) }
           { ((remote)[,(remote)]...) }
[,SDATA=(area[,area]...)]
[,SDATA=(ALLPSA,CSA,LPA,NUC,RGN,SQA,SUM,TRT)]
[,STRLIST=(s-option[,s-option]...) ]
[,SUMLIST=(start,end[,start,end]...)]
[,TARGETID=(trapid)]
```

Where remote in the REMOTE parameter is:

```
[,SYSLIST=(sysname,group.member,group.*,...)]
[,ACTION=SVCD | WAIT]
[,ASIDLST=(asid[,asid]...)]
[,DSPNAME=(asid.name | 'jobname'.name[,asid.name | ,jobname'.name]...)]
[,JOBLIST=(jobname[,jobname]...)]
[,LIST=(start,end[,start,end]...)]
[,SDATA=(area[,area]...)]
[,STRLIST=(s-option[,s-option]...) ]
```

Where s-option in the STRLIST parameter is:

```
STRNAME=strname
[,CONNNAME=connname]
[,ACCESSTIME={ENFORCE | NOLIMIT}]
[,LOCKENTRIES]
[,USERCNTLS]
[,EVENTQS]
[,({EMCONTROLS={ALL | (list)})}]
[,{({COCLASS | STGCLASS | LISTNUM}={ALL | (list)})}
   {[ADJUNCT={CAPTURE|DIRECTIO}][,ENTRYDATA={UNSERIALIZE|SERIALIZE}]})
   {[SUMMARY]}]
```

SLIP Command

```
ACTION=SYNCSVCD[,options]
ACTION=(SYNCSVCD[,REFAFTER],[REFBEFOR],[STOPGTF],[TARGETID])[,options]
```

Where the options are:

```
[,ASIDLST=(asid[,asid]...)]
[,DSPNAME=(asid.name | 'jobname'.name[,asid.name | , 'jobname'.name]...)]
[,LIST=(start,end[,start,end]...)]
[,REFAFTER=(triplet[,triplet]...)]
[,REFBEFOR=(triplet[,triplet]...)]
[,REMOTE= { (UNCOND | COND,remote) }
           { (UNCOND | COND,(remote)[,(remote)]...) }
           { (remote) }
           { ((remote)[,(remote)]...) }
[,SDATA=(area[,area]...)]
           [,SDATA=(ALLPSA,CSA,LPA,NUC,RGN,
SQA,SUM,TRT)]
[,STRLIST=(s-option[,s-option]...)]
[,SUMLIST=(start,end[,start,end]...)]
[,TARGETID=(trapid)]
```

Where remote in the REMOTE parameter is:

```
[,SYSLIST=(sysname,group.member,group.*,...)]
[,ACTION=SVCD | WAIT]
[,ASIDLST=(asid[,asid]...)]
[,DSPNAME=(asid.name | 'jobname'.name[,asid.name | , 'jobname'.name]...)]
[,JOBLIST=(jobname[,jobname]...)]
[,LIST=(start,end[,start,end]...)]
[,SDATA=(area[,area]...)]
[,STRLIST=(s-option[,s-option]...)]
```

Where s-option in the STRLIST parameter is:

```
STRNAME=strname
[,CONNNAME=connname]
[,ACCESTIME={ENFORCE | NOLIMIT}]
[,LOCKENTRIES]
[,USERCNTLS]
[,EVENTQS]
[,EMCONTROLS={ALL | (list)}]
[,{(COCLASS | STGCLASS | LISTNUM)={ALL | (list)}}
   {[,ADJUNCT={CAPTURE|DIRECTIO}][,ENTRYDATA={UNSERIALIZE|SERIALIZE}]})
   {[,SUMMARY]})]
```

```
ACTION=TARGETID,TARGETID=(trapid)
```

```
ACTION=TRACE[,options]
ACTION=(TRACE[,RECOVERY][,REFAFTER][,REFBEFOR],[STOPGTF],[TARGETID])[,options]
```

Where the options are:

- [,RECORD]
- [,REFAFTER=(*triplet*[,*triplet*]...)]
- [,REFBEFOR=(*triplet*[,*triplet*]...)]
- [,TARGETID=(*trapid*)]
- [,TRDATA={({STD[,REGS][,list]}))
 {REGS[,list]}
 {list}}}

```
ACTION=TRDUMP[,options]
ACTION=(TRDUMP[RECOVERY][,REFAFTER][,REFBEFOR],[STOPGTF],[TARGETID])[,options]
```

Where the options are:

- [,ASIDLST=(*asid*[,*asid*]...)]
- [,DSPNAME=(*asid.name* | '*jobname*'.*name*[,*asid.name* | ,'*jobname*'.*name*]...)]
- [,LIST=(*start,end*[,*start,end*]...)]
- [,RECORD]
- [,REFAFTER=(*triplet*[,*triplet*]...)]
- [,REFBEFOR=(*triplet*[,*triplet*]...)]
- [,SDATA=(*area*[,*area*]...)]
 |,SDATA=(NOALLPSA,NOALLSQA,NOSUM,TRT)
- [,STRLIST=(*s-option*[,*s-option*]...)]
- [,SUMLIST=(*start,end*[,*start,end*]...)]
- [,TARGETID=(*trapid*)]
- [,TRDATA={({STD[,REGS][,list]}))
 {REGS[,list]}
 {list}}]

Where *s-option* in the STRLIST parameter is:

- STRNAME=*strname*
- [,CONNNAME=*connname*]
- [,ACCESSTIME={ENFORCE | NOLIMIT}]
- [,LOCKENTRIES]
- [,USERCNTLS]
- [,EVENTQS]
- [,(EMCONTROLS={ALL | (*list*)})]
- [,({COCLASS | STGCLASS | LISTNUM}={ALL | (*list*)})
 {[,ADJUNCT={CAPTURE|DIRECTIO}][,ENTRYDATA={UNSERIALIZE|SERIALIZE}]})
 {[,SUMMARY]})]

SLIP Command

```
ACTION=WAIT[,options]
ACTION=(WAIT[,RECOVERY][,REFAFTER][,REFBEFOR],[STOPGTF],[TARGETID])[,options]
```

Where the *options* are:

```
[,RECORD]
[,REFAFTER=(triplet[,triplet]...)]
[,REFBEFOR=(triplet[,triplet]...)]
[,REMOTE= { (UNCOND | COND,remote) }
          { (UNCOND | COND,(remote)[,(remote)]...) }
          { (remote) }
          { ((remote)[,(remote)]...) }
[,TARGETID=(trapid)]
```

Where *remote* in the REMOTE parameter is:

```
[,SYSLIST=(sysname,group.member,group.*,...)]
[,ACTION=SVCD | WAIT]
[,ASIDLST=(asid[,asid]...)]
[,DSPNAME=(asid.name | 'jobname'.name[,asid.name | , 'jobname'.name]...)]
[,JOBLIST=(jobname[,jobname]...)]
[,LIST=(start,end[,start,end]...)]
[,SDATA=(area[,area]...)]
[,STRLIST=(s-option[,s-option]...) ]
```

Where *s-option* in the STRLIST parameter is:

```
STRNAME=strname
[,CONNNAME=connname]
[,ACCESSTIME={ENFORCE | NOLIMIT}]
[,LOCKENTRIES]
[,USERCNTLS]
[,EVENTQ]
[,{EMCONTROLS={ALL | (list)}}]
[,{(COCLASS | STGCLASS | LISTNUM)={ALL | (list)}}
  {[ADJUNCT={CAPTURE|DIRECTIO}][,ENTRYDATA={UNSERIALIZE|SERIALIZE}]})
  {[SUMMARY]}])
```

SLIP SET Parameters

The parameters are presented alphabetically.

ACTION=value

ACTION=(value[,value]...)

Specifies what you want the system to do when the trap matches system conditions. The *value* is:

- IGNORE
- *nodump*, which is NODUMP, NOSVCD, NOSYSA, NOSYSM, and/or NOSYSU
- NOSUP
- RECORD
- RECOVERY
- REFAFTER
- REFBEFOR
- STOPGTF
- STDUMP
- STRACE
- SUBTRAP

|

- SVCD
- SYNCVCD
- TARGETID
- TRACE
- TRDUMP
- WAIT

If you omit the ACTION parameter, the default is ACTION=SVCD. If you specify more than one value, enclose the values in parentheses and separate them by commas.

Abbreviation: A

ACTION=IGNORE

When the trap matches for an error or PER event, requests that the system resume normal processing.

You can use IGNORE in a narrowly defined trap to exclude a subset of events from being trapped by a more general trap. For PER traps, the IGNORE trap must be the same type (IF, SA, SAS or SBT) as the more general trap or it will not be tested. For IF and SBT PER traps, use IGNORE traps to simulate multiple ranges for monitoring.

An IGNORE trap does not prevent PER interrupts from occurring in the range specified in the IGNORE trap; consider this fact when you set a percent time limit (PRCNTLIM) for a more general IF or SBT PER trap.

Use MATCHLIM on an IGNORE trap to ignore a specified number of events before SLIP takes the action on an associated non-IGNORE trap.

Note: If a recovery routine requests a dump, ACTION=IGNORE on a SLIP trap matching the error event will not suppress the dump. Specify NODUMP to suppress the dump.

Example:

ACTION=IGNORE

ACTION=nodump

ACTION=(nodump[,nodump]...)

When the trap matches for an error event, suppresses any dumps for the error requested by the system or a program.

This ACTION value is not valid for a PER trap.

The *nodump* is one of the following. If you specify only one value, omit the parentheses.

NODUMP

Suppresses SVC, SYSABEND, SYSUDUMP, or SYSMDUMP dumps requested while the system processes the error. NODUMP is useful for preventing dumps that may not be needed because accompanying messages provide all the needed problem data.

Note: NODUMP may not be effective for abend codes of 13E, 222, 33E, and 922.

NOSVCD

Suppresses all SVC dumps requested while the system processes the error.

SLIP Command

NOSYSA

Suppresses all SYSABEND dumps requested while the system processes the error.

NOSYSM

Suppresses all SYSDUMP dumps requested while the system processes the error.

NOSYSU

Suppresses all SYSUDUMP dumps requested while the system processes the error.

Example:

```
ACTION=(NOSYSA,NOSYSM)
```

When you specify ACTION=NODUMP, make sure the SLIP trap is specific. If your SLIP trap is too general, you might suppress dumps needed for other problems. For example, if you specify only a system completion code, all dumps for that code are suppressed. However, if you specify both a completion code and a job name, other jobs that abend with that completion code still produce dumps. The following example shows a specific SLIP definition with a completion code and job name:

```
SLIP SET,ACTION=NODUMP,COMP=806,JOBNAME=D10AXH1A,END
```

If a second error occurs during system processing for an event with ACTION=NODUMP specified, any dump requested for the second error is also suppressed. You can determine if a second error has occurred by checking both the job output messages and the logrec output. If either contains more than one abend, a second error occurred. If you need a dump for the second error, disable the SLIP trap that specifies ACTION=NODUMP and rerun the failing job.

ACTION=NOSUP

When the trap matches for an error event, prevents suppression by dump analysis and elimination (DAE) of any dumps requested for the error by the system or a program.

This ACTION value is not valid for a PER trap.

Example:

```
ACTION=NOSUP
```

ACTION=RECORD

When the trap matches for an error event, forces logrec data set recordings for every recovery routine, regardless of what the recovery routine specifies.

This ACTION value is not valid for a PER trap.

You can specify RECORD as the only ACTION value or as an option of the ACTION parameter in an error event trap. RECORD is not valid for a PER trap.

Example: To force a logrec recording for a X'0C6' abend by function recovery routines (FRRs) and ESTAE routines, enter:

```
SLIP SET,C=0C6,ACTION=RECORD,END
```

ACTION=RECOVERY

Initiates recovery processing for the interrupted process when the trap matches for a PER event. Any other action you specify is done before the recovery processing begins. Recovery processing initially causes the system to issue an X'06F' abend, but the recovery routines might change the abend code.

This ACTION value is not valid for an error event trap.

You can specify RECOVERY as the only ACTION value or with an ACTION value of IGNORE, SVCD, TRACE, TRDUMP, or WAIT in a PER trap.

Use the RECOVERY value carefully to avoid unexpected results. Before using RECOVERY, familiarize yourself thoroughly with the MVS recovery principles. In particular, ensure that recovery procedures have been established at the point where you are forcing recovery processing, and know what the recovery routines will do under the circumstances in which you are forcing recovery processing. See *z/OS MVS Programming: Authorized Assembler Services Guide* for more information about recovery processing.

ACTION=REFAFTER

ACTION=REFBEFOR

When the trap matches for a PER event, refreshes the contents of storage locations or registers.

This ACTION value is not valid for an error event trap or when the action is IGNORE.

When another action value is specified, REFAFTER or REFBEFOR specifies the time when SLIP performs the refresh. For example:

- ACTION=(SVCD,REFAFTER) causes the refresh to occur after the dump has been requested.
- ACTION=(SVCD,REFBEFOR) causes the refresh to occur before the dump is requested.

When you specify REFAFTER or REFBEFOR as an ACTION value, you must also specify the REFAFTER or REFBEFOR parameter as an option.

ACTION=STDUMP

Writes a SLIP system trace record when the trap matches for a PER event, and schedules an SVC dump when the trap is disabled or deleted. This SVC dump includes the registers and PSW for the current task.

This ACTION value is not valid for an error event trap. The ACTION=STDUMP parameter overrides DAE suppression.

Consider the following in selecting parameters to appear with ACTION=STDUMP:

- The following parameters require less SLIP processing to find a matching trap for an IF or SBT trap:

```
ASIDLST
ENABLE/DISABLE
END
ID
LIST
LPAEP/LPAMOD/NUCEP/NUCMOD/RANGE
MATCHLIM
SDATA
SUMLIST
```

- The following parameters require less SLIP processing to find a matching trap for an SA or SAS trap:

```
ASIDLST
ENABLE/DISABLE
END
ID
LIST
MATCHLIM
```

SLIP Command

RANGE
SDATA
SUMLIST

- If you use parameters other than these with ACTION=STDUMP, SLIP processing uses more system resources.
- The default match limit for STDUMP is 50 when the only parameters specified are those listed above. Otherwise there is no default.

ACTION=STRACE

When the trap matches for a PER event, writes at least one SLIP system trace record.

This ACTION value is not valid for an error event trap.

Considerations in selecting parameters to appear with ACTION=STRACE are:

- The following parameters require less SLIP processing to find a matching trap for an IF or SBT trap:

ENABLE/DISABLE
END
ID
LPAEP/LPAMOD/NUCEP/NUCMOD/RANGE
MATCHLIM

- The following parameters require less SLIP processing to find a matching trap for an SA or SAS trap:

ENABLE/DISABLE
END
ID
MATCHLIM
RANGE

- If you use parameters other than these with ACTION=STDUMP, SLIP processing uses more system resources.
- The default match limit for STRACE is 50 when the only parameters specified are those listed above. Otherwise there is no default.

ACTION=STOPGTF

Turns off GTF tracing when the SLIP trap becomes disabled (or deleted). This can happen either by operator command (for example the SLIP MOD or SLIP DEL command) or when the MATCHLIM or PRCNTLIM parameters on the SLIP command take effect. Using STOPGTF helps prevent the trace buffer from wrapping once the trap matches.

You can use STOPGTF with the GTFID, TRACE, and TRDUMP keywords. For example, if you specify ACTION=(STOPGTF,TRACE), SLIP produces a GTF trace records until the trap becomes disabled.

ACTION=SUBTRAP

When the trap matches for a PER event, allows the SLIP user to request different sets of actions depending upon system conditions at the time of the PER interrupt.

You can use the SUBTRAP action much the same way as the IGNORE action, except that most other actions are permissible on the trap. The SUBTRAP action imposes no default MATCHLIM value, but other specified ACTION keywords can impose one.

This ACTION value is not valid for an error event trap and when the action is TARGETID.

Example:

ACTION=SUBTRAP

ACTION=SVCD

When the trap matches for an error or PER event, schedules an SVC dump for the current or failing address space and issues an SPER system trace record to record exactly when the match occurred. For more information about SVC dumps, see the SVC dump chapter of *z/OS MVS Diagnosis: Tools and Service Aids*.

The SVCD value overrides DAE suppression but does not override suppression specified in a CHNGDUMP NODUMP operator command. If the dump cannot be written, perhaps because another SVC dump is in progress, SLIP issues message IEA412I, continues processing, and does not reschedule the dump.

The ASIDLST, DSPNAME, JOBLIST, LIST, SDATA, and SUMLIST parameters specify the data to be included in the dump. The SVC dump the system schedules includes the registers and PSW for the current or failing task.

If you omit an ASIDLST parameter, the dump includes the following address spaces. Note that the home address space is the space of the issuer of the CALLRTM TYPE=MEMTERM macro.

- RTM1: Failing address space, which is identified in SDWAFMID, or the home address space
- RTM2: Failing address space, which is identified in RTM2FMID, or the home address space
- MEMTERM: Master address space for the scheduled dump and the home address space for the summary dump.
- PER: Home address space
- REMOTE: For a dump on a remote system as requested by the REMOTE parameter, the XCF address space (XCFAS)

If dumping of a failing address space fails, that is, the SDUMP macro returns a nonzero return code, SLIP schedules a dump in the home address space but puts no problem data in the SDUMP 4K SQA buffer. If the second dump fails, SLIP issues message IEA412I.

Attention: Avoid using a general trap, such as, SLIP SET,COMP=0C4,ACTION=SVCD,END. The system normally has many expected program interrupts, each resulting in a 0C4 completion code.

SLIP has a default match limit of 1 on all traps that specify, or default to, ACTION=SVCD. The match limit can be changed by the MATCHLIM parameter when setting the SLIP trap. You can further qualify the SLIP trap by using other parameters, such as DATA and PVTMOD. These additional parameters prevent unwanted matches.

Example:

ACTION=SVCD,SDATA=(SQA,PSA,SUM)

ACTION=SYNCSVCD

When the trap matches for a PER event, schedules a synchronous SVC dump (SYNCSVCD) for the current or failing address space and issues an SPER system trace record to record exactly when the match occurred. The SYNCSVCD parameter overrides DAE suppression.

This ACTION value is not valid for an error event trap.

SLIP Command

The ASIDLST, DSPNAME, JOBLIST, LIST, SDATA, and SUMLIST parameters indicate the data to be included in the dump. The SVC dump the system schedules includes the registers and PSW for the current or failing task.

SLIP will stop the unit of work before starting the dump to ensure that the restart occurs after the dump has completed. SLIP stops the work only when all of the following conditions exist when the PER interrupt occurs:

- The system is enabled and unlocked
- The system is in task mode

Because the PER interrupt occurs on completion of the instruction, an instruction type such as SVC will cause the system not to meet the criteria listed above. If the system is disabled or locked when the PER interrupt occurs, a regular SVC dump will be taken instead.

SLIP has a default match limit of 1 on all traps that specify, or default to, ACTION=SYNCSVCD. You can change the match limit by the MATCHLIM parameter when setting the SLIP trap. You can further qualify the SLIP trap by using other parameters, such as DATA and PVTMOD. These additional parameters prevent unwanted matches.

Examples:

```
ACTION=SYNCSVCD  
ACTION=SYNCSVCD,SDATA=(SQA,PSA,SUM)
```

ACTION=TARGETID

When the trap matches for a PER event, specifies that another PER trap is to be activated once the trap specified in this SLIP command has been deactivated as a result of MATCHLIM.

This ACTION value is not valid for an error event trap or when the action is IGNORE.

When you specify TARGETID as an ACTION value, you must also specify the TARGETID parameter as an option.

The default match limit for ACTION=TARGETID is 1.

ACTION=TRACE

When the trap matches for an error or PER event, writes at least one generalized trace facility (GTF) SLIP trace record. Use a TRDATA parameter to specify the type and content of the GTF SLIP trace record.

For the record to be built and recorded, GTF with the SLIP option must be active. Be sure to stop GTF after the SLIP trap completes.

ACTION=TRACE can record small or large amounts of data. It is useful when you need a small amount of data each time a trap matches, such as when checking the path through a module.

Two problems can cause fields to be unavailable:

- The system uses the registers at the time of the event to resolve indirect addresses specified for trace record fields. If circumstances cause the registers to be unavailable, indirect addresses using a register value cannot be resolved, and related fields cannot be collected.
- The field is paged out or one of the pointers to the field is paged out.

If using indirect addresses, specify the REGS option of TRDATA to see the general purpose registers and the access registers used to resolve the addresses.

Example:

ACTION=TRACE

ACTION=TRDUMP

When the trap matches for an error or PER event, writes at least one generalized trace facility (GTF) SLIP trace record and, when the trap is disabled or deleted, schedules an SVC dump.

For the record to be built and recorded, GTF with the SLIP option must be active. Be sure to stop GTF after the SLIP trap completes.

The ASIDLST, DSPNAME, LIST, SDATA, and SUMLIST parameters are used to tailor the dump and TRDATA is used to tailor the trace records. The SVC dump the system schedules includes the registers and PSW for the current or failing task.

The ACTION=TRDUMP parameter overrides DAE dump suppression.

Example:

ACTION=TRDUMP, TRDATA=(STD,REGS), SDATA=(TRT,SQA)

ACTION=WAIT

When the trap matches for an error or PER event, issues an SPER system trace record to record exactly when the match occurred and displays the following information in message IEE844W and places the system in a wait:

- SLIP identifier
- Type of trap and related information:
 - RTM1: The address of the system diagnostic work area (SDWA)
 - RTM2: The address of the RTM2 work area (RTM2WA)
 - MEMTERM: The address of the address space control block (ASCB)
 - PER: The address and code of PER
- PSW at the time of error event or PER interruption
- Control registers 3 and 4 contents, which indicate the primary (PASID) and secondary (SASID) address spaces
- Contents of general registers 0 to 15

When SLIP processes an ACTION=WAIT trap, it also places information in an area pointed to by PSA location X'40C'. On a z/Architecture system, a 7th word at offset X'18' is added to that information. The 7th word contains the address of a 64-byte area that contains the high halves (bits 0–31) of the 64-bit GPRs in order GPR0 to GPR15. The second word, at offset X'04', contains the address of a 64-byte area that contains the low halves (bits 32–63) of the 64-bit GPRs in order GPR0 to GPR15.

The system instructs you to restart the system when you are finished looking at the message. The system restarts unless you did something during the wait to prevent a restart. Generating a stand-alone dump, for example, prevents a restart.

If the system cannot display the message, for example, because the console is unavailable, the system enters a restartable wait state (code 01B). You can then restart the system, if a restart is possible, depending on your actions during the wait.

Note: If the SLIP command is entered from a TSO terminal in OPERATOR mode, the system does not accept this parameter.

Example:

ACTION=WAIT

SLIP Command

ADDRESS=*start*

ADDRESS=(*start[,end]***)**

For a storage alteration PER trap, specifies the virtual address or range of addresses that must contain the instruction that causes the storage alteration.

For an error event trap, specifies the virtual address or range of addresses in which the error must occur. The RBLEVEL parameter controls the PSW address that is used to decide if ADDRESS matches, for an enabled, unlocked, task mode error that is matched against an error event trap.

Each address is 1 to 8 hexadecimal digits. The starting address must be less than or equal to the ending address.

ADDRESS is not valid for an instruction fetch (SLIP SET,IF) or successful branch (SLIP SET,SBT) PER trap.

Abbreviation: AD

Example:

ADDRESS=(CD300,CD400)

ASID=*asid*

ASID=(*asid[,asid]...***)**

For an error event or PER trap, specifies the address space identifier (ASID) for the address space that must be in control when the error event or PER interruption occurs.

Each *asid* is 1 to 4 hexadecimal digits. You can specify one to 16 ASIDs. If you specify one ASID, you can omit the parentheses.

For PER traps with an action of IGNORE, you can limit the address spaces being monitored by the following:

- ASID parameter
- JOBNAME parameter
- MODE=HOME parameter

If you omit all of these parameters, PER monitoring is active in all address spaces and may cause performance problems.

If you specify both ASID and JOBNAME, one of the specified address spaces must be the one in which the job is running or the trap will not match.

If you omit MODE=HOME, PER monitoring will be active only for the address space(s) specified by the ASID parameter for the job specified by the JOBNAME parameter. If you omit MODE=HOME and ASID but specify JOBNAME, PER monitoring will be active for any address space(s) in which the job runs.

The ASID parameter is useful when monitoring storage alteration of an address range that falls in the private area.

Note: For PER traps that do not specify ACTION=IGNORE and for which the ASIDs to be monitored are known, but for which no extra PER interrupts will occur if all ASIDs are monitored, it is best not to specify the ASID parameter or MODE=HOME.

Abbreviation: AS

Example:

ASID=(1,7,1A)

ASIDLST=*asid*

ASIDLST=(*asid*[,*asid*]...)

As an option of an ACTION or REMOTE parameter, specifies the address space or spaces to dump.

The *asid* is 1 to 4 hexadecimal digits or a symbolic ASID. You can specify one to 15 ASIDs. If you specify only one ASID, you can omit the parentheses. The symbolic values are:

CURRENT or CU	Current address space.
HASID or H	Home address space.
I	Address space where the instruction ran.
LLOC	Address space that is locked.
PASID or P	Primary address space.
SA	Current alteration space used by an SA trap. If the alteration space is a data space, the data space will be dumped.
SASID or S	Secondary address space.

Note: Zero indicates the current address space.

Abbreviation: AL

Example:

ASIDLST=(0,C)

ASIDSA=*asid***ASIDSA='*jobname*'****ASIDSA=(*asid* I '*jobname*', [*asid* I ,'*jobname*']...)**

For a storage alteration (SLIP SET,SA or SAS) trap, specifies up to 16 address spaces to be monitored for storage alterations.

If neither ASIDSA nor DSSA is specified, the trap applies to all storage alterations in all address spaces, data spaces, and hiperspaces in the address range in the RANGE parameter.

ASIDSA eliminates data spaces and hiperspaces from being monitored and restricts the trap to one or more address spaces or jobs. You can specify an address space name in two forms:

asid
'*jobname*'

asid

Specifies the address space identifier (ASID) of an address space to be monitored. The *asid* can be an explicit hexadecimal ASID or a symbolic ASID. The symbolic values are:

CURRENT or CU	Current address space.
HASID or H	Home address space.
I	Address space where the instruction executed.
PASID or P	Primary address space.
SA	Current alteration space used by an SA or SAS trap. It must be an address space. If the storage alteration occurs in a data space or a hiperspace, the trap will not match.
SASID or S	Secondary address space.

jobname

Specifies the job name associated with an address space that is being altered. The *jobname* is 1 to 8 alphanumeric and national characters (\$, #, @) and is enclosed in single quotes. You can specify wildcards in the

SLIP Command

jobname with the following exception: an * must be a suffix and cannot appear alone. See “Using Wildcards in Commands” on page 1-17.

Abbreviation: ASA

Examples:

Example 1: When the PER2 hardware feature is present and the address space is known explicitly (not symbolically), it is particularly beneficial to specify ASIDSA.

ASIDSA=(5,3A,17B,24E,'J0B1','J0B*')

Example 2: A storage alteration of the word at common location 600 in the prefix storage area (PSA) results in a trap for which SLIP performs match processing. Because ASIDSA=SA is specified, SLIP will not check for a match for a storage alteration event that occurs in either a data space or a hyperspace.

RANGE=(600,603),ASIDSA=SA

COMP=code

For an error event trap, specifies a system or user completion code that is associated with the error.

For a system completion code, the form is *hhh*, three hexadecimal digits. You can indicate a set of codes by substituting x's for one or more of the digits. For example, x23 means 123, 223, 323, 423, and so forth. You can use an x in any position.

For a user completion code, the form is U₁ddd, where U indicates a user code followed by four decimal numbers. Use an x for any of the numbers to specify a set of codes. For example, U102x means U1021, U1022, U1023, and so forth.

Notes:

1. The SLIP action is not taken when the abend completion code is originally a program check (code 0C4) that the system converts to a new value. The following abend completion codes may be originally a program check and converted ones: 11A, 12E, 15D, 15F, 200, 212, 25F, 279, 282, 42A, 430, 57D, 700, 72A, A00, B00, and E00. Note that the SLIP action is taken when the abend is not converted, even for these completion codes. For example, the action is taken for SLIP C=15F,RE=190,A=.
2. For abend completion codes 201, 202, 402, 6FC and 702, the SLIP action might not be taken. In certain paths, each of these codes is originally a program check, as described in Note 1. In other paths, the abend is issued directly. To cover fully such a case, you should set, for example, one SLIP trap specifying COMP=201 and another specifying COMP=0C4 or ERRTYP=PROG.
3. SLIP cannot trap the 922 and 13E abend codes used to purge subtasks.
4. If any completion code is changed by a recovery routine with the SETRP macro, specify the original completion code in the SLIP command. For example, if a code of 800 was originally a code of 171, specify a code of 171 on the COMP parameter.
5. Avoid setting a general trap, such as:
SLIP SET,COMP=0C4,ACTION=SVCD,END
The system normally has many expected program interrupts, each resulting in a 0C4 completion code.

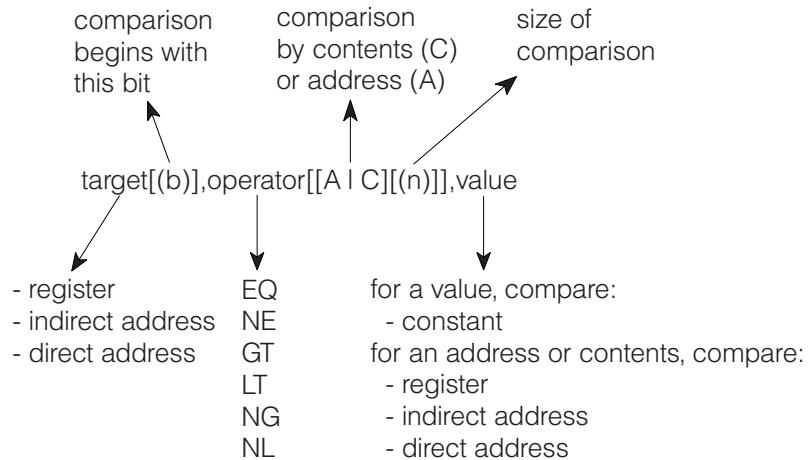
Abbreviation: C

Example:

COMP=U123x, REASON=8

DATA=(comparison)**DATA=(comparison[,comparison]...)**

For an error event or PER trap, specifies logical comparison of a target location to a value. The *comparisons* represents a triplet or group of triplets, where each **triplet** consists of a target, operator, and value, as follows:



You can specify any number of DATA triplets. You can use AND (&) and OR (|) to logically combine the DATA triplets. Use parentheses to group and order your comparisons. You can specify any number of parentheses, up to a nested depth of 16 parentheses. Some rules for coding the DATA parameter are:

- Either a comma or a parenthesis must precede and follow each AND and OR.
- The symbols & and | do not need to be preceded and followed by a comma or a parenthesis. The commas and parentheses are optional.
- If two DATA triplets are separated by only a comma (without a parenthesis on either side of the comma), AND is the default for the logical comparison. This is the only default on the DATA parameter.

The parts of a triplet are:

target

Specifies the address of a storage location or a general purpose register (GPR) whose contents SLIP compares against the value supplied. The target can be:

- A direct address of 1 to 16 hexadecimal digits using the G suffix, or 1 to 8 hexadecimal digits using the R suffix. (See “Qualifying Direct or Indirect Addresses to Address Spaces” on page 4-511.)
- A GPR in the form xG or xR where x is 0 – 15.
- An indirect address (see “Indirect Addresses” on page 4-508)

If SLIP cannot establish addressability to the target location, SLIP issues message IEA413I and increments the counter for the trap.

SLIP Command

b — binary compare

If specified, *b* modifies the target address by indicating the bit position where a binary comparison is to start. For 64-bit registers, *b* can be 0 – 63. The starting bit position plus the bit size for 64-bit register comparison must not exceed 63. For 31-bit registers, *b* can be 0 – 31. The starting bit position plus the bit size for 31-bit register comparison must not exceed 31.

operator

Specifies the relationship that must exist between the contents of the target location and the value for the comparison to be successful.

EQ – equal

NE – not equal

GT – target greater than value

LT – target less than value

NG – target not greater than value

NL – target not less than value

If you do not include either *C* or *A* with the operator, SLIP does a *value compare*. In a value compare, SLIP compares the contents of the target address to the specified value.

C — contents compare

SLIP processing is to compare the contents of the address specified as the target with the contents of the address specified as the value.

A — address compare

SLIP processing is to compare the contents of the address specified as the target with the address specified as the value. Do not request a binary compare with an address compare. For example, (1R(0),EQA,2R%) is not a valid combination of binary compare (1R(0)) and address compare (EQA). In a 4-byte compare, bit 0 is ignored; bits 1 to 31 are compared.

- n* The number of bytes or bits that SLIP processing is to compare for a contents or address compare. When you request a binary compare, the range is 1 to 8 bits, with the default being one bit. Otherwise, the range is 1 to 4 bytes with the default being 4 bytes.

Note: When SLIP does either a contents compare or an address compare, it looks at the first *n* bytes of storage, but the last *n* bytes of a register.

value

Specifies the value to which the contents of the target are to be compared.

- If you omit *A* or *C* as part of the operator, the value can be a constant, with the length determining the number of bytes or bits SLIP processing compares with the target.

If *b* is specified, the value is binary digits. If *b* is not specified, the value is hexadecimal digits. For example, 5R,EQ,01 is hexadecimal; 5R(0),EQ,01 is binary.

For binary comparisons, the length of the value establishes the length of the comparison. The maximum length for a binary comparison is 8 bits. The binary comparison can cross a byte boundary but not a register boundary.

For hexadecimal comparisons when the target is not a register, the length of the value establishes the length of the comparison. The maximum length of the comparison, however, is 4 bytes. The value can be 1 to 8 hexadecimal digits (for example, A24,EQ,3CA79 compares two and a half bytes of data starting at location A24).

For hexadecimal comparisons when the target is a register, the length of the comparison is 4 bytes, and the value is right-justified (for example, 2R,EQ,4 and 2R,EQ,00000004 are equivalent).

Underscores ('_') may be used at any point within the hexadecimal specification to make entering 64-bit data easier. Underscores are ignored during processing and do not count towards the limit on the number of hexadecimal digits.

If you specify A or C as part of the operator, the value can be:

- A direct address of 1 to 16 hexadecimal digits
- A GPR in the form xG or xR where x is 0 - 15
- An indirect address

Abbreviation: DA

Examples: Examples of DATA parameters follow.

Example 1

The following expression is true if general purpose register 2 contains either 0 or 4:

```
DATA=(2R,EQ,0,OR,2R,EQ,4)
where 2R is the target (general purpose register 2)
      EQ is the operator (equal)
      0 and 4 are the values in hexadecimal
```

Example 2

The following expression groups triplets for SLIP to evaluate.

```
DATA=(3R,EQ,8,AND(4R,EQ,0,OR(4R,EQ,4,AND,5R,NE,0)OR,4R,EQ,8))
```

Example 3

The following expression is true if the contents of register 1 are the same as the address that is 6 bytes beyond the location that register 3 is pointing to:

```
DATA=(1R,EQA(4),3R?+6)
```

Note that SLIP compares bits 1 — 31 of register 1 to the address that is 6 bytes beyond the location to which register 3 is pointing.

Example 4

The following expression is true if the contents of register 1 is the same as the contents of register 2:

```
DATA=(1R,EQC,2R)
```

Example 5

The following expression is true if the contents of register 1 is the same as the contents at location 224:

```
DATA=(1R,EQC,224)
```

Example 6

The following expression is true if the contents of register 1 is 224.

```
DATA=(1R,EQ,224)
```

SLIP Command

Example 7

The following expression is true if the first two bytes at the location pointed to by register 1 are equal to the first two bytes at the location determined by adding 150 to the contents of the word at location X'10'.

DATA=(1R?,EQC(2),10?+150)

Example 8

The following expression is true if the first two bits at the location pointed to by register 1 are equal to the first two bits at the location determined by adding X'150' to the contents of the word at location 10.

DATA=(1R?(0),EQC(2),10?+150)

Example 9

The following expression is true if the last 3 bytes (bits 8 — 31) of the address in register 1 are the same as the address that is 4 bytes beyond the location that register 3 is pointing to:

DATA=(1R,EQA(3),3R?+4)

Example 10

The following expression is true if the last 3 bytes of register 1 are the same as the first 3 bytes at the location pointed to by register 2:

DATA=(1R,EQC(3),2R?)

Example 11

The following expression is true if the 64-bit contents of GPR 11 equal the hexadecimal value X'0123456799999999':

DATA=(11G,EQ,01234567_99999999)

DEBUG

For a SLIP SET trap, allows you to determine why a trap that you set is not working as you expected by indicating which of the conditions you established is not being met. DEBUG provides trap information each time the trap is tested rather than just when it matches.

The generalized trace facility (GTF) and its trace option for SLIP records must be active. Each DEBUG trace record contains SLIP information plus two bytes: the first byte contains a value indicating the failing parameter and the second byte contains zero.

For a description of the DEBUG values, see the SLIP debug trace record for GTF in *z/OS MVS Diagnosis: Tools and Service Aids*.

Example:

DEBUG

DISABLE

For a SLIP SET trap, indicates that the trap set is to be initially inactive, that is, ineligible for checking. If DISABLE is omitted, ENABLE is the default.

Abbreviation: D

Example:

DISABLE

DSPNAME=*asid.name*
DSPNAME='*jobname'.name*
DSPNAME=(*asid.name* | '*jobname'.name*[, *asid.name* | '*jobname'.name']...**)***

As an option of an ACTION or REMOTE parameter, specifies the data space or spaces to be included in an SVC dump.

Specify from 1 to 15 data space names in the parameter. When you specify more than one name, enclose the data space names in parentheses and separate them by commas. When you specify only one name, you can omit the parentheses. You can specify a data space name in two forms:

asid.name
'i*jobname'.name*

asid

Specifies the address space identifier (ASID) of the address space related to a data space to be monitored. The *asid* can be an explicit hexadecimal ASID or a symbolic ASID. The symbolic values are:

CURRENT or CU	Current address space.
HASID or H	Home address space.
HOME	Home address space.
I	Address space where the instruction executed.
LLOC	Address space that is locked.
PASID or P	Primary address space.
SA	Current alteration space used by an SA trap
SASID or S	Secondary address space.

jobname

The name of the job associated with the data space. The *jobname* is 1 to 8 alphanumeric and national (\$, #, @) characters and must be enclosed in single quotes. You can specify *jobname* in a DSPNAME parameter:

- With ACTION=SVCD
- On the REMOTE parameter for an ACTION=SVCD trap

You can specify wildcards in the *jobname*. See “Using Wildcards in Commands” on page 1-17.

name

Specifies the 1 to 8 character *name* associated with the data space at its creation. You can specify wildcards in the *name* on the DSPNAME option. See “Using Wildcards in Commands” on page 1-17.

The *name* must be specified, unless the trap event is SA. For an SA trap event, the data space of the storage being altered is dumped.

When the interrupted unit of work holds a lock higher than the RSM lock, the system cannot determine the specific data spaces. In this case, no data spaces are included in the dump.

Abbreviation: DN

Examples:

```
DSPNAME=(0006.SDUMPCSA)
DSPNAME=( '*' .SPD*, '*ABC*'.S?P?,0012.SPD20)
```

DSSA=*asid.name*
DSSA='*jobname'.name*

SLIP Command

DSSA=(*asid.name* | '*jobname'.name*[, *asid.name* | '*jobname'.name']...)*

For a storage alteration (SLIP SET,SA or SAS) trap, specifies one or more data spaces to be monitored.

You can specify 1 to 16 data space names. You can specify a data space name in two forms:

asid.name
'jobname'.name

asid

Specifies the address space identifier (ASID) of the address space related to a data space to be monitored. The *asid* can be an explicit hexadecimal ASID or a symbolic ASID. The symbolic values are:

CURRENT or CU	Current address space.
HASID or H	Home address space.
I	Address space where the instruction executed.
LLOC	Address space that is locked.
PASID or P	Primary address space.
SA	Current alteration space used by an SA trap
SASID or S	Secondary address space.

jobname

The name of the job associated with the data space. The *jobname* is 1 to 8 alphanumeric and national (\$, #, @) characters and must be enclosed in single quotes. You can specify wildcards in the *jobname* with the following exception: an * must be a suffix and cannot appear alone. See “Using Wildcards in Commands” on page 1-17.

name

Specifies the 1 to 8 character *name* associated with the data space at its creation.

Notes:

1. When SA is specified with *name*, the storage alternation must occur in the named data space for the trap to match.
2. When *name* is not specified, the trap will match on a storage alternation into any data space owned by the specified address space.
3. When SA is specified without *name* and the storage alternation is in an address space, no match will result for that address space.
4. When the PER2 hardware feature is present and you can specify the data space explicitly (not symbolically), it is particularly beneficial to specify DSSA.

Example:

DSSA=(ASID.DSPACE1,I.DSPACE2)

ENABLE

For a SLIP SET trap, indicates that the trap defined is to be initially active, that is, eligible for checking. If DISABLE is omitted, ENABLE is the default.

Abbreviation: EN

Example:

ENABLE

END

For a SLIP SET trap, marks the end of the SLIP SET command. If you omit this parameter, the system prompts you for additional parameters.

Abbreviation: E**ERRTYP=***type***ERRTYP=(***type*[,*type*]...**)**

For an error event trap, specifies one or more error events, which satisfy the match test. If you specify ALL or more than one error type, the occurrence of any one of them satisfies the match test. If you omit ERRTYP, the omission has the same effect as specifying ERRTYP=ALL.

The *type* is one of the following:

ABEND	An ABEND macro issued by a task
ALL	All of the error conditions
DAT	Dynamic address translation error
MACH	Software error caused by a machine check
MEMTERM	Abnormal end of an address space
PGIO	Paging I/O error
PROG	Program check interruption
REST	Restart interruption

Note: The system intercepts restart interruptions only when both of the following conditions are true:

- The operator initiated the restart.
- The operator requested that the system abnormally end a program running on the restarted processor.

SVCERR SVC error caused by issuing an SVC instruction while locked, disabled, or in SRB mode

Abbreviation: ER**Example:**

ERRTYP=(MACH,DAT)

GTFID=*id***GTFID=(***id*[,*id*]...**)**

For a SLIP SET trap, specifies a GTF identifier (GTFID) to selectively stop a subset of GTF tracing. The *id* is 1 to 8 characters. You can specify one to a maximum of 15 IDs.

You must use the GTFID parameter in conjunction with ACTION=STOPGTF. When the SLIP trap matches, the GTF instances associated with the specified IDs are stopped. If you do not specify GTFID, the default is that all GTF instances are stopped.

Example:

GTFID=(HM1)

ID=*trapid*

For a SLIP SET trap, specifies a trap identifier. The *trapid* is 1 to 4 alphanumeric or national (\$, #, @) characters. If ID is not indicated in a SLIP SET command, the system assigns a unique id.

Example:

ID=PER1

SLIP Command

IDGROUP=idgroup

For a SLIP SET trap, specifies the name of a group of related traps. The *idgroup* name is 1 to 16 alphanumeric or national (\$, #, @) characters. An *idgroup* is self-defining, that is, it consists of all SLIP traps that specify the same *idgroup* name.

In a sysplex, the traps in the group can be on the same or different systems. More than one *idgroup* can be defined at the same time on a system or sysplex.

When a system disables one trap on reaching the MATCHLIM or PRCNTLIM for the trap, the system or systems disable all the traps in the group. Note that, if the operator deletes or disables one trap in a group, only that trap is disabled; the other traps in the group continue.

Set up a group when you are trying to diagnose a problem that can occur on several or all of the systems in a sysplex. After the trap occurs and is disabled on one system, the IDGROUP parameter disables the traps on the other systems, so that manual operator intervention is not needed.

Abbreviation: IG

Example:

```
IDGROUP=TRAPGROUP3
```

IGNORE

See the ACTION=IGNORE parameter.

IF

Specifies the event as an instruction fetch. This parameter is positional; it must appear following SET and a comma.

JOBLIST=jobname

JOBLIST=(jobname[,jobname]...)

As an option of an ACTION or REMOTE parameter, identifies the names of jobs whose address spaces are to be dumped when the action is SVCD or SYNCSCD in:

- The system in a sysplex that consists of one system
- The local system in a sysplex
- Another system in a sysplex, if REMOTE is specified

Specify from 1 to 15 job names. When you specify more than one name, enclose the names in parentheses and separate them by commas. When you specify only one name, you can omit the parentheses.

A *jobname* is 1 to 8 alphanumeric and national (\$, #, and @) characters. You can specify wildcards in a *jobname*. See “Using Wildcards in Commands” on page 1-17.

Abbreviation: JL

Examples:

```
JOBLIST=(job1)
JOBLIST=(job?,TRAN*,A??XYZ)
JOBLIST=(job1,job2,job3)
```

JOBNAME=userid

JOBNAME=jobname

For an error event or PER trap, specifies the user ID of a TSO/E user or the job name of the job or started task to be monitored.

The *userid* is 1 to 7 characters and the *jobname* is 1 to 8 characters. You can specify wildcards in the *userid* or *jobname* with the following exception: an *

must be a suffix and cannot appear alone. For example, JOBNAME=MASTER* is NOT valid, while JOBNAME=?MASTER? is valid. See “Using Wildcards in Commands” on page 1-17.

For error event traps, the specified job name must be for the home (dispatched) address space.

For non-IGNORE PER traps, JOBNAME limits PER monitoring to the address spaces in which the specified job runs. If the unit of work runs in an address space other than the one in which it was dispatched, PER monitoring will also be active in that address space for that particular job.

For PER traps that do not specify ACTION=IGNORE, you can limit the address spaces being monitored by the following:

- ASID parameter
- JOBNAME parameter
- MODE=HOME parameter

If you omit all of these parameters, PER monitoring is active in all address spaces and may cause performance problems.

If you specify both ASID and JOBNAME, one of the specified address spaces must be the one in which the job is running or the trap will not match.

If you omit MODE=HOME, PER monitoring will be active only for the address space(s) specified by the ASID parameter for the job specified by the JOBNAME parameter. If you omit MODE=HOME and ASID but specify JOBNAME, PER monitoring will be active for any address space(s) in which the job runs.

Abbreviation: J

Example:

JOBNAME=D10AXX1

JSPGM=name

For an error event or PER trap, specifies the 1 to 8 character name of the job step program that is to be monitored for a match. On error event traps, JSPGM causes a no-match if ERRTYP specifies MEMTERM.

Abbreviation: JS

Example:

JSPGM=IFOX00

LIST=(start,end)

LIST=(start,end[,start,end]...)

As an option of an ACTION or REMOTE parameter, specifies one or more storage areas to be included in an SVC dump. Each area is defined by a set of starting and ending addresses, which can be either direct or indirect. (See “Indirect Addresses” on page 4-508.) A starting address must be less than or equal to the ending address.

LIST is similar to the SUMLIST parameter; the area specified by LIST is in a scheduled SVC dump while the area specified by SUMLIST is in a disabled summary dump.

Abbreviation: LS

Example:

ACTION=SVCD,LIST=(152,155,10%+20%,+3F)

SLIP Command

LPAEP=name
LPAEP=(name[,start[, end]])

For an error event or PER trap, monitors modules in the link pack area (LPA).

The values are:

name

The entry point name or alias. The *name* is 1 to 8 characters. If the last character is an asterisk (*), SLIP interprets the asterisk as X'C0'. (Certain module names end with the character X'C0'.) If only *name* is specified, the range of monitoring is from the entry point or alias to the end of the module.

start

end

Offsets from the entry point or alias; they indicate the start and end of the range to be monitored. The *start* must be less than or equal to the *end*. For more information, see the notes for the RANGE parameter. If you specify only *start*, the range consists of that single address.

For an error event trap and an IF or SBT PER trap, LPAEP establishes the range of addresses to be monitored.

For an SA or SAS PER trap, LPAEP indicates that the storage alteration must be caused by an instruction within the specified range.

For an enabled, unlocked, task mode error that is matched against an error event trap, the RBLEVEL parameter controls the PSW address that is used to decide if LPAEP matches.

Example:

LPAEP=IEECB907, ...

If the system does not find the module in the LPA when processing the SLIP command, it assumes that you may later add the module dynamically to the LPA, and issues message IEE101I.

LPAMOD=name
LPAMOD=(name[,start[, end]])

For an error event or PER trap, monitors modules in the link pack area (LPA).

The values are:

name

The module name or alias. The *name* is 1 to 8 characters. If the last character is an asterisk (*), SLIP interprets the asterisk as X'C0'. (Certain module names end with the character X'C0'.) If *name* is the alias, SLIP processing uses the load module name. If only *name* is specified, the range of monitoring is the entire module.

start

end

Offsets from the start of the module; they indicate the start and end of the range to be monitored. The *start* must be less than or equal to the *end*. For more information, see the notes for the RANGE parameter. If you specify only *start*, the range consists of that single address.

For an error event trap and an IF or SBT PER trap, LPAMOD establishes the range of addresses to be monitored.

For an SA or SAS PER trap, LPAMOD indicates that the storage alteration must be caused by an instruction within the specified range.

For an enabled, unlocked, task mode error that is matched against an error event trap, the RBLEVEL parameter controls the PSW address that is used to decide if LPAMOD matches.

If the system does not find the module in the LPA when processing the SLIP command, it assumes that you may later add the module dynamically to the LPA, and issues message IEE101I.

Abbreviation: L

Examples:

Example 1:

```
LPAMOD=(IEAVXYZ,2C)
```

Example 2: To monitor events in a load module for occurrences of SVC60, enter:

```
LPAMOD=IGC0006*
```

MATCHLIM=m

For an error event or PER trap, specifies that the SLIP trap is to be disabled after *m* matches, where *m* is an integer from 1 to 65535. The default values for MATCHLIM are:

- If you specify TARGETID the default for MATCHLIM is 1.
- If you omit MATCHLIM but specify ACTION=SVCD or ACTION=SYNCSVCD, the trap is disabled after one match.
- If you omit MATCHLIM for a PER trap with ACTION=STRACE or ACTION=STDUMP, see the documentation of those keywords earlier in this chapter regarding MATCHLIM processing.
- If you omit MATCHLIM for any other type of trap, the trap can match any number of times. No other parameters are considered in determining the default for MATCHLIM.
- If you specify multiple action parameters that each have a default match limit, the system selects the lowest default.

Use a DISPLAY operator command to display the number of times that the conditions for a SLIP trap are met since the last time the trap was enabled.

Note: Between the instant matchlim is reached and when the trap is actually disabled, a small amount of time elapses. It is possible for the trap to match on another CPU during this small time interval. If this occurs, matchlim will actually be exceeded, with unexpected results. Therefore, use caution in setting a trap in a heavily used module as, for example, the dispatcher.

Abbreviation: ML

Example:

```
MATCHLIM=50
```

MODE=mode

MODE= (mode[,mode]...[, ANY | EVERY])

For an error event or PER trap, specifies the mode the system must be in for

SLIP Command

the trap to match. You can specify more than one mode. You can indicate how many modes are needed to cause a match by one of the following.

ANY	Any one of the listed modes
EVERY	Every one of the listed modes

ANY and EVERY cannot appear alone without one or more modes and cannot appear together.

The modes are:

ALL	All of the above except HOME
DIS	Physically disabled for I/O and external interruptions
GLOC	Holding any global lock
GLOCSD	Holding a global suspend lock
GLOCSP	Holding a global spin lock
HOME	Executing in the home (dispatched) address space
LLOC	Holding a local lock
LOCK	Holding any lock
PKEY	Problem program key (8 or more)
PP	Problem program state
RECV	Recovery routine in control
SKEY	System key (7 or less)
SRB	SRB mode
SUPER	Supervisor state
SUPR	Supervisor control mode (any bit set in PSASUPER)
TCB	TCB mode
TYP1	Type 1 SVC in control

Notes:

1. Specifying the LLOC, LOCK, or ALL option of the MODE parameter automatically includes the cross memory local lock (CML).
2. Like ASID and JOBNAME, MODE=HOME limits PER monitoring. For non-IGNORE PER traps, specifying MODE=HOME indicates that PER monitoring is to be active only when the unit of work executes in the address space in which it was dispatched.
3. For an enabled, unlocked, task mode error that is matched against an error event trap, the RBLEVEL parameter controls the PSW address that is used to decide if MODE matches.
4. Whenever you specify HOME, regardless of the ANY/EVERY option you specify or default to, the unit of work must be executing in the home (dispatched) address space. If you specify or default to the ANY option, at least one of the other modes you specify must be the same as the system mode for a match to occur.
5. The RECV mode cannot be specified on a PER trap.

Abbreviation: M

Example:

MODE=(LLOC,SRB,EVERY)

MSGID=*message-id*

Causes control to be passed to the SLIP action processor under the unit of work issuing the WTO when the MSGID of the WTO matches the message ID specified on the MSGID parameter. The slip action processor gets control after SSI and MPF processing.

The input MSGID can be a maximum of 10 characters. When the MSGID is not contained within single quotes, only alphanumeric characters are accepted. The

character immediately following the MSGID in the WTO must be blank for the trap to match. When a quoted MSGID is specified, the MSGID might contain any characters. The character immediately following the MSGID in the WTO need not be blank. For example, MSGID='AMSGID' will match a WTO whose first characters are AMSGIDX, but MSGID=AMSGID will not.

Except in the case of connected branch entry WTOs, SLIP will perform match processing on each line of the multiline messages. For example, if a SLIP is set on the IEE852I message with a MATCHLIM=5 and ACTION=SVCD, five dumps will be produced for the same issuance of the message since it has multiple lines of output.

SLIP does not get control:

- For messages that are reissued, for example, messages that are issued on one system and appear on another, or branch entry WTOs that are reissued.
- For branch entry WTO if the NLCKS, LOADWAIT, or SYNCH=YES parameter is specified on the WTO invocation.
- For branch entry WTO in certain situations where abending the unit of work may cause system problems.
- For minor lines associated with a branch entry WTO.

When the SLIP action processor gets control these registers are set:

- Register 2 — contains the address of the SLIP message data area, found in mapping macro IHASLMSG.
- Register 3 — contains the address of the text (or the major line if there are minors associated with the major).
- Register 4 — contains the address of the SLIP message data area, associated with the minor line, or 0 if there is no minor line (applicable for SVC WTO only).
- Register 5 — contains the address of the text for the minor line or 0 if there is no minor line (applicable for SVC WTO only).

Once in the SLIP action processor, the MSGID filter along with all other SLIP keywords that may have been specified (for example, DATA or JOBNAME) apply in determining whether the trap will match.

NODUMP

NOSVCD

NOSYSA

NOSYSM

NOSYSU

See the ACTION=*nodump* parameter.

NOSUP

See the ACTION=NOSUP parameter.

NUCEP=*name*

NUCEP=(*name*[,*start*[,*end*]])

NUCMOD=*name*

NUCMOD=(*name*[,*start*[,*end*]])

For an error event or PER trap, monitors modules in the nucleus. There is no difference between the NUCMOD and NUCEP parameters. The values are:

name

The module name. The *name* is 1 to 8 characters. If the last character is an

SLIP Command

asterisk (*), SLIP interprets the asterisk as X'C0'. (Certain module names end with the character X'C0'.) If only *name* is specified, the range of monitoring is the entire module.

start

end

Offsets from the start of the module; they indicate the start and end of the range to be monitored. The *start* must be less than or equal to the *end*. For more information, see the notes for the RANGE parameter. If you specify only *start*, the range consists of that single address.

For an error event trap or IF,SBT PER trap, NUCMOD or NUCEP establishes the range of addresses to be monitored.

For an SA or SAS PER trap, NUCMOD or NUCEP indicates that the storage alteration must be caused by an instruction within the nucleus module or within the specified range.

For an enabled, unlocked, task mode error that is matched against an error event trap, the RBLEVEL parameter controls the PSW address that is used to decide if NUCMOD or NUCEP matches.

Abbreviation: N

Example:

NUCMOD=(IEAVRTS,C4)

OK

For a SLIP SET trap, omits checking that could result in WTOR messages IEE604D and IEE831D. IBM recommends that you use the OK parameter only when issuing SLIP from a parmlib member.

PRCNTLIM=*p*

For a PER trap, specifies a software limit for PER processing by indicating a maximum percentage of system time that can be devoted to processing caused by PER interruptions. At least 33.55 seconds must have elapsed since the first PER interruption before a trap will be disabled because of this limit.

The range of integers for *p* is 1 to 99. You should use caution in specifying 99 because it means that no percent limit checking is done.

Use PRCNTLIM to tell SLIP processing to disable a SLIP trap when both of the following occur:

- A PER interrupt occurs.
- SLIP determines that the specified PRCNTLIM value has been reached.

The value computed to test PRCNTLIM is an approximation. SLIP makes this calculation only when a PER interrupt occurs, so the PRCNTLIM parameter does not cause the trap to be disabled until a PER interrupt occurs.

If you omit PRCNTLIM, the default is 10.

Abbreviation: PL

Example:

PRCNTLIM=20

PSWASC=*mode*

PSWASC=(*mode*[,*mode*]...)

For an error event or PER trap, specifies the PSW address space control (PSWASC) mode the system must be in for the trap to match. The *mode* is:

HOME	Home ASC mode
PRIMARY	Primary ASC mode
SECONDARY	Secondary ASC mode
AR	Access register ASC mode

One or more modes can be specified; the trap will match if the PSWASC mode is one of the specified values.

Allowable Abbreviations:

PA for PSWASC
H for HOME
P for PRIMARY
S for SECONDARY

Example:

PSWASC=(AR,S)

PVTEP=*name***PVTEP=(*name*[,*start*[,*end*]])**

For an error event or PER trap, requests that SLIP monitor modules in the private area. The values are:

name

The entry point name or alias. The *name* is 1 to 8 characters. If the last character is an asterisk (*), SLIP interprets the asterisk as X'C0'. (Certain module names end with the character X'C0'.) If only *name* is specified, the range of monitoring is from the entry point or alias to the end of the module.

'name'

The last characters of a posix executable file. The '*name*' is 1 to 80 characters. It may contain any characters, and it is case sensitive. PVTMOD or PVTEP may be used interchangeably with the same results.

Note: If entering the slip trap through a parmlib member (such as ieaslpxx) then you may not be able to enter 80 characters because the PVTMOD/EP parameter must fit on a line.

All the rules that apply to PVTMOD/PVTEP processing apply here, except that only the last characters of posix executable files are compared with the specified '*name*' to determine whether a match has occurred. For example, if PVTMOD='n/sh' and the path name of the file being matched with is /bin/sh, a match will occur.

*start**end*

Offsets from the entry point or alias; they indicate the start and end of the range to be monitored. The *start* must be less than or equal to the *end*. For more information, see the notes for the RANGE parameter. If you specify only *start*, the range consists of that single address.

For an error event trap and an IF or SBT PER trap, PVTEP establishes the range of addresses to be monitored.

For an SA or SAS PER trap, PVTEP indicates that the storage alteration must be caused by an instruction within the specified range.

SLIP Command

For an enabled, unlocked, task mode error that is matched against an error event trap, the RBLEVEL parameter controls the PSW address that is used to decide if PVTEP matches.

When an error event or PER interruption occurs, SLIP searches for the private module first in the current primary address space and then in the dispatched address space. A match occurs only when the address of the instruction with the error or interrupt is found within the boundaries of this copy of the module.

Notes:

1. To check in a private area module, SLIP must obtain the local lock when PVTEP or PVTMOD is specified for any trap other than a PER SBT trap or a PER IF trap.
2. SLIP is capable of monitoring a single PER range. When a duplicate module is found, SLIP will recognize only the first module and will not notify the user of the possible conflict.

Example:

PVTEP=(MYEP,10,40)

PVTMOD=*name*
PVTMOD=(*name[,start[, end]]*)
PVTMOD (or PVTEP) ='*name*'
PVTMOD (or PVTEP) =('i*name[,start[, end]]*')

For an error event or PER trap, monitors modules in the private area. The values are:

name

The module name or alias. The *name* is 1 to 8 characters. If the last character is an asterisk (*), SLIP interprets the asterisk as X'C0'. (Certain module names end with the character X'C0'.) If *name* is an alias, SLIP processing uses the load module name. If only *name* is specified, the range of monitoring is the entire module.

'*name*'

The last characters of a posix executable file. The '*name*' is 1 to 80 characters. It may contain any characters, and it is case sensitive. PVTMOD or PVTEP may be used interchangeably with the same results. All the rules that apply to PVTMOD/PVTEP processing apply here, except that only the last characters of posix executable files are compared with the specified '*name*' to determine whether a match has occurred. For example, if PVTMOD='n/sh' and the path name of the file being matched with is /bin/sh, a match will occur.

start

end

Offsets from the start of the module; they indicate the start and end of the range to be monitored. The *start* must be less than or equal to the *end*. For more information, see the notes for the RANGE parameter. If you specify only *start*, the range consists of that single address.

Private area modules can reside (starting at different addresses) in several address spaces. To resolve the range of addresses to be monitored for PVTMOD to a particular address space, use either the JOBNAME parameter with MODE=HOME or the ASID parameter. If neither parameter is specified and the module is loaded into several address spaces, the system might resolve the PER range to any one of those address spaces.

For an error event trap and an IF or SBT PER trap, PVTMOD establishes the range of addresses to be monitored.

For an SA or SAS PER trap, PVTMOD indicates that the storage alteration must be caused by an instruction within the specified range.

For an enabled, unlocked, task mode error that is matched against an error event trap, the RBLEVEL parameter controls the PSW address that is used to decide if PVTMOD matches.

To check for a private area module, SLIP must obtain the local lock. SLIP cannot obtain the local lock, it cannot check the private area module, and the trap will not match.

On an error or interrupt, SLIP searches for the private module first in the current primary address space and then in the dispatched address space. A match occurs only when the address of the instruction with the error or interrupt is found within the boundaries of this copy of the module.

Notes:

1. To check in a private area module, SLIP must obtain the local lock when PVTEP or PVTMOD is specified for any trap other than a PER SBT trap or a PER IF trap.
2. SLIP is capable of monitoring a single PER range. When a duplicate module is found, SLIP will recognize only the first module and will not notify the user of the possible conflict.
3. PVTMOD processing does not support modules brought into storage using the LOAD macro with the ADDR parameter. To monitor those modules, use the RANGE parameter instead of PVTMOD.

Abbreviation: P

Example:

PVTMOD=(MYMOD,1C,1F)

RANGE=*start*

RANGE=(*start,end*)

For a PER trap, specifies the starting and ending addresses of virtual storage to be monitored. The addresses can be either direct or indirect. For indirect addressing, see "Indirect Addresses" on page 4-508.

If you specify only *start*, the range consists of that 1 byte. If the starting address is greater than the ending address, the addresses wrap around.

RANGE is not valid for error event traps. RANGE cannot be specified on an ACTION=IGNORE storage alteration PER trap.

The following notes apply to all SLIP parameters that have *start[,end]* options to indicate a range.

Notes:

1. Consider the range carefully on any PER trap. A wide range could cause performance to degrade because of the processing overhead for many PER interrupts. For example, for an address range that wraps storage, such as (700,600), PER events might occur too fast for the system to disable the trap. If this happens, manually reset control registers 9, 10, and 11 to zero. This disables PER and also defines a minimum address range.

SLIP Command

2. For successful branch monitoring, hardware PER processing does not check the address range specified on the LPAEP, LPAMOD, NUCEP, NUCMOD, PVTEP, PVTMOD, and RANGE parameters. Therefore, a branch taken by an instruction anywhere in the system would cause a successful branch PER interrupt.

To simulate successful branch monitoring for an address range, SLIP initially sets up instruction fetch monitoring for the desired address range. Then, when the processor gets to an instruction within the requested range (indicated by an instruction fetch PER interrupt), SLIP automatically switches PER monitoring to successful branch mode. Thus, the branch into the range does not cause a PER interrupt and does not match the trap for that instruction.

You should be aware that the first PER event that occurs when the processor enters the requested range may not be a successful branch event. This *extra* instruction fetch event might affect values you supplied for other parameters, such as MATCHLIM. When the processor leaves the requested range, PER monitoring returns to instruction fetch monitoring on the range, thus avoiding unnecessary PER interrupts. If the instructions being monitored are enabled for I/O and/or external interrupts, control may leave and then re-enter the monitored range due to normal interrupt processing.

3. Mode switching does not occur for successful branch PER traps with ACTION=IGNORE specified. This means that if the initial entry into a monitored area matches an IGNORE trap, the mode remains instruction fetch and the *extra* event is delayed.

For successful branch monitoring, if an Execute instruction has a successful branch target, the location of the Execute instruction is used to determine whether or not the branch was within the monitored area without regard to the location of the executed branch.

For an SAS PER trap, a storage alteration by a STURA instruction at any address is accepted.

Abbreviation: RA

Example:

RANGE=(600,700)

RBLEVEL=ERROR
RBLEVEL=NOTSVRB
RBLEVEL=PREVIOUS

For an error event trap, indicates the request block (RB) that contains the registers and PSW of interest for a particular error. SLIP uses the following identified by RBLEVEL:

- PSW when processing the ADDRESS, LPAEP, LPAMOD, PVTEP, PVTMOD, MODE, NUCEP, and NUCMOD parameters
- Registers when processing the DATA, TRDATA, LIST, SUMLIST, and TRDATA parameters

RBLEVEL applies only to unlocked, task mode errors.

ERROR

The PSW is obtained from the request block (RB) prior to the ABEND RB. The registers are obtained from the ABEND RB.

PREVIOUS

The RBs used are each one RB prior to the RBs used for ERROR.

NOTSVRB

The PSW is obtained from the most recent non-SVRB and the registers are obtained from the associated SVRB.

Abbreviation: RB

Example:

RBLEVEL=NOTSVRB

REASON=code

For an error event trap, specifies a user or system reason code to be associated with the error in the accompanying COMP parameter. The REASON parameter cannot be coded without the COMP parameter.

For the reason code, enter 1 to 8 hexadecimal digits. If the code is fewer than 8 digits, the system pads it on the left with zeroes. For example, REASON=4 means a reason code of 00000004.

You can indicate a set of reason codes by substituting x's for up to 7 digits. For example, REASON=44XXXX means any reason code that begins with the digits 0044; the last 4 digits can be any hexadecimal value. You can use x's in any position.

Note: To match the REASON parameter, the reason code must have been specified via the REASON parameter of the ABEND, SETRP, or CALLRTM macro.

Abbreviation: RE

Example:

COMP=U123x,REASON=8

RECORD

See the ACTION=RECORD parameter.

RECOVERY

See the ACTION=RECOVERY parameter.

REFAFTER

See the ACTION=REFAFTER parameter.

REFBEFOR

See the ACTION=REFBEFOR parameter.

REFAFTER=(triplet)**REFAFTER=(triplet[,triplet]...)****REFBEFOR=(triplet)****REFBEFOR=(triplet[,triplet]...)**

As an option of an ACTION parameter, specifies the refresh to be taken after or before the action specified for the SLIP trap. The parameter must be specified if the ACTION parameter includes REFATER or REFBEFOR.

The following syntax for a triplet is identical for REFATER and REFBEFOR.

target[(b)] ,EQ|EQA|EQC[(n)] ,value

The parameters in a *triplet* are:

SLIP Command

target

Specifies the first triplet operand, which could be the address of a storage location or a general purpose register (GPR), to be refreshed with the supplied value. The target can be:

- A direct address (virtual address) of 1 to 8 hexadecimal digits
- A general purpose register, xR , where x is 0 through 15
- An indirect address (see “Indirect Addresses” on page 4-508)

b

If specified, b modifies the target address by indicating the starting bit for a binary refresh. For registers, b can be 0 through 31. The starting bit position plus the bit size for the refresh must not exceed 31.

EQ

Specifies the second triplet operand, which refreshes the contents of the target address with a binary or hexadecimal value.

EQA

Refreshes the target address with the address specified in the *value* parameter. A binary refresh must not be specified with EQA.

EQC

Refreshes the contents of the address specified as the target with the contents of the address specified in the *value* parameter.

n

Specifies the number of bytes or bits processed for a contents (EQC) or address (EQA) refresh. When b is specified with the target, the range is from 1 to 8 bits with the default being 1 bit. Otherwise, the range is 1 to 4 bytes with the default being 4 bytes.

Note: When SLIP does a contents or an address refresh, it refreshes the first n bytes of storage and the last n bytes of a register.

value

Specifies the third triplet operand. The *value* refreshes the target address. When the **EQ** parameter is specified without A or C, the value can be:

- A constant, whose length determines the number of bytes or bits to be refreshed.
- Binary digits, if b is specified with the target address. If b is not specified, the value is in hexadecimal digits. For example, 6R(0),EQ,01 is binary and 6R,EQ,01 is hexadecimal. For binary refreshes, the length of the value establishes the length of the refresh.
 - The maximum length for a binary refresh is 8 bits.
 - Binary refreshes can cross byte boundaries but not register boundaries.
- Hexadecimal digits, whose length determines the number of bytes to be refreshed, when the target is not a register. However, the maximum length of the value is 4 bytes. If you specify B36,EQ,8AD62, two and a half bytes of data are refreshed starting at location X'B36'.

If the target address is a register, the length of the refresh is 4 bytes, and the value is right-justified. For example, 4R,EQ,8 is equivalent 4R,EQ,00000008.

If **EQA** or **EQC** is specified, the value can be:

- A direct address (virtual address) of 1 to 8 hexadecimal digits
- A general purpose register, xR or xG , where x is 0 to 15
- An indirect address (see “Indirect Addresses” on page 4-508)

Considerations for storage refreshes:

- SLIP processes the PER interrupt after the instruction has fully completed, except possibly for MVCL or CLCL instructions. The refresh will occur at that time.
- Storage being modified **must** be paged in when the modification occurs.
- Refreshes are processed one triplet at a time until one fails, at which time the refreshing processing stops.
- Use REFBEOF when you need to refresh storage on which some subsequent SLIP action may depend; use REFAFTER otherwise.
- Low storage refreshes, for addresses 0 to X'1FF', are allowed if direct addressing is used.

Abbreviation: RFA or RFB

Examples:

Example 1: In the following trap, every time the contents of locations X'10000' to X'10003' in the address space for a job named CONS is modified, the trap causes the current contents to be recorded in a trace record and then stores X'00000A24' into location X'10000'.

```
SLIP SET,SA,RANGE=(10000,+3),ACTION=(REFAFTER,TRACE),
TRDATA=('CONS'.10000,+3),
REFAFTER=('CONS'.10000,EQ,00000A24),ASIDSA='CONS',END
```

Example 2: When the trap matches, an SVC dump is requested the storage will contain the value of X'3000' in address space A, then storage locations X'3000' and X'3001' are modified and register 1 is set to 0.

```
SLIP SET,IF,LPMOD=(MYMOD,40,42),ACTION=(SVCD,REFAFTER),
SUMLIST=(0FC.3000,4000),REFAFTER=(0FC.3000,EQC(2),
0FD.4000,1R,EQ,00000000),END
```

Example 3: The following trap sets the first 3 bits of location '3000'X in the home address space of DUMPSRV to '101'B.

```
SLIP SET,IF,LPMOD=(MYMOD,40,42),ACTION=(SVCD,REFBEFOR),
REFBEFOR=('DUMPSRV'.2R?(0),EQ,101),END
```

Example 4: The low-order 3 bytes of the virtual address are generated by adding X'30' to the value in general purpose register (GPR) 2 will be stored in DUMPSRV at location X'3001'.

```
SLIP SET,IF,LPMOD=(MYMOD,40,42),A=(SVCD,REFBEFOR),
REFBEFOR=('DUMPSRV'.3001,EQA(3),2R?+30),END
```

REMOTE=(UNCOND | COND,*remote*)
REMOTE=(UNCOND | COND,(*remote*)[,(*remote*)]...)
REMOTE=(*remote*)
REMOTE=((*remote*)[,(*remote*)]...)

As an option of an ACTION parameter, allows SLIP to specify actions to be taken within the sysplex, on systems other than the system on which the trap matches. A SLIP trap on one system can initiate an SVC dump or load a wait state on another system. The REMOTE parameter values specify other system(s) in the sysplex, actions for those systems, and options for dumps on those systems. The REMOTE parameter can be specified only when the ACTION for the local system is SVCD, SYNCVCD, or WAIT.

SLIP Command

The parameters within the REMOTE parameter are: UNCOND, COND, SYSLIST, ACTION, ASIDLST, DSPNAME, JOBLIST, LIST, SDATA, and STRLIST.

For a system running z/OS V1R2 or higher in z/Architecture mode, the remote LIST parameter can contain 64-bit addresses. However, if the LIST parameter is sent to either a pre-z/OS V1R2 system, or to a z/OS V1R2 system running in ESA mode, the entire remote LIST specification is ignored.

UNCOND

COND

On a REMOTE parameter, indicates if the remote actions should be performed conditionally or unconditionally when the trap matches. COND or UNCOND must be the first value specified on the REMOTE parameter. If you omit UNCOND, you do not need to code a comma in its place.

You can specify COND only on a PER trap when the action for the local system is ACTION=WAIT. Use COND when it is more important that a unit of work be stopped than for the action on the remote system to occur.

When COND is specified:

- When the SLIP trap matches for a PER interrupt and the task is enabled, unlocked, and in task mode, then the task is stopped, the actions on the other systems in the sysplex are scheduled, and the local system is put into a restartable wait state.
- When the SLIP trap matches for a PER interrupt and the task cannot be safely stopped, the actions on the other systems are ignored and the local system is put into a restartable wait state.

When UNCOND is specified and the SLIP trap matches for a PER interrupt:

- The actions on the other systems in the sysplex are scheduled.

SYSLIST=([sysname,group.member,group.*,...])

On a REMOTE parameter, identifies systems in the sysplex on which the actions specified in *remote* will be performed. You can specify any combination of system names and/or member specifications. When group.* is specified, all systems where any member of the group is running are affected. If you omit SYSLIST, the default is all systems. When a system is identified more than once, implicitly or explicitly, the first occurrence is used, the others are ignored.

Examples:

```
SYSLIST=(sys1)
SYSLIST=(sys1,sys2,sys3)
SYSLIST=(mygroup.member1,hisgrp.*)
SYSLIST=(sys1,group1.memberA,thegroup.member2)
```

ACTION=SVCD

ACTION=WAIT

On a REMOTE parameter, identifies the action to be taken by the systems identified in SYSLIST: to initiate an SVC dump or load a wait state. The dump options are ASIDLST, JOBLIST, DSPNAME, LIST, and SDATA. Dump options are processed only when the action is specified as SVCD or is the default.

When ACTION is not specified within the REMOTE parameter:

- If the local action is WAIT, the default action is WAIT.
- If the local action is SVCD or SYNCVCD, the default action is SVCD.
All systems identified in SYSLIST use the default SLIP SVCD parameters as their default dump options.

ASIDLST, DSPNAME, JOBLIST, LIST, SDATA, and STRLIST

On a REMOTE parameter, the syntax is identical to the parameters for the dump option on the local system. When specified without an equal sign (=) and value, the options specified for the local system are used for the systems identified in SYSLIST.

Note: The only symbolic ASIDs accepted for the ASIDLST parameter are PRIMARY or CURRENT.

If no options are specified for the subparameters in the REMOTE parameter, the systems identified in SYSLIST use the options of the local system. For example, if you issue the SLIP SET,...,SDATA=(SQA),RM=(SDATA),END command, the systems identified in SYSLIST would dump SQA for SDATA.

Allowable Abbreviations:

RM for REMOTE
SY for SYSLIST

Examples:

```
REMOTE=(COND,(SYSLIST=MYGROUP.* ,ACTION=WAIT))
REMOTE=((SYSLIST=(SYS1,SYS2),ACTION=WAIT),(SYSLIST=SYS3,ACTION=SVCD))
REMOTE=(COND,ACTION=SVCD,JL=(J1,J2),AL=(5,6),SDATA=COUPLE)
REMOTE=(ACTION=SVCD,SDATA,DSPNAME)
```

SA

Specifies the event as a storage alteration. This parameter is positional; it must appear following SET and a comma. See “ASIDSA parameter” on page 4-537 for further information.

SAS

Specifies the event as a storage alteration caused by a STURA instruction. This parameter is positional; it must appear following SET and a comma. See “ASIDSA parameter” on page 4-537 for further information.

SBT

Specifies the event as a successful branch caused by a branch instruction into the specified range or a branch within the specified range. This parameter is positional; it must appear following SET and a comma.

SDATA=area**SDATA=(area[,area]...)**

As an option of an ACTION or REMOTE parameter, specifies the kind of system areas to dump. You can specify any combination of the following, enclosed in parentheses and separated by commas:

- ALLNUC
- ALLPSA
- COUPLE
- CSA
- GRSQ
- LPA
- LSQA
- NOALLPSA
- NOSQA
- NOSUMDUMP
- NUC
- PSA
- RGN
- SQA

SLIP Command

SUMDUMP
SWA
TRT
WLM
XESDATA

If you specify only one area, you can omit the parentheses. See the CHNGDUMP command in this book for descriptions of these dump options. Note, however, that the CHNGDUMP command does not affect SLIP processing of the SDATA parameter.

If you specify SDATA, the default SDATA options are:

- For ACTION=SVCD or SYNCVCD: SQA.
- For ACTION=STDUMP or TRDUMP: NOSQA.

If you do not specify SDATA, the default SDATA options are:

- For ACTION=SVCD or SYNCVCD: ALLPSA, CSA, LPA, NUC, RGN, SQA, SUMDUMP, and TRT.
- For ACTION=STDUMP or TRDUMP: NOALLPSA, NOSQA, NOSUM, and TRT.

SDATA options are added and overridden according to installation-defined defaults.

SDATA options override the installation-defined defaults set by the CHNGDUMP command for any parameters that can be specified. Also, although it can not be specified via the SLIP command, SDATA=Servers is always used for SLIP.

Allowable Abbreviations:

SD for SDATA
NOALL for NOALLPSA
NOSUM for NOSUMDUMP
SUM for SUMDUMP

Example:

ACTION=SVCD, SDATA=(SQA, TRT, SUM)

SET

Specifies that the SLIP command sets a trap. This parameter is positional; it must appear following a blank after SLIP.

STDUMP

See the ACTION=STDUMP parameter.

STRACE

See the ACTION=STRACE parameter.

STRLIST=(*s-option*)

STRLIST=(*s-option*[,*s-option*]...)

As an option of an ACTION or REMOTE parameter, includes in the dump one or more coupling facility structures. A structure is identified in a *s-option* value, which consists of the following parameters; STRNAME is required in STRLIST, the other parameters are optional.

STRNAME=*strname*

In the STRLIST parameter, designates a coupling facility list or cache structure. The *strname* is the name of the structure to be included in the dump. The *strname* is 1 to 15 characters and must begin with a letter.

Any dump options for this structure are replaced when you issue this SLIP command.

Abbreviation: STRNM

CONNNAME=*connname*

In the STRLIST parameter for a coupling facility cache structure, requests the user registry information for this user be included in the dump. The *connname* is the name of a connected user. If the connected user represented by the *connname* does not exist, the dump will not contain user registry information.

Abbreviation: CONNM

ACCESSTIME=ENFORCE

ACCESSTIME=NOLIMIT

In the STRLIST parameter, indicates whether the dump time limit specified on the ACESSTIME parameter of the IXLCONN macro is in effect.

When ACESSTIME=ENFORCE is specified, the system holds structure dump serialization no longer than the time interval specified on the IXLCONN macro. This is the default. If ACESSTIME=0 is specified on the IXLCONN macro and ACESSTIME=ENFORCE is specified on the SLIP command, the structure will not be included in the dump.

When ACESSTIME=NOLIMIT is specified, the dump time limit is not in effect and the system will hold structure dump serialization until processing is completed.

Allowable Abbreviations:

ACC for ACESSTIME

ENF for ENFORCE

NOLIM for NOLIMIT

LOCKENTRIES

In the STRLIST parameter for a coupling facility list structure, the system includes in the dump the lock table entries for the requested structure. Because coupling facility cache structures do not have lock table entries, this parameter is ignored when specified for a coupling facility cache structure.

Abbreviation: LOCKE

USERCNTLS

In the STRLIST parameter, requests that the user attach controls be included in the dump.

Abbreviation: UC

COCLASS=ALL

COCLASS=(/list)

In the STRLIST parameter for a coupling facility cache structure, specifies which cast-out classes are included in the dump. For each cast-out class, the cast-out class controls are dumped and the directory information for each of the entries within the requested cast-out classes are dumped (if SUMMARY is not specified).

COCLASS is not valid for a coupling facility list structure.

When COCLASS=ALL is specified, the cast-out class controls for all cast-out classes are dumped along with the directory information for all entries within the classes (if SUMMARY is not specified).

SLIP Command

When COCLASS=(*list*) is specified, the cast-out class controls for (*list*) are dumped along with the directory information for the entries in the requested cast-out classes (if SUMMARY is not specified). The values specified in a range are the decimal cast-out class values in the range 0 to 65535. When a requested class does not exist, it is not dumped.

The *list* represents a list of values, ranges of values, or values and ranges of values, in any combination. For example:

(start1-end1,value2,start3-end3, ...)

Abbreviation: COC

STGCLASS=ALL

STGCLASS=(*list*)

In the STRLIST parameter for a coupling facility cache structure, specifies which storage classes are included in the dump. For each storage class, the storage class controls are dumped and the directory information for each of the entries within the requested storage classes are dumped (if SUMMARY was not specified).

COCLASS is not valid for a coupling facility list structure.

When STGCLASS=ALL is specified, the storage class controls for all storage classes are dumped along with the directory information for all entries within the classes (if SUMMARY is not specified).

When STGCLASS=(*list*) is specified, the storage class controls for (*list*) are dumped along with the directory information for the entries in the requested storage classes (if SUMMARY is not specified). The values specified are the decimal storage class values, 0 to 255. When a requested class does not exist, it is not dumped.

The *list* represents a list of values, ranges of values, or values and ranges of values, in any combination. For example:

(start1-end1,value2,start3-end3, ...)

Abbreviation: SC

LISTNUM=ALL

LISTNUM=(*list*)

In the STRLIST parameter for a coupling facility list structure, specifies which lists are included in the dump. The list controls are dumped along with the entry controls for the entries on each requested list (if SUMMARY is not specified).

LISTNUM is not valid for a coupling facility cache structure.

When LISTNUM=ALL is specified, the list controls for all lists in the coupling facility list structure are dumped along with the entry controls (if SUMMARY is not specified).

When LISTNUM=(*list*) is specified, the list controls for (*list*) are included in the dump along with the entry controls for those lists. The values specified are the decimal list values, 0 to 4294967295. The system ignores a zero, but does not treat a zero as an error. When a requested list does not exist, it is not dumped.

The *list* represents a list of values, ranges of values, or values and ranges of values, in any combination. For example:

(start1-end1,value2,start3-end3, ...)

Abbreviation: LNUM

ADJUNCT=CAPTURE

ADJUNCT=DIRECTIO

In the STRLIST parameter, requests that the adjunct data for each entry specified by the range be included in the dump. When this parameter is not specified or when adjunct data does not exist for this structure, the adjunct data is not included in the dump.

ADJUNCT may not be specified with SUMMARY.

When ADJUNCT=CAPTURE is specified, the adjunct data is captured in the facility dump space along with the directory information while dumping serialization is held.

When ADJUNCT=DIRECTIO is specified, the adjunct data is written directly to the dump data set after the directory information is captured. The adjunct data is not captured in the structure dump table. Note that the adjunct data may be changing as dumping proceeds.

Allowable Abbreviations:

ADJ for ADJUNCT

CAP for CAPTURE

DIO for DIRECTIO

ENTRYDATA=UNSERIALIZE

ENTRYDATA=SERIALIZE

In the STRLIST parameter, indicates that the entry data for each entry within the requested range is included in the dump. When this parameter is not specified or when entry data does not exist for the structure, entry data is not included in the dump.

ENTRYDATA may not be specified with SUMMARY.

When ENTRYDATA=UNSERIALIZE is specified, the entry data is dumped after structure dump serialization is released. Note that the entry data may be changing relative to the entry controls that were captured while structure dump serialization was held.

When ENTRYDATA=SERIALIZE is specified, the entry data is dumped while serialization is held. If ACESSTIME=ENFORCE is specified and the dump time limit expires before the entry data is written to the dump data set, the system continues to write the entry data to the dump data set even though serialization is not held.

Allowable Abbreviations:

EDATA for ENTRYDATA

UNSER for UNSERIALIZE

SER for SERIALIZE

SUMMARY

In the STRLIST parameter, requests a summary of the range of classes or lists that is dumped. The directory information for the entries is excluded from the dump.

SUMMARY may not be specified with ADJUNCT or ENTRYDATA.

Abbreviation: SUM

Notes for the STRLIST Parameter:

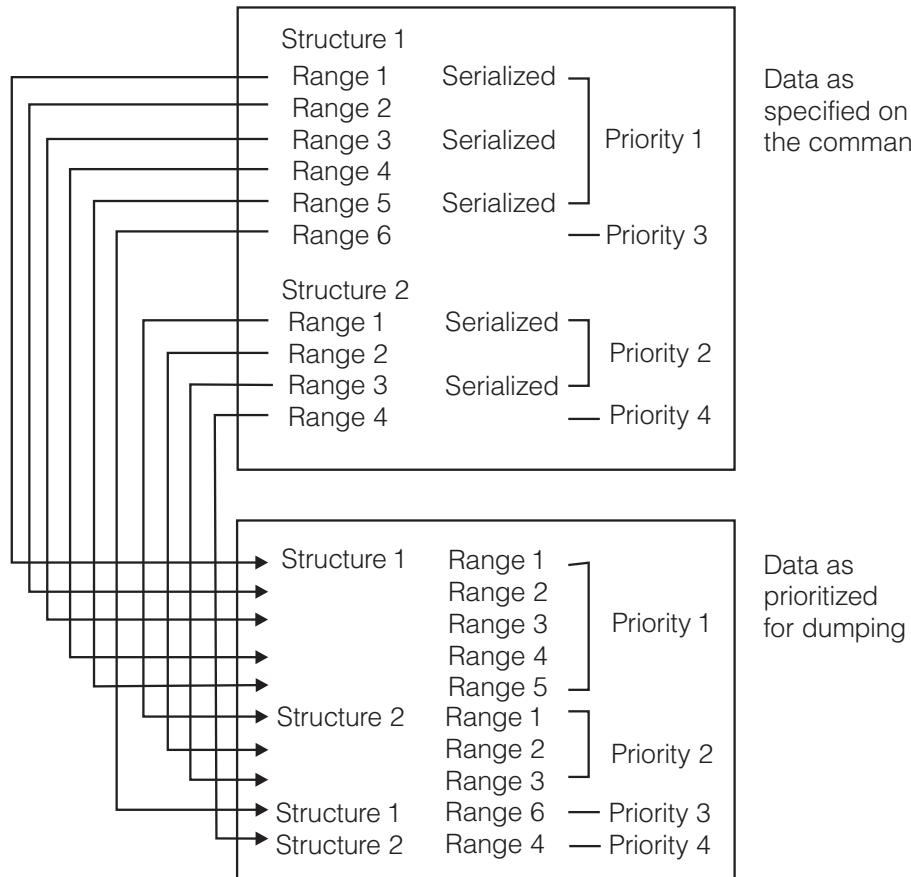
1. A syntax error message is issued if STRNAME is not the first parameter in STRLIST.

SLIP Command

2. If CONNAME and ACESSTIME are specified more than one time for a structure, the first CONNAME and the last ACESSTIME are used.
3. When a list number, a storage class, a cast-out class, or an entry is specified in the STRLIST more than once, it will be dumped more than once. An example of this is when STGCLASS=ALL is specified with COCLASS=ALL. All entries in the coupling facility cache structure are dumped twice. Once grouped by storage class and again grouped by cast-out class.
4. When LISTNUM, STGCLASS, or COCLASS is **not** specified, no list or class controls are dumped and no entries are dumped.
5. If a large amount of data is requested to be dumped, the system may not be able to dump all the data completely. You can expect to dump up to a maximum of 47 structures, if you specify no more than six ranges. If you must specify more than six ranges, you must specify fewer structures. For each structure less than 47 that you specify, you can specify another 10 ranges, as follows:

Number of Structures	Number of Ranges
47	6
46	16
45	26
44	36
:	:

6. If the system cannot dump all the data you requested, it prioritizes the data according to your specifications on the command.



- a. The system will attempt to dump the first requested structure first.
 - Within that structure, the system processes the LOCKENTRIES, USERCNTLS, COCLASS, STGCLASS, and LISTNUM parameters in the order that they are specified. COCLASS, STGCLASS, and LISTNUM may be specified more than once for a single structure.
 - The system dumps requested serialized data before requested unserialized data, starting with the first requested data in the structure and proceeding through the last data that was requested as *serialized*.
- b. The system then dumps the next-requested structure data, starting with the first requested data in the structure and proceeding through the last data that was requested as *serialized*.
- c. The system continues in this manner until all *serialized* data in all requested structures has been prioritized for dumping.
- d. The system then dumps any remaining data that was requested as *unserialized* and that may not have been dumped, beginning with the first-requested structure.

SUMLIST=(*start,end*)

SUMLIST=(*start,end[,start,end]...*)

As an option of an ACTION parameter, specifies one or more storage areas to be included in a disabled summary dump. Each area is defined by a set of starting and ending addresses, which can be either direct or indirect. (See "Indirect Addresses" on page 4-508.) A starting address must be less than or equal to the ending address.

SLIP Command

Specify SUMLIST with ACTION=SVCD or ACTION=TRDUMP when the SUMDUMP option is specified or is the default in the SDATA parameter.

SUMLIST is similar to the LIST parameter; the area specified by LIST is in a scheduled SVC dump while the area specified by SUMLIST is in a disabled summary dump.

Abbreviation: SL

Example:

```
ACTION=SVCD,SUMLIST=(152,155,2R%,+3)
```

SVCD

See the ACTION=SVCD parameter.

SYNCSVCD

See the ACTION=SYNCSVCD parameter.

TARGETID

See the ACTION=TARGETID parameter.

TARGETID=*trapid*

As an option of an ACTION parameter, specifies the trap to be activated when the current trap is deactivated as a result of MATCHLIM. The *trapid* identifies the next PER trap to be activated, where 1 to 4 alphanumeric or national (\$, #, @) characters specifies the trap identifier. For example:

```
TARGETID=PER2.
```

Specifying TARGETID=*trapid* on the first PER trap associates the trap with the SLIP trap ID of the second trap. The second PER trap can specify a third trap and so on. There is no limit to the number of traps in a chain of dynamic traps, called a *dynamic PER activation chain*. The range of the target trap is evaluated in the matching environment of the prior trap in the activation chain. Storage areas associated with RANGE **must** be paged in when the match occurs.

Indirect addressing on the RANGE parameter allows storage locations and registers to be resolved when their values are known.

When TARGETID is used as a parameter, it must also be specified as one of the ACTION values. Failure to do so will result in the TARGETID parameter being ignored.

Abbreviation: TI

Example 1

The following dynamic per trap example can do alterations on the first X'20' bytes of the storage obtained by the GETMAIN command. Assume that the GETMAIN command is issued prior to location X'100' in module MYMOD, such that the address of the obtained storage is in GPR 1.

```
SLIP SET,IF,PVTMOD=(MYMOD,100),DISABLE,ACTION=(TARGETID,TRACE),
      TARGETID=H2,ID=H1,ASID=(0FC),END
```

```
SLIP SET,SA,ASIDSA=(0FC),RANGE=(1R?,+20),
      ACTION=WAIT,ID=H2,DISABLE,END
```

Example 2

This example uses indirect addressing on the RANGE keyword to trap on alterations to one byte of storage. The address is found at offset X'50' into the CTV. See "Dynamic PER Traps" on page 4-518 for more details.

```
SLIP SET,SA,RANGE=10?+50?,A=WAIT,JOBNAME=CONS,END
```

TRACE

See the ACTION=TRACE parameter.

TRDATA=(STD[,REGS][,/list])

TRDATA=(REGS[,/list])

TRDATA=(list)

As an option of an ACTION=TRACE or ACTION=TRDUMP parameter, indicates the type and contents of the generalized trace facility (GTF) records to be collected in the requested GTF trace.

STD

Indicates that GTF is to write a SLIP standard trace record, as described in *z/OS MVS Diagnosis: Tools and Service Aids*.

REGS

Indicates that the SLIP trace records are to contain the contents of the 16 general purpose registers at the time of the error event or PER interruption.

list Specifies one or more sets of addresses for one or more storage areas to dump. Each area cannot exceed 65535 bytes. The addresses can be direct or indirect. For each set, the starting address must be less than or equal to the ending address. (See "Indirect Addresses" on page 4-508.)

Abbreviation: TD

Example:

```
SLIP ACTION=TRACE,TRDATA=(STD,REGS,152,155)
```

TRDUMP

See the ACTION=TRDUMP parameter.

WAIT

See the ACTION=WAIT parameter.

SLIP SET Examples

Example 1

This example establishes an enabled SLIP trap with an ID of 0002. It requests an SVC dump (by default) if there is an 0C4 program check interruption while module MOD01 and job JOBXYZ are in control.

```
SLIP SET,ENABLE,ID=0002,COMP=0C4,ERRTYP=PROG,JOBNAME=JOBXYZ,  
LPAMOD=MOD01,END
```

Example 2

This example sets up an error event (non-PER) trap with an ID of DUM1 that suppresses all SYSABEND dumps for the 806 system completion code.

```
SLIP SET,COMP=806,ID=DUM1,ACTION=NOSYSA,END
```

Example 3

SLIP Command

This command sets an instruction fetch PER trap that will cause an SVC dump when the instruction at CD3100 is executed. PER monitoring will be active in all address spaces in the system because neither ASID nor JOBNAMES was specified.

```
SLIP SET,IF,ENABLE,ACTION=SVCD,RANGE=CD3100,END
```

Example 4

This example sets up a trap for a successful branch trace of the path taken through the LPA module MOD01 starting at offset 108 (hex) through 4FC during the execution of JOBX. After 20 standard SLIP records have been written the trap is automatically disabled.

```
SLIP SET,SBT,ENABLE,ID=PER1,LPAMOD=(MOD01,108,4FC),JOBNAMES=JOBX,  
ACTION=TRACE,MATCHLIM=20,END
```

Example 5

This command defines a trap that will cause an SVC dump when storage location CD3010 is altered. MATCHLIM is 1 and PRCNTLIM is 10 by default.

```
SLIP SET,SA,ENABLE,ACTION=SVCD,RANGE=CD3010,END
```

Example 6

This example sets up two SLIP traps, both initially disabled, and then enables them both with a SLIP MOD command. The second trap prevents the first trap from matching for the subset of instructions specified by LPAMOD.

```
SLIP SET,IF,DISABLE,LPAMOD=(MODX,110,1FB),JOBNAMES=JOB1,MATCHLIM=500,  
ACTION=TRACE,TRDATA=(STD,REGS),END  
SLIP SET,IF,DISABLE,LPAMOD=(MODX,1C4,1D7),ACTION=IGNORE,END  
SLIP MOD,ENABLE,ALL
```

Note: The IGNORE trap is specified after the non-IGNORE trap because traps are processed for match tests in last-in, first-out order.

Example 7

The SLIP trap will match on a branch instruction into LPA module MODZ starting at offset 220 (hex) through 240. When the trap is matched, the actions specified in the REMOTE parameter will be performed unconditionally. Systems SYS1 and SYS2 will be placed into a restartable wait state. System SYS3 will schedule an SVC dump. The dump for system SYS3 will include address spaces 3 and 4. The local system will also schedule an SVC dump. The dump for this system will include address spaces 0 and C.

```
SLIP SET,SBT,ACTION=SVCD,ASIDLST=(0,C),LPAMOD=(MODZ,220,240),  
REMOTE=(UNCOND,(SYSLIST=(SYS1,SYS2),ACTION=WAIT),  
(SYSLIST=(SYS3),ACTION=SVCD,ASIDLST=(3,4))),END
```

Example 8

The SLIP trap does not contain the parameters ASIDSA or DSSA. Normally, message IEE604D would be issued; however, because the OK parameter is also specified, SLIP will continue processing without issuing the WTOR.

```
SLIP SET,SA,OK,RANGE=(1000,2000),END
```

Example 9

This example and the next show how to use the STOPGTF parameter. This one stops GTF tracing when the instruction at offset 50 in module MYMOD is executed. In this case, the GTF records originate from some source, perhaps other than SLIP:

```
SLIP SET,IF,PVTMOD=(MYMOD,50),ACTION=STOPGTF,END
```

Example 10

Like the previous example, this example shows how to use the STOPGTF parameter. This example stops GTF tracing when the instruction at offset 50 into module MYMOD is executed. With ML=100 specified, only 100 trace records are written, after which the SLIP trap is disabled, and all GTF tracing is stopped:

```
SLIP SET,IF,PVTMOD=(MYMOD,50),ACTION=(STOPGTF,TRACE),ML=100,END
```

Example 11

This example shows how to use the GTFID parameter. In this example, the operator starts two GTF instances HM1 and HM2. When a unit of work takes an abend with code FFFx, the SLIP trap deactivates the instance HM1, leaving HM2 active:

```
START GTFX.HM1
START GTFX.HM2
SLIP SET,C=FFF,A=STOPGTF,GTFID=(HM1),E
```

Example 12

This example shows how to use the SUBTRAP parameter. When the first instruction in IEFBR14 is executed, a PER interrupt occurs and the SLIP action processor tries to determine whether trap TRP1 is a match. If it matches, a trace record is generated and no further action is taken; otherwise, the SLIP action processor checks whether TRP2 matches:

```
SLIP SET,IF,LPMOD=(IEFBR14,0),A=SVCD,ID=TRP2,E
SLIP SET,IF,DATA=(1R,EQ,0),A=(SUBTRAP,TRACE),ID=TRP1,E
```

Example 13

This example shows how to use the MSGID parameter. This trap matches on the command response from the DISPLAY TIME command:

```
IEE136I LOCAL: TIME=16.37.02 DATE=2001.102...
```

where the significant time is hour 16 (meaning the time is between 4:00 and 4:59 p.m.):

```
SLIP SET,MSGID=IEE136I,DATA=(3R?+14,EQ,F1F6),ACTION=WAIT,ML=1,END
```

Example 14

This example shows how to use 64-bit addresses on the RANGE parameter. It requests a dump when the first above-the-bar page is modified by job TEMP5:

```
SLIP SET,SA,ASIDSA=(SA),RANGE=(1_00000000,+FFF),ACTION=SVCD,JOBNAME=TEMP5,END
```

Example 15

This example shows how to include a small amount of above-the-bar virtual storage as part of the summary dump capture phase, which is captured synchronously under the unit of work that took the PER interrupt, and a larger portion as part of

SLIP Command

the non-summary dump phase. Note that there is a limit on the amount of storage that can be captured in the summary dump phase:

```
SLIP SET,IF,PVTMOD=(MYMOD,13E),SUMLIST=(6G!,+FFF),LIST=(1_00000000,2_00000000),END
```

Example 16

This is an example of a REMOTE LIST, which can be used to dump storage in the address space whose jobname is TEMP5. It can be used on systems running z/OS V1R2 or higher in z/Architecture mode. The REMOTE LIST parameter is ignored on z/OS V1R1 and OS/390 systems:

```
SLIP SET,IF,PVTMOD=(MYMOD,13E),
  REMOTE=(ACTION=SVCD,LIST=('TEMP5'.1_00000000,2_00000000,100000,+FFF)),
  ACTION=SVCD,END
```

Example 17

This is an example of using 64-bit operands in the TRDATA keyword. Eight bytes of data are fetched from address 1_00000000; the resulting address is used to fetch another 8 bytes, which is used as the starting address to capture X'10' bytes of storage in a trace buffer:

```
SLIP SET,IF,PVTMOD=(MYMOD,13E),ACTION=TRACE,TRDATA=(1_00000000!,+10),ML=1,END
```

Example 18

This example shows how to compare data in bits 32–64 of register 6 (the default), but ensures that bits 0–31 of register 6 are zero:

```
SLIP SET,IF,PVTMOD=(MYMOD,1A6),DATA=(1_00100000,EQC,6G),ACTION=WAIT,END
```

Example 19

This example is similar to the previous example, except that the order of the values specified on EQC is reversed. This SLIP trap compares data in all 8 bytes of register 6 (the default):

```
SLIP SET,IF,PVTMOD=(MYMOD,1A6),DATA=(6G,EQC,1_00100000),ACTION=WAIT,END
```

Example 20

This example compares four bits of data starting at bit 10 in register 6 with the data at address 1_00101000 starting at bit 10:

```
SLIP SET,IF,PVTMOD=(MYMOD,1B2),DATA=(6G(10),EQC(4),1_00101000),ACTION=WAIT,END
```

Example 21

This example compares two bits of data starting at bit 2 at the specified address with a constant:

```
SLIP SET,IF,PVTMOD=(MYMOD,1B2),DATA=(1_00101001(2),EQ,10),ACTION=WAIT,END
```

Example 22

This example stores eight bytes from 1_00000000 into the buffer addressed by register 7:

```
SLIP SET,IF,PVTMOD=(MYMOD,1B2),REFBEFOR=(7G!,EQC(8),1_00000000),ACTION=(REFBEFOR),END
```

Example 23

This example moves eight bits starting from bit position 4 at 1_00101000 to bit position 4 of register 6:

```
SLIP SET,IF,PVTMOD=(MYMOD,1B2),REFBEFOR=(6G(4),EQC(8),1_00101000),ACTION=(REFBEFOR),END
```

Modifying an Existing SLIP Trap

Use the SLIP MOD command to modify an existing SLIP trap.

Syntax

```
SLIP MOD{,ENABLE | ,DISABLE} {,ALL | ,ID=trapid}
```

Parameters

MOD

Specifies that the SLIP command modifies one or all existing SLIP traps. This parameter is positional; it must appear following a blank after SLIP.

ENABLE

The specified SLIP trap is to be made active.

Abbreviation: EN

DISABLE

The specified SLIP trap is to be made inactive.

Abbreviation: D

ALL

Every SLIP trap present in the system is to be modified. To find out what SLIP traps are in the system, issue DISPLAY SLIP.

ID=trapid

Only the SLIP trap with the identifier *trapid* is to be modified.

Where asterisks (*) replace any or all of the 4 characters of the identifier, all SLIP traps whose IDs match the non-asterisk characters are to be modified. If you specify fewer than 4 characters, the ID is padded on the right with blanks. A matching identifier must have blanks in those positions.

The asterisks allow you to group your SLIP traps by common characters in their IDs and enable and disable them as a group.

CAUTION:

The ALL parameter is extremely powerful because it modifies both the traps of every SLIP user and the traps the system uses to suppress unnecessary dumps. Issue SLIP MOD,ENABLE,ALL or SLIP MOD,DISABLE,ALL, therefore, only if you understand all its consequences for your system.

Example 1

The following SLIP command is used to deactivate the SLIP trap associated with identifier 0024. This SLIP trap can be activated again with the SLIP MOD,ENABLE,ID=0024 command.

```
SLIP MOD,DISABLE,ID=0024
```

Example 2

SLIP Command

Disable all SLIP traps with an identifier having 'A' as the first character and 'B' as the third character.

SLIP MOD,DISABLE, ID=A*B*

Deleting an Existing SLIP Trap

Use the SLIP DEL command to delete a SLIP trap.

Syntax

```
SLIP DEL{,ALL | ,ID=trapid}
```

Parameters

DEL

Specifies that the SLIP command remove one or all SLIP traps from the system. This positional parameter must appear following a blank after SLIP.

ALL

Every SLIP trap in the system is to be deleted. To find out what SLIP traps are in the system, issue DISPLAY SLIP.

ID=trapid

Only the SLIP trap with the identifier *trapid* is to be deleted.

Note: To delete more than one, but not all SLIP traps, you must enter a separate SLIP DEL command for each ID.

Attention: The ALL parameter is extremely powerful because it deletes both the traps of **every** SLIP user and the traps the system uses to suppress unnecessary dumps. Issue SLIP DEL,ALL, therefore, only if you understand all its consequences for your system.

Example

Use the following SLIP command to delete the SLIP trap with identifier 0008. This SLIP trap cannot be reactivated by a SLIP MOD command.

SLIP DEL, ID=0008

START Command

Use the START command to start started tasks, which support system functions such as IMS, CICS, and RACF. Started tasks are defined in cataloged procedures (residing in procedure libraries) or through jobs residing in a partitioned data set defined in master JCL.

Note: In general, IBM recommends that when you start a subsystem, you make the subsystem name specified in the IEFSSNxx parmlib member the same as that of the member you use from SYS1.PROCLIB. If the names do not match, you may receive error messages when you start the subsystem.

The following figure lists the tasks the START command can perform. Use it as an index to details about particular uses of the command.

Table 4-40. Summary of the START Command

Topic:	Command:
“Starting a System Task from a Console”	START membername
“Starting the APPC/MVS Address Space” on page 4-579	START APPC
“Starting the APPC/MVS Transaction Scheduler Address Space” on page 4-580	START ASCH
“Starting the Generalized Trace Facility” on page 4-581	START GTF
“Starting the Library Lookaside (LLA) Address Space” on page 4-583	START LLA
“Starting the Object Access Method (OAM)” on page 4-583	START OAM
“Starting Resource Recovery Services (RRS)” on page 4-584	START RRS
“Starting the System Object Model (SOM) Subsystem” on page 4-585	START SOM
“Starting TSO/VTAM Time-Sharing” on page 4-586	START membername
“Starting the Virtual Lookaside Facility or Data Lookaside Facility” on page 4-587	START VLF or START DLF
“Starting an External Writer” on page 4-588	START XWTR

Syntax

The syntax for each variation of the START command is shown immediately preceding its respective parameter list.

START or S
Note: For any variation of the START command, if you omit <i>devicetype</i> (or <i>devnum</i>), or <i>classes</i> , or <i>volumeserial</i> , you must supply a comma for each one of these parameters that you leave out. Do not supply any commas, however, after the last parameter you specify.

Starting a System Task from a Console

Use the START command to start a system task from a console.

Naming Considerations for Started Tasks: The job name or identifier of the started task is important for the following reasons:

- The name or identifier is used on DISPLAY, MODIFY, RESET, CANCEL, FORCE, and STOP commands for the started task

START Command

- The jobname or identifier is part of the RACF resource name passed to the SAF interface
- The job name and member name are passed to a security product (such as RACF to give the task a user identification)
- The job name is used in SMF records to identify the task.

The job name for a given started task can be assigned based on a variety of inputs. These inputs are examined in the following order, so that if item #1 is not specified, item #2 is used. If neither #1 nor #2 is specified, then #3 is used, and so on.

1. The jobname specified in the JOBNAME= parameter of the START command
or
The identifier specified on the START command.
2. The jobname specified on the JOB JCL statement within the member.
3. The device number specified on the START command, or the device number associated with the device type specified on the START command
or
The device number associated with the device type specified on the START command.
4. The device number associated with the IEFRDER DD statement within the member.
5. The member name.

IBM recommends that you use the JOBNAME parameter rather than an identifier. If you use the JOBNAME parameter, SMF records, messages, and automated programs can reflect or react to job status; identifiers can only be viewed at a console.

Note: JOBNAME and *identifier* are mutually exclusive; you cannot specify both parameters on the START command.

```
S membername[.identifier][,devicetype|, [/]devnum[,volumeserial]
 [,parameters][,JOBNAME=jobname][,JOBACCT=acct_info]
 [,SUB=subsystemname][,keyword=option[,keyword=option]...]
```

membername

The 1 to 8 character name of a member of a partitioned data set that contains the source JCL for the task to be started. The member may be either a job or a catalogued procedure. The subsystem that selects the job determines which JCL procedure library is called, usually MSTR, JES2 or JES3.

identifier

The name identifying the task to be started. This name can be up to 8 characters long. The first character must be alphabetical. If you omit an identifier and the started task does not have an IEFRDER DD statement, the system uses the job name as the identifier.

If you omit an identifier and the started task has an IEFRDER DD statement, the device allocated to that started task becomes the identifier name.

Note: The system allows multiple started tasks with the same *membername* and *identifier* to execute concurrently.

JOBNAME=jobname

The name that will be assigned to the job.

- If the source JCL is a procedure and you omit the JOBNAME keyword, the member name will be assigned as the job name.
- If the source JCL is a job and you omit the JOBNAME keyword, the system will use the job name assigned on the JOB statement in the JCL.

If you specify the JOBNAME keyword, the member name will be used only to identify the member that contains the JCL for the started task.

If you specify the JOBNAME keyword, you cannot specify *identifier*.

JOBACCT=acct_info

An account number, and any other accounting information that your installation requires, such as your department and room number.

The JOBACCT parameter specifies accounting data in the JCL JOB statement for the started task. If the source JCL was a job and already specified accounting data, the value specified on the JOBACCT parameter overrides the accounting data in the source JCL.

For a detailed description of the accounting information format and syntax rules, see *z/OS MVS JCL Reference*.

The length of *acct_info* cannot exceed 44 characters.

devicetype

The device type of the output device (if any) associated with the task.

Note: *devicetype* and *devnum* are mutually exclusive. You cannot specify both parameters on a START command.

[/]devnum

The device number of the device to be started. A device number is 3 or 4 hexadecimal digits. A slash (/) must precede a 4-digit number and is optional before a 3-digit number.

Note: *devnum* and *devicetype* are mutually exclusive. You cannot specify both parameters on a START command.

volumeserial

If *devicetype* is a tape or direct access device, the volume serial number of the volume mounted on the device.

parameters

Program parameters passed to the started program.

SUB=subsystemname

The name of the subsystem that selects the task for processing. The name must be one to four characters, defined in the IEFSSNxx parmlib member, and the subsystem must be active. If SUB= is not specified, the primary subsystem, as specified in IEFSSNxx, will select the task.

There is an exception to the above: If the name of the task being started is the same as that of a subsystem, the task will be started under the Master subsystem (MSTR). Then, because the only procedure libraries available to the Master subsystem are those specified in the MSTJCLxx's IEFPDSI data set, any procedures being started that are defined in the Job Entry subsystem's PROC00 data set but not in the MSTJCLxx data set, will be unavailable and will therefore not be found; the system will issue message IEFC612I.

START Command

When the task being started is a subsystem and you omit SUB=, it will be started under the master subsystem (MSTR) unless the subsystem itself asks to start under the Job Entry subsystem.

When you specify SUB=, you override the subsystem's request regarding under which subsystem the task should be started.

keyword=option

Any appropriate keyword parameter you specify to override the corresponding parameter in the cataloged procedure. The maximum length of each keyword=option is 66 characters. No individual value within this field may be longer than 44 characters in length. You can specify:

- JCL JOB statement keyword parameters, to modify the JOB statement for the started task
- JCL EXEC statement keyword parameters, to override EXEC statements in the procedure being started (if the target of the START command is a procedure)
- JCL DD statement keyword parameters, to modify the IEFRDER DD statement (if it exists)
- JCL symbol names, to provide values for use in the started task

Parameters on JOB Statements

JOB statement keyword parameters are accepted for both started jobs and started procedures. For started jobs, the keyword specified on the START command overrides or nullifies the corresponding keyword in the source JCL.

You can specify the following job-level keywords:

- ADDRSPC
- BYTES
- CARDS
- COND
- JESLOG
- LINES
- MSGCLASS
- MSGLEVEL
- NOTIFY
- PAGES
- PERFORM
- PRTY
- REGION
- TIME

You *cannot* specify the following job-level keywords, unless you code them with no values (to nullify the same keywords in the source JCL):

- USER
- GROUP
- PASSWORD
- RESTART
- SECLABEL
- TYPRUN

The system ignores the following keywords when they are specified on the START command, unless you code them with no values (to nullify the same keywords in the source JCL):

- RD
- CLASS

Note: For a started task:

- In a JES2 environment the system ignores the CLASS keyword.
- In a JES3 environment the system ignores all CLASS related attributes and functions except for device fencing, SPOOL partitioning, and track group allocation. For more information about CLASS attributes and functions, see *z/OS JES3 Initialization and Tuning Guide*.

Parameters on EXEC Statements

EXEC statement keywords that have the same name as JOB statement keywords (for example, COND, REGION, and TIME), are treated as JOB statement keywords.

Notes:

1. If you specify an option that must have lower case characters, enclose it within apostrophes.
2. If you specify an option within apostrophes, for example, 'DUMMY', use the correct case letters within the apostrophes. START command processing does not convert lower case characters to uppercase. Thus, entering lower case letters might cause a JCL error or an abend. Similarly, entering upper case letters where lower case is required (UNIX Services, for example) may cause incorrect results.
3. If you are overriding a data set name in the cataloged procedure and the name of the data set is 44 characters long, use DSN=name. If you specify DSNAME=name, the START procedure stops and returns message IEF640I.
4. If you need information on started task considerations or use of overrides and symbols, see *z/OS MVS JCL Reference*.

Example 1

To start JES2, enter:

```
S jes2
```

Example 2

To start JES2 with a job name of JES2 from a procedure named JESNOW, with an account number and other accounting information, enter:

```
S JESNOW,JOBNAME=JES2,JOBACT=(D548-8686,'12/8/85',PGMBIN)
```

Starting the APPC/MVS Address Space

Use the START APPC command to start the Advanced Program-to-Program Communication/MVS (APPC/MVS) address space.

```
S APPC,SUB=MSTR[,APP=(nn[,nn]...[,L])]
```

The parameters are:

APPC,SUB=MSTR

Invokes the APPC/MVS procedure and creates the APPC/MVS address space.

START Command

APPC=

Indicates which APPCPMxx parmlib members APPC/MVS is to use. The default member is APPCPM00. If you specify only one parmlib member, you do not need to enter the parentheses.

The APPCPMxx members can reside in SYS1.PARMLIB or any other parmlib data set that is specified on an //IEFPARM DD statement in the master scheduler JCL. However, APPC/MVS cannot process data sets specified in the master scheduler JCL until the operator enters a SET APPC command. For information about how to specify configuration information in a parmlib data set other than SYS1.PARMLIB, see the description of starting APPC and ASCH in *z/OS MVS Planning: APPC/MVS Management*.

- L The system will display parmlib statements on the operator's console as it processes them.

Example 1

START the APPC/MVS address space with parmlib member APPCPM00.

S APPC,SUB=MSTR

Example 2

START the APPC/MVS address space with parmlib member APPCPM01.

S APPC,SUB=MSTR,APPC=01

Example 3

START the APPC/MVS address space with parmlib members APPCPM03 and APPCPM06, and list the parmlib statements as the system processes them.

S APPC,SUB=MSTR,APPC=(03,06,L)

Starting the APPC/MVS Transaction Scheduler Address Space

Use the START ASCH command to start the ASCH (APPC/MVS transaction scheduler) address space.

```
S ASCH,SUB=MSTR[,ASCH=(nn[,nn]...[,L])]
```

The parameters are:

ASCH,SUB=MSTR

Invokes the ASCH procedure and creates the ASCH address space.

ASCH=

Indicates which ASCHPMxx parmlib members APPC/MVS is to use. The default member is ASCHPM00. If you specify only one parmlib member, you do not need to enter the parentheses.

The ASCHPMxx members can reside in SYS1.PARMLIB or any other parmlib data set that is specified on an //IEFPARM DD statement in the master scheduler JCL. However, APPC/MVS cannot process data sets specified in the master scheduler JCL until the operator enters a SET ASCH command. For information about how to specify configuration information in a parmlib data set other than SYS1.PARMLIB, see the description of starting APPC and ASCH in *z/OS MVS Planning: APPC/MVS Management*.

- ↳ The system will display the parmlib statements on the operator's console as it processes them.

Starting the Generalized Trace Facility

Use the START GTF command to start the generalized trace facility.

```
S {GTF|membername} [.identifier][,devicetype|[/]devnum[,volumeserial]
[([,MODE={INT|DEFER|EXT}] [,TIME=YES] [,DEBUG=YES]
[,BLOK={numpages|nnnK|nnM}] [,MEMBER=xxxxxxxx] [,REGION=nnnnK]
[,{SADMP|SA}={nnM|nnnk}] [,,{SDUMP|SD}={nnM|nnnk}]
[,{NOPROMPT|NP}] [,,{ABDUMP|AB}={nnM|nnnk}])
[,keyword=option[,keyword=option]...]
```

The parameters are:

GTF

The name of the IBM-supplied cataloged procedure that invokes GTF.

membername

The name of the member that contains the source JCL that invokes GTF. The source JCL can be either a cataloged procedure (for example, residing in SYS1.PROCLIB) or a job residing in a partitioned data set. The name can be either the IBM-assigned name (GTF) or a user-assigned name.

identifier

The user-determined name identifying this specific GTF session. If you omit an identifier, the system default is the device number of the device where the trace data set resides. the job name as the identifier.

devicetype

The device type, such as 3211, of the writer to be started. The device type provided in the IBM-supplied cataloged procedure for GTF is used unless overridden by this command.

Note: *devicetype* and *devnum* are mutually exclusive.

[/]devnum

The device number of the writer to be started. A device number is 3 or 4 hexadecimal digits. A slash (/) must precede a 4-digit number and is optional before a 3-digit number.

Note: *devicetype* and *devnum* are mutually exclusive.

volumeserial

The serial number of a magnetic tape or direct access volume to receive the trace data.

MODE=INT

Trace data is to be maintained in the GTF address space.

MODE=DEFER

Trace data is to be maintained in the GTF address space until the operator enters STOP GTF. Then, during end processing GTF will transfer the data from its address space to the output data set.

MODE=EXT

Trace data is to be maintained in an external data set.

START Command

BLOK=

Reserves common storage buffers to collect GTF data.

nnnK or nnM

The decimal number for the amount of storage in kilobytes(K) or megabytes(M). The minimum amount, also the default amount, is 40K.

numpages

The decimal number for the amount of 4096-byte pages of storage.

TIME=YES

Each logical trace record is to be time-stamped. If this parameter is omitted, individual records will be time-stamped. TIME=NO is not supported.

DEBUG=YES

GTF stops whenever an error is encountered while a trace record is being created. If this parameter is not specified, GTF attempts to recover from the error condition but might not be able to record future events of the same type.

MEMBER=xxxxxxxx

The parmlib member to be accessed by this invocation of GTF. If this parameter is omitted, GTFPARM is used.

REGION=*nnnnK*

The maximum size of the GTF address space in bytes. You can specify any number from 832K to 2880K.

keyword=option

Any appropriate keyword specified to override the corresponding parameter in the cataloged procedure. The maximum length of each keyword=option is 66 characters. No individual value within this field may be longer than 44 characters in length. If you are overriding a symbolic parameter, do not use any of the DD keywords. For example, do not use UNIT= to override the devicetype positional parameter in the cataloged procedure.

SADMP= or SA={*nnnK or nnM*}

The number of bytes of GTF data that will appear in a stand alone dump. The default is 40K.

SDUMP= or SD={*nnnK or nnM*}

The number of bytes of GTF data that will appear in an SVC dump (SDUMP). The default is 40K.

NOPROMPT or NP

If specified, indicates that the operator will not be prompted to specify trace options. Message AHL125A and AHL100A will not be issued. Use this parameter when you have a parmlib member set up with the desired GTF options and you want to avoid multiple replies in a sysplex environment.

ABDUMP= or AB={*nnnK or nnM*}

The number of bytes of GTF data that will appear in a SNAP or ABEND dump. The default is 0 K, which means that no GTF output data will appear in a SNAP or ABEND dump.

Example 1

This example starts a GTF session with the identifier EXAMPLE and with trace data maintained in the GTF address space. The DSN keyword is entered to override a symbolic parameter defined in the cataloged procedure.

```
S GTF.EXAMPLE,,, (MODE=INT),DSN=NULLFILE
```

Example 2

This example starts a GTF session with the trace data recorded on an external device. Because it is not apparent which is the GTF recording device, you have to display active jobs with the D A,LIST command before you can stop GTF. The GTF session started in this example could run in an address space of a maximum of one megabyte.

```
S GTF,,, (MODE=EXT),REGION=1000k
```

Starting the Library Lookaside (LLA) Address Space

Use the START LLA command to start the LLA address space.

```
S LLA[,SUB=MSTR] [,LLA=xx]
```

The parameters are:

LLA

Invokes the LLA procedure and creates the LLA address space.

The START LLA command is initially issued by the system from the logical parmlib member IEACMD00. Use this command if LLA stops, either because of an error or as a result of a STOP LLA command.

LLA=xx

Indicates which CSVLLAnn parmlib member LLA is to use. If you omit LLA=xx, LLA will build its directory using only the LNKLST libraries.

SUB=MSTR

Indicates that the master subsystem will process the task. If you omit this parameter, the system issues message CSV209I indicating that the request to start LLA is ended and the system re-issues the command with SUB=MSTR.

Notes:

1. System performance is degraded if LLA stops.
2. LLA provides better performance when VLF services are available, so it is better (although not necessary) to start VLF before starting LLA. However, the operation of LLA does not depend upon VLF.
3. Place the S LLA and S VLF commands in either the IEACMD00 or COMMANDxx members of the logical parmlib data set.

Starting the Object Access Method (OAM)

Use the START OAM to start the object access method.

```
S {OAM[membername] [.identifier] [,OAM=xx]}
```

For a detailed discussion of the START OAM command, refer to the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*, and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

START Command

Starting Resource Recovery Services (RRS)

Use the START RRS command to start resource recovery services (RRS). To start RRS during system initialization, add a START RRS command to the COMMNDxx parmlib member.

Before you can start RRS, your installation must have defined RRS as a subsystem in the IEFSSNxx parmlib member. For RRS to process requests for resources, system logger must be active.

The name of the cataloged procedure that IBM supplies in SYS1.SAMPLIB for starting the RRS subsystem is ATRRRS. Your installation should copy SYS1.SAMPLIB(ATRRLS) to SYS1.PROCLIB(RRS). If your installation replaces membername RRS with its own procedure for starting RRS, it should ensure that the name of its procedure matches the name of the subsystem specified in the IEFSSNxx parmlib member it uses. Otherwise, you may receive error messages when you start the subsystem.

You can **cold start** RRS/MVS when the RRS resource manager data log is empty. Cold start processing clears all active log streams except the RRS archive log, which is never cleared.

When the RRS data log contains resource manager data you can **warm start** RRS/MVS. Warm start processing restores the status of all incomplete backout or commit requests.

Only one copy of RRS can be running on a system. The system will reject an attempt to start a second RRS, even if you specify a different procedure as the first parameter of the START command.

```
S RRS|membername[,CTMEM=CTnRRSxx][,GNAME=lgrpname][,JOBNAME=jobname]
```

The parameters are:

RRS|membername

Invokes the RRS procedure and creates the RRS address space. If your installation has created a different procedure for starting RRS, use the member name of your procedure.

CTMEM=CTnRRSxx

Identifies the CTnRRSxx parmlib member that contains the options RRS component trace is to use when RRS starts the trace. If you omit this optional parameter, RRS traces only unexpected events until a time when the TRACE CT command specifies different trace options.

GNAME=lgrpname

Specifies the log group name. A log group is a group of systems that share an RRS workload. Specify a value if your installation has multiple RRS workloads. Otherwise, the name defaults to the sysplex name. If you specify a name, it must be 1-8 characters long. The first character must be alphabetic or one of the national characters (\$, #, or @), while the remaining characters may be alphanumeric or \$, #, or @.

You can find additional information about RRS logging in *z/OS MVS Programming: Resource Recovery*.

JOBNAME=jobname

Specifies the subsystem name defined in the IEFSSNxx parmlib member corresponding to RRS. The issuer of FORCE jobname,ARM must use this name to bring RRS down, if that proves necessary. If you omit this parameter, the system uses the started JCL procedure name.

Example

The following example starts RRS with the CTWRRS01 parmlib member:

```
START RRS,JOBNAME=RRS,CTMEM=CTWRRS01
```

Starting the System Object Model (SOM) Subsystem

Use the START SOM command to start the distributed system object model (SOM) subsystem. This is the extension to the OS/390 SOMobjects product that allows programs to communicate with objects in other processes that reside in other address spaces or on other systems. Your installation may place this command in a COMMNDxx member of the logical parmlib data set.

Note: You must activate the OMVS address space before you issue a START SOM command. See "Configuring Your SOMobjects Environment" in *OS/390 SOMobjects Configuration and Administration Guide*, GC28-1851, for more information on what things you must do before starting SOM.

```
S {SOM | procedure-name} [,SUB={MSTR,JES2,JES3}]
```

The parameters are:

SOM

The name for the SOM subsystem. It corresponds to the cataloged procedure in SYS1.PROCLIB that starts SOM.

procedure_name

Your installation's name for the SOM subsystem. The name must correspond to a procedure catalogued in SYS1.PROCLIB or to a data set concatenated to SYS1.PROCLIB. IBM recommends that an installation use the name **SOM**. If your installation uses a different name, that name must be one to four characters long, begin with an alphabetic character (A-Z), contain only alphanumeric characters (A-Z or 0-9), and not conflict with any other subsystem name in the same system, or with the name of any system command.

SUB=subsystem name

Specifies the subsystem you designate to process the task, such as MSTR (the master subsystem) or JES2 or JES3 to specify that subsystem's scheduler. (Make sure to send all your output to a data set if you specify MSTR.)

If you omit this parameter, the JES subsystem scheduler starts SOM. The resulting dependency on JES requires SOM to be stopped when stopping JES.

Example

The following command starts the SOM subsystem:

```
S SOM
```

START Command

Starting TSO/VTAM Time-Sharing

Use the START command to start TSO once VTAM is active. This command creates the terminal control address space (TCAS), which accepts requests by terminal users for logon to TSO.

```
S membername[.identifier][,devicetype|, [/]devnum  
[,volumeserial][([, MEMBER={name|nn}][,USERMAX=nnnnn]  
[,GNAME=[name | NONE]])  
[,keyword=option[,keyword=option]...]]
```

The parameters are:

membername

The name of the member that contains the source JCL for the started task. The member can be a cataloged procedure (for example, residing in SYS1.PROCLIB) or a job that starts TSO/VTAM time-sharing. Many installations use TCAS as the membername.

identifier

The user-determined name identifying this specific time-sharing session. If you omit an identifier, the system automatically uses the job name as the identifier.

devicetype

The device type of an output device where time-sharing parameters from SYS1.PARMLIB are listed. This operand is useful only if the PRINTOUT DD statement of the cataloged procedure invoked by the START command specifies DDNAME=IEFRDER. If the PRINTOUT DD statement specifies a SYSOUT device, the parameter values are listed on that device.

Note: *devicetype* and *devnum* are mutually exclusive.

[/]devnum

The device number of an output device where time-sharing parameters from SYS1.PARMLIB are listed. A device number is 3 or 4 hexadecimal digits. A slash (/) must precede a 4-digit number and is optional before a 3-digit number.

Note: *devicetype* and *devnum* are mutually exclusive.

volumeserial

If *devicetype* is a tape or direct access device, the volume serial number of the volume mounted on the device.

MEMBER=name

The name of the member in Parmlib that contains TSO/VTAM time-sharing parameters.

MEMBER=nn

A two-digit decimal number that forms the suffix for specifying the name of the member in Parmlib that contains TSO/VTAM time-sharing parameters, where the name has the form TSOKEYnn.

USERMAX=nnnnn

The maximum number (0 to 32,767) of users that can be logged on to TSO/VTAM time-sharing at any one time. If USERMAX is not specified here or in the parmlib member that contains TSO/VTAM time-sharing parameters, a value of 40 is used.

GNAME=name

The generic resource name for TSO/VTAM to use while operating in an OS/390 sysplex environment. You may specify this keyword only when TSO/VTAM is operating within a sysplex.

GNAME=NONE

Specifies that TSO/VTAM will not use a generic resource name. If you specify this value on the START command, the system will ignore any GNAME value in the TSOKEY00 parmlib member.

keyword=option

Any appropriate keyword specified to override the corresponding parameter in the cataloged procedure. The maximum length of each keyword=option is 66 characters. No individual value within this field may be longer than 44 characters in length. If you are overriding a symbolic parameter, do not use any of the DD keywords. For example, do not use UNIT= to override the device type positional parameter in the cataloged procedure.

If more than one Parmlib name is specified, or if no name is specified, the order of priorities that determines which time-sharing parameters are used is:

1. The member name coded on the PARMLIB DD statement.
2. The MEMBER operand of the START command.
3. The keyword operand of the START command.
4. The default member TSOKEY00 if a member is not specified but a Parmlib (that contains TSOKEY00) is.
5. The default values in the TCAS program if neither a member nor a parmlib is specified.

Starting the Virtual Lookaside Facility or Data Lookaside Facility

Use the START VLF command to start the virtual lookaside facility (VLF) and the START DLF command to start data lookaside facility (DLF). VLF is an MVS service that enables applications to minimize I/O operations for frequently retrieved objects. DLF is an MVS service that provides the capability for multiple jobs to share access to large data objects in storage. It enables QSAM and VSAM applications to minimize I/O operations.

VLF and DLF will not start if already active on the system; the START command will be rejected. Also, they will not start if the parmlib is not allocated to the appropriate started task; if the specified parmlib member is not found, or is empty; or if there is no valid class found in the specified parmlib member. If you need to change the specified parmlib member, you must stop the service, VLF or DLF, then start it again, specifying the new parmlib member.

```
S {VLF|DLF},SUB=MSTR[,NN=xx]
```

The parameters are:

VLF,SUB=MSTR

Invokes the VLF procedure that starts the virtual lookaside facility (VLF).

DLF,SUB=MSTR

Invokes the DLF procedure that starts DLF.

START Command

NN=xx

Indicates that the system is to start VLF using the COFVLFxx member of the logical parmlib or that the system is to start DLF using the COFDLFxx member of the logical parmlib. In each case, replace xx with two alphanumeric characters that match the suffix of the parmlib member.

If you do not identify a parmlib member, VLF uses the default parmlib member COFVLF00 and DLF uses the default parmlib member COFDLF00. See *z/OS MVS Initialization and Tuning Reference* for further information about the use of the COFVLFxx or COFDLFxx member.

Starting an External Writer

Use the START XWTR command to start an external writer.

```
S {XWTR|membername} [.identifier] [,devicetype|, [/]devnum  
[,volumeserial][,classes][,keyword=option[,keyword=option]...]
```

The parameters are:

XWTR

The name of the IBM-supplied cataloged procedure that invokes the external writer.

membername

The name of the member that contains the source JCL that starts and defines the external writer. The member can contain a cataloged procedure (for example, residing in SYS1.PROCLIB) or a job residing in a partitioned data set. The name can be either the IBM-assigned name (XWTR) or a user-assigned name.

identifier

The identifier of the writer to be started. This name consists of up to eight characters. The first one must be alphabetical. If you do not assign an identifier, the system uses the device number of the device allocated to the writer as the identifier.

devicetype

The device type, such as 3211, of the writer to be started.

Note: devicetype and devnum are mutually exclusive.

[/]devnum

The device number of the writer to be started. A device number is 3 or 4 hexadecimal digits. A slash (/) must precede a 4-digit number and is optional before a 3-digit number.

Note: devicetype and devnum are mutually exclusive.

volumeserial

The serial number, up to six characters, of the magnetic tape or direct access volume the writer is to use.

classes

The output classes, in priority sequence, the writer is to process. You can specify up to eight output classes, naming them in sequence without separating them by commas.

keyword=option

Any appropriate keyword specified to override the corresponding keyword in the cataloged procedure. The maximum length of each keyword=option is 66 characters. No individual value within this field may be longer than 44 characters in length. If you are overriding a symbolic parameter, do not use any of the DD keywords. For example, do not use UNIT= to override the devicetype positional parameter in the cataloged procedure.

Example

To start an external writer with the identifier A, enter:

```
S XWTR.A,282
```

STOP Command

STOP Command

Use the STOP command to stop system functions and jobs in execution. Note that you can communicate with the currently running program only if it was designed to recognize the STOP command. If the program does not recognize the STOP command, MVS issues message IEE342I STOP REJECTED--TASK BUSY.

Note to Programmers: For more information, see the section on communicating with a program using EXTRACT and QEDIT in *z/OS MVS Programming: Authorized Assembler Services Guide*.

The following figure lists tasks the STOP command can perform. Use it as an index to details about particular uses of the command.

Table 4-41. Summary of the STOP Command

Topic:	Command:
“Stopping a Running Program”	STOP jobname.identifier
“Stopping an ASCH Initiator” on page 4-593	STOP ASCHINT
“Stopping the Data Lookaside Facility (DLF)” on page 4-593	STOP DLF
“Stopping the Library Lookaside (LLA) Address Space” on page 4-593	STOP LLA
“Stopping the Object Access Method (OAM) Address Space” on page 4-594	STOP OAM
“Stopping a System Object Model (SOM)” on page 4-594	STOP SOM
“Stopping a Temporary File System (TFS)” on page 4-595	STOP TFS
“Stopping the Virtual Lookaside Facility (VLF)” on page 4-595	STOP VLF

Syntax

The syntax for each variation of the STOP command is shown immediately preceding its respective parameter list.

STOP or P

Stopping a Running Program

You can use the STOP command to stop a running program when that program was designed to listen for the STOP command. If the program does not recognize the input specified on the STOP command, MVS issues message IEE342I STOP REJECTED--TASK BUSY.

Note to Programmers: For more information, see the section on communicating with a program using EXTRACT and QEDIT in *z/OS MVS Programming: Authorized Assembler Services Guide*.

P [jobname.]identifier[,A=asid]

Parameters

The parameters are:

jobname

The name of the batch job, started task, or APPC/MVS transaction program to be stopped.

The job name for a given started task can be assigned based on a variety of inputs. These inputs are examined in the following order, so that if item #1 is not specified, item #2 is used. If neither #1 nor #2 is specified, then #3 is used, and so on.

1. The jobname specified in the JOBNAME= parameter of the START command
or
The identifier specified on the START command.
2. The jobname specified on the JOB JCL statement within the member.
3. The device number specified on the START command, or the device number associated with the device type specified on the START command
or
The device number associated with the device type specified on the START command.
4. The device number associated with the IEFRDER DD statement within the member.
5. The member name.

identifier

The identifier assigned to the job or address space. (Refer to “Displaying Started Task Status” on page 4-157 for information about determining the jobname and identifier of currently active address spaces.)

The following types of identifiers can be used:

- The identifier that was specified on the START command.
- */Jdevnum*, the device number specified on the START or MOUNT command. A device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). You can precede the device number with a slash to prevent ambiguity between the device number and a device type or identifier.
- *devicetype*, the type of device specified when the START or MOUNT command was issued.

If no identifier was specified, the identifier “STARTING” is temporarily assigned until the system can assign another according to the following order of precedence:

1. If an identifier was not specified on the START command, the identifier is the device type (for example, 3410) or device number (for example, X'0000') specified on the START or MOUNT command.
2. If an identifier, a device type, or a device number was not specified on the START or MOUNT command, the identifier is the device type specified on an IEFRDER DD statement (invoking a cataloged procedure) in the JCL.
3. If none of the above was specified, the identifier defaults to the job name.

When (and only when) you specify *jobname* with *identifier*, the identifier can be represented by any of the following:

- An asterisk
- One or more characters from the beginning of the identifier,
- The entire identifier

STOP Command

When you use the asterisk format, the command takes effect on *all* address spaces that begin with the specified characters. Device numbers are assumed to be 4-digit numbers; for example, **/13*** would match on **1301**, **1302**, and so on, but would not match on **13C**.

Specifying both the job name and the entire identifier causes the command to take effect on *all* the work units running with that combination of job name and identifier. If you are uncertain of an identifier or device number:

- Use the DISPLAY A,LIST command to display active jobs
- Use the DISPLAY ASCH,A command to display active TP jobnames.
- Use the DISPLAY A,A command to display active initiator address spaces.

A=asid

The address space identifier, in hexadecimal, of the job, started task, writer, transaction program, or initiator address space to be stopped.

If you issue a STOP command with a non-unique identifier, device name, or device type, multiple tasks might be stopped. To make sure that the STOP command stops only one task, add a unique job name to the START command membername; use that same unique job name on the STOP command.

If you enter the STOP command and one or more terminals are still active, you are asked to respond to the following message:

* id IKT010D nn USERS ACTIVE – REPLY 'SIC' or 'FSTOP'

Reply '*' 'SIC' to cancel the active users normally. This reply allows them to receive any messages queued for them, and it allows TSO/VTAM to perform its normal termination processing. Reply 'FSTOP' to force immediate cancellation of the active users. The users do not receive any messages queued for them, and TSO/VTAM does not perform its normal termination processing (that is, task resource manager processing is bypassed). Use FSTOP only if 'SIC' is ineffective.

Example 1

If the job SYSDA has been written to accept a STOP command, it stops. Otherwise, the following command has no effect.

P SYSDA

Example 2

To stop writer 00E after it processes the current data set, enter:

P XWTR.00E

or

P 00E

Example 3

To stop the GTF session started with an identifier of ABCD, enter:

P ABCD

Example 4

To stop writer ABCD after it processes the current data set, enter:

P XWTR./ABCD

or
P /ABCD

Example 5

To stop the GTF session with an identifier of EXAMPLE, enter:
P EXAMPLE

Stopping an ASCH Initiator

Use the following form of the STOP command to stop an ASCH initiator.

```
P {ASCHINT,A=asid}
```

ASCHINT,A=asid

ASCHINT is the generic name for the ASCH initiator and A=asid is the address space identifier, in hexadecimal, of the ASCH initiator to be stopped.

Example

To stop the ASCH initiator address space, whose asid is E, enter:
P ASCHINT,A=E

Stopping the Data Lookaside Facility (DLF)

Use the STOP DLF command to stop the data lookaside facility (DLF). Before entering STOP DLF, you must enter the MODIFY DLF,MODE=DRAIN or QUIESCE command.

```
P DLF
```

The parameter is:

DLF

The job name assigned to the data lookaside facility (DLF) address space. Using this parameter will stop DLF as soon as all DLF objects are disconnected for all users.

This command activates the address space termination process. You can use the MODIFY DLF,MODE=NORMAL command to end the DLF shutdown process. To restart DLF after the shutdown process has completed, you can use the START DLF command.

Stopping the Library Lookaside (LLA) Address Space

Use the STOP LLA command to stop the LLA address space.

```
P LLA
```

The parameter is:

STOP Command

LLA

The job name assigned to the LLA address space.

Stopping LLA might be done when two or more systems have shared access to the same LLA directories and modifications are to be made to these shared directories. After stopping LLA and modifying the shared LLA data sets as needed, the operator can use the START LLA command to restart LLA. System performance will be degraded until you restart LLA, but this procedure allows that the updates will be made simultaneously on all of the sharing systems.

Stopping the Object Access Method (OAM) Address Space

Use the STOP OAM command to stop the OAM address space.

For a detailed discussion of the STOP OAM command, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*, and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Stopping a System Object Model (SOM)

Use the STOP SOM command to stop the distributed SOM subsystem. This is the extension to the OS/390 SOMobjects product that allows programs to communicate with objects in other processes that reside in other address spaces or other systems. Executing this command causes the SOM subsystem to wait for all servers to end, and then to stop. See *OS/390 SOMobjects Configuration and Administration Guide*, GC28-1851, for more information on stopping the SOM subsystem.

```
P SOM
```

The parameter is:

SOM

The name for the SOM subsystem. It corresponds to the cataloged procedure in SYS1.PROCLIB that starts SOM. A sample cataloged procedure is located in MVSDSOM.DR03.SGOSJCL(PROCDSVR).

Note: In order for this command to be effective, the SOM daemon address space must be active.

Example

To stop the SOM daemon address space, enter:

```
P SOM
```

When you enter the STOP SOM command, the system displays these messages:

```
GOS041I SOM/MVS SOMI WAITING FOR SERVERS TO END.  
GOS010I SOM/MVS SOMI ENDED.
```

The system might also display the following message:

```
IEF352I ADDRESS SPACE UNAVAILABLE
```

This is normal and you should not report it as a problem. The consequence of purging SOM is that it makes the address space identifier (ASID) permanently unusable, for system integrity reasons.

Stopping a Temporary File System (TFS)

Use the STOP TFS command to stop a TFS running in a colony address space. If no file TFS systems are mounted, executing this command causes TFS to exit. A WTOR is issued, allowing TFS to be restarted. This command is not supported if TFS runs in the z/OS Unix kernel address space.

```
P TFS
```

The parameter is:

TFS

The name of the TFS colony address space to be stopped.

Stopping the Virtual Lookaside Facility (VLF)

Use the STOP VLF command to stop the virtual lookaside facility (VLF).

```
P VLF
```

The parameter is:

VLF

The jobname assigned to the virtual lookaside facility (VLF). Using this parameter stops VLF with message number COF033I.

Performance Implication: Stopping VLF can degrade system performance.

STOPMN Command

STOPMN Command

Use the STOPMN command to stop the continual display of job status, data set status, or time-sharing user session activity initiated in response to the MONITOR command or MONITOR parameters on the CONSOLE and INIT statements in the CONSOLxx parmlib member.

Syntax

The complete syntax for the STOPMN command is:

```
PM {JOBNAMES[,L={a|name|name-a}] }  
  {DSNAME }  
  {SPACE }  
  {STATUS[,L={a|name|name-a}] }  
  {SESS[,L={a|name|name-a}] }
```

Parameters

The parameters are:

JOBNAMES

Stop the jobname display specified in the MONITOR JOBNAMES command.

DSNAME

Stop the display of non-temporary data set names specified in the MONITOR DSNAME command.

SPACE

Stop the display of available space on direct access volumes specified in the MONITOR SPACE command.

STATUS

Stop the display of data set names, volume serial numbers, and status specified in the MONITOR STATUS command.

SESS

Stop the display of time-sharing user identifiers.

L=a, name, or name-a

The display area (*a*), name of the console (*name*), or both (*name-a*) where the display is to be stopped unless routing instructions are in effect. If you omit this keyword, the display is stopped at the console at which you enter the PM command.

Example: To discontinue the display of job name information that appears when a job is initiated or stopped, enter:

```
PM JOBNAMES
```

STOPTR Command

STOPTR COMMAND NO LONGER SUPPORTED

The STOPTR command is not supported on systems at z/OS V1R7 and higher.

Use the STOPTR command to halt or reduce the information periodically displayed as a result of the TRACK command.

Scope in a Sysplex

The STOPTR command has sysplex scope only when you specify L=. See “Using Commands That Have Sysplex Scope” on page 1-11 for more about sysplex scope.

Syntax

The complete syntax for the STOPTR command is:

```
PT {TS|JOBS|J|A} [,L={a|name|name-a}]
```

Parameters

The parameters are:

TS

Stop the display of active time-sharing users.

JOBS or J

Stop the display of active jobs, active Advanced Program-to-Program Communication/MVS (APPC/MVS) transaction programs, MOUNT commands in execution, and active tasks.

A Stop the display of active jobs, active APPC/MVS transaction programs, active time-sharing users, MOUNT commands in execution, and active tasks.

L=a, name, or name-a

The display area (a), console name (name), or both (name-a) of the active MCS or SMCS console where the display appears. STOPTR is not valid for extended MCS consoles or system consoles. If you omit this operand, the command affects the dynamic display on the console through which the command is entered.

Note: TRACK A followed by STOPTR TS or STOPTR JOBS has the same effect as TRACK JOBS or TRACK TS.

Example 1

To discontinue the periodic display of system status information directed to area A of the console named CON22, enter:

```
pt a,1=CON22-A
```

Example 2

STOPTR Command

To display the number of starts, mounts, jobs, time-sharing users, and a list of jobs, including APPC/MVS transaction programs (but not time-sharing users), enter:

```
TR A,L  
PT TS
```

SWAP Command

Use the SWAP command to initiate an operator request for dynamic device reconfiguration (DDR) and to activate or deactivate system-initiated DDR. DDR is described in “Responding to Failing Devices” on page 1-39.

Because a system-initiated DDR swap is automated in an IBM 3495 Tape Library Dataserver, the operation usually completes without operator intervention. Messages that do not require operator intervention are not sent to the console but are sent to the hardcopy log, where they are available for tracing and debugging. Note that the operator can still initiate swaps in an IBM 3495 Tape Library Dataserver. Table 4-42 summarizes the information that the SWAP command provides. Use it to access the pages on which you can find details about a particular use of the SWAP command.

Table 4-42. Summary of the SWAP Command

Command:	Topic:
SWAP devnum1,devnum2	“Operator-Requested DDR”
SWAP OFF SWAP ON	“System-Initiated DDR” on page 4-600

Syntax

The syntax for each variation of the SWAP command is shown immediately preceding its respective parameter list.

SWAP or G

Operator-Requested DDR

Use the following form of the SWAP command when it is necessary to move a volume to another device, for example, when the device requires maintenance.

G [/]devnum1, [/]devnum2

[/]devnum1

The device number of the device from which the volume is to be swapped.

[/]devnum2

The device number of the device to which the volume is to be swapped.

devnum1 and *devnum2* must be of the same device type and have compatible features. A device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/).

To perform a dynamic device reconfiguration, enter a SWAP command. Then:

1. Wait for system message IGF500D or IGF509D and reply with one of the following:

YES The system is to proceed as indicated.

NO The swap request is to be canceled.

devnum2 An alternate “to” device is to be used.

2. After replying YES or *devnum2*, wait for the following message before proceeding with the swap:

* id IGF502E PROCEED WITH SWAP OF *devnum1* TO *devnum2*

SWAP Command

Notes:

1. When the alternate “to” (*devnum2*) device is offline, the system takes the ‘from’ device offline:
IEF880I *devnum1* NOW OFFLINE BY DDR
2. The system never requests DDR for readers, printers, or punches, but you can request a swap of these devices. Unit record devices must be in a not-ready state when you enter the SWAP command. Devices with mounts pending cannot be swapped.
3. If you are using a 3348 Model 70F Data Module, make sure that the “to” 3340 device has the fixed-head feature installed.
4. A device with a 3-digit device number can be swapped to a device with a 4-digit device number, but not the reverse.

Example 1

To move a volume from a device with device number 183 to a device with device number 283, enter:

```
swap 183,283
```

Example 2

To move a volume from a device with device number 183 to a device with device number 3330, enter:

```
swap 183,/3330
```

System-Initiated DDR

If a permanent I/O error occurs on a DDR-supported device, and it is an error that DDR recognizes, the system requests that the volume be moved.

Use the following form of the SWAP command to activate or deactivate system-initiated DDR.

G {OFF <u>ON</u> }

The parameters are:

OFF

System-initiated DDR is to be deactivated. Any further swapping of devices must be done by operator-initiated DDR.

ON

System-initiated DDR is to be activated for any further swapping of devices.

Note: If a system-initiated DDR is in progress, the swap will be completed before system-initiated DDR can be deactivated. Entering G=OFF will not effect any swap already in progress.

Example

To deactivate system-initiated DDR, enter:

```
g off
```

SWITCH Command

SWITCH CN COMMAND NO LONGER SUPPORTED

The SWITCH CN command is not supported on systems at z/OS V1R8 and higher.

Use the SWITCH command to:

- Manually switch the recording of SMF (system management facilities) data from one data set to another
- Switch the following console attributes from one console to another console:
 - Routing codes
 - Message levels
 - MCS and SMCS console attributes
 - Message scope

You can use SWITCH CN to append console attributes so that one console has the attributes of both or to reverse the append. You can also:

- Switch the console attributes of an active or inactive console to another console.
- Switch the console attributes of an active or inactive console to its first available alternate console.
- Restore the console attributes of an inactive console that has switched to an active alternate.

Scope in a Sysplex

The SWITCH command has sysplex scope only when you specify the CN parameter. See “Using Commands That Have Sysplex Scope” on page 1-11 for an explanation of sysplex scope.

Syntax

The complete syntax for the SWITCH command is:

```
I {SMF }
```

Parameters

The parameters are:

SMF

The recording of SMF data is transferred from one SMF data set to another. All SMF data in storage is to be written out before the transfer is made.

CN=(*consname1*)

Indicates the console whose attributes the system is to switch. This console can be an MCS, SMCS or extended MCS console, either active or inactive. The console you specify switches only to an active and compatible console that is a member in its alternate group. Alternate console groups are defined in parmlib member, CNGRPxx. For more information see *z/OS MVS Initialization and Tuning Reference*.

After it is switched, the system deactivates the console.

SWITCH Command

If the console you specify cannot switch to any console, the system rejects the command and issues error message IEE686I. The system uses the following rules to determine if one console can switch to another, assuming both are active:

- A full capability console can switch only to another full capability console.
- A status display console can switch to another status display console or a full capability console.
- A message stream console can switch to another message stream console or a full capability console.
- A printer console device can switch only to another printer console device.
- Extended MCS consoles can serve as full capability consoles.
- Only MCS and SMCS consoles can serve as the master console. Extended MCS and subsystem consoles cannot serve as the master console.
- It is acceptable to list extended MCS consoles as alternates for MCS or SMCS consoles. However, since extended MCS consoles can never be the master consoles, extended MCS consoles will be ignored when attempting to switch the master console.
- You may not VARY an extended MCS console to be the master console using an operator command.
- You may not switch the master console (that is, the console currently having COND=M) to an extended MCS console.

Note: If you specify only one console name, you do not need to enter the parentheses.

(*consname1,consname2*)

Indicates that the system is to append the attributes of the console specified for *consname1* with the attributes of the console specified for *consname2*. *Consname1* can be an MCS, SMCS or extended MCS console, either active or already switched to another console. If *consname1* has already been switched to another console, the system removes *consname1*'s attributes from its alternate console before *consname1* switches to *consname2*. *consname2* must be active and compatible with *consname1*. If it is not active, the system rejects the SWITCH command with error message IEE274I.

(*consname1,consname1*)

Indicates the console whose attributes the system is to remove from its alternate console. This console must have been previously switched to another console. If you specify a console that has not been switched or one that is currently active, the system rejects the SWITCH command with error message IEE686I. This command will not work on an extended MCS console that has been switched but not deactivated.

In a sysplex:

Notes:

1. You can switch only one console to another console at a time.
2. If you specify an MCS or SMCS console, the system switches its attributes, then varies it offline.
3. If you specify an extended MCS console, the system:
 - a. Switches the console's attributes.
 - b. Alters the console's message delivery status so that no more messages are queued to this console.

TRACE Command

Use the TRACE command to:

- Start, stop, or modify system trace
- Start, stop, or modify master trace
- Start, stop, or modify component trace
- Start, stop, or modify transaction trace
- Display the current status of system trace, master trace, component trace, and transaction trace

During system initialization, or whenever you reactivate system trace after a system trace failure, the system creates a TRACE address space. That address space contains the system trace table. When the TRACE address space is created, the initial status of system trace (address space and explicit tracing functions) is **on**, the initial status of the branch tracing function of system trace is **off**, and the initial space set aside for system trace entries for each processor is **64K**.

You can issue TRACE ST, TRACE MT, TRACE CT, and TRACE TT from any console with master authority. You can issue TRACE STATUS from any console.

Syntax

The complete syntax for the TRACE command is:

```
TRACE [STATUS
      [ST[,nnnK][,BR={ON|OFF}]]]
      [ST[,,OFF]]
      [MT[,nnnK],OFF]
      [CT{,WTRSTART=membername[,WRAP|NOWRAP]}]
      [CT{,WTRSTOP=jobname[,FLUSH|NOFLUSH]}]
      [CT{[,ON],COMP=name[,SUB=(subname)][,PARM=mem]}]
      [,nnnnK]
      [,nnnnM]
      [,OFF]
      [TT[{},COLL|C=collection name}]
      [,CON=connection type]
      [,COR=correlation info]
      [,LU=logical unit name]
      [,LVL=level]
      [,NET=netid]
      [,PKG=package name]
      [,PLAN|PL=plan name]
      [,PRF=perform]
      [,PROC|PR=proc name]
      [,PRS=process]
      [,SUB=subsystem]
      [,TC=transaction class]
      [,TRAN|T=transaction name]
      [,USER|U=userid]
      [,WTR=membername|STOP]
      [,LATENT=Y|N]
      [,BUFSIZ=nnnK|nnM]
      [,OFF={nn|ALL}]]]
```

Parameters

STATUS

The system is to display the current status of master trace. Status information includes the current size of the master trace table.

TRACE Command

The display tells the operator to use the DISPLAY TRACE command to obtain status for system and component trace.

ST

The system is to change the on or off status of system trace, the size of the system trace table, or the on or off status of the branch tracing function of system trace. Unless you specify **ST,OFF**, the system assumes you want to recreate the TRACE address space if it has terminated and turn system trace on if it is not on already.

nnnK

The amount of preferred, central storage in K bytes set aside for system trace table entries for each processor. You can specify for *nnn* any decimal number from 1 to 999. If you specify a number that is not a multiple of 4, the system rounds it up to the next multiple of 4. If you omit this parameter, the system assumes 256K for each processor, or the size established by the last TRACE command during the IPL that specified a table size.

BR=ON or OFF

The system is to turn on or turn off the branch and mode tracing functions of system trace. The address space and explicit tracing functions remain on as long as system trace remains on. If you omit this parameter, the system assumes that the status of branch tracing remains unchanged.

BR=ON is intended for use in system software problem determination and diagnosis situations only. Branch tracing consists of tracing these four types of branches:

- Branch and stack (BAKR)
- Branch and link (BALR)
- Branch and save (BASR)
- Branch and save and set mode (BASSM)

By enabling branch tracing you may be impacting system performance.

Mode tracing consists of recording occurrences of entering and leaving 64-bit mode.

Note: You can not specify the **BR=** parameter if you specify **ST=OFF**.

OFF

The system is to stop system trace and free the system trace table. The system does not terminate the TRACE address space. Therefore, if you start system trace again while the TRACE address space is still active, the on or off status of the different system trace functions and the size of the system trace table return to the values they had the last time system trace was on.

MT

The system is to change the on or off status of master trace or the size of the master trace table. Unless you specify **MT,OFF**, the system assumes you want to turn master trace on if it is not on already.

nnnK

The master trace table size you want the system to use. You can specify for *nnn* any decimal number from 16 to 999. If a master trace table already exists, this new table replaces it. If you omit this parameter, the system assumes a table size of 24K.

OFF

The system is to stop master trace.

CT

Specifies the component tracing options for an MVS component or an application. The system programmer will supply the trace parameters. To determine if the component to be traced allows the following parameters, see "component traces" in *z/OS MVS Diagnosis: Tools and Service Aids*.

WTRSTART=membername

Identifies the name of the member that contains the source JCL that invokes a component trace external writer. The *membername* is 1 to 7 characters. The system also opens the data sets the writer uses. The member can be a SYS1.PROCLIB cataloged procedure or a job. Many installations use a cataloged procedure in SYS1.PROCLIB.

After you enter a TRACE CT,WTRSTART command, you should turn the trace on and connect the writer with a WTR parameter in the reply for a TRACE CT command or in a parmlib member, if specified.

WRAP or NOWRAP

If you specify WRAP, when the system reaches the end of the data set or group of data sets, it writes over the oldest data at the start of the data set or the start of the first data set. If you specify NOWRAP, the system stops writing to the data set or sets when the data set or sets are full.

If the WTRSTART parameter on the CTncccxx parmlib member or TRACE CT command specifies NOWRAP, the system uses the primary and secondary extents of the data set or sets. If the WTRSTART parameter specifies WRAP or omits the parameter, the system uses only the primary extent or extents.

WTRSTOP=jobname

Identifies the name of the job for a currently running component trace external writer that the system is to stop. The *jobname* is 1 to 7 characters. The system also closes the data sets the writer used.

The *jobname* is either:

- Member name, if the source JCL is a procedure
- Job name, if provided on a JOB statement within the source JCL

If you specify or default FLUSH, the writer will stop when it has finished writing out its current buffers. If you specify NOFLUSH, the writer stops immediately.

Before you enter a TRACE CT,WTRSTOP command, you must either:

- Turn the trace off, or
- Disconnect the component trace external writer from the trace leaving the trace on.

To disconnect the external writer while leaving the trace on enter the TRACE CT,ON command with WTR=DISCONNECT in the reply or in a CTncccxx parmlib member, if specified.

If the trace is not turned off or disconnected from the writer, message ITT121I informs the operator of the condition and the writer will not stop.

ON

If the component trace is currently off, a TRACE CT,ON command turns it

TRACE Command

on. If the component trace is currently on and can be changed, a TRACE CT,ON command changes the trace options. An installation-supplied application trace can also have head level and sublevel traces, if specified in the CTRACE DEFINE macro that defined the trace.

Whenever a trace that has sublevel traces is changed, all sublevel traces currently in the LIKEHEAD state will also be changed. Therefore, a change may cascade down a number of levels.

A head trace may have been defined so that it is not allowed to be changed (HEADOPTS=NO on the CTRACE DEFINE macro). If this is the case, the trace is really just a place holder for options for other traces.

nnnnK

nnnnM

Specifies the size, in kilobytes (K) or megabytes (M), of the trace buffer you want the system to use. Specifying the buffer size also turns the trace on.

nnnnK is the buffer size in kilobytes, where *nnnn* is a decimal number from 1 to 9999. *nnnnM* is the buffer size in megabytes, where *nnnn* is a decimal number from 1 to 2047.

When the size is not specified, the system uses the component-defined default or the size specified in a CTnccxx parmlib member.

The size specified for an installation-supplied application trace must be within the range specified on the CTRACE DEFINE macro for the trace; see the programmer for the size value.

OFF

The system is to turn off tracing for the component. If the component is connected to a component trace external writer, the trace will be implicitly disconnected from the writer.

Some components do not turn tracing completely off. Instead, they reduce the tracing activity to the minimum required for serviceability data in a dump. If the CTRACE DEFINE macro that defined the trace specified the MINOPS parameter, tracing is reduced to a minimum and component trace writes a message to the operator.

If a component level trace has sublevel traces that are defined with the LIKEHEAD parameter on the CTRACE DEFINE macro, the sublevel traces will either be reduced to the minimal tracing or turned off, in the same manner as their head level component trace.

COMP=*name*

Identifies the component trace affected by the command. *name* is the external name for the component trace; it is provided for an IBM-supplied component and must be provided by a system programmer for an installation-supplied application trace. This parameter is required.

SUB=(*subname*)

Identifies a sublevel trace for a component or application with multiple traces. *Subname* is defined by the component or installation-supplied application.

The SUB parameter is limited to a single *subname*; multiple *subnames* are not supported.

If the sublevel trace name contains any national characters (\$ # @), you must enclose the name in quotes. Otherwise, quotes are not required. In all cases you may specify the alphabetic characters in upper or lower case.

If *subname* is a head level, all of the head's sublevel traces that are defined with a LIKEHEAD=YES parameter inherit the options specified in the reply to this command. Therefore, the options you specify for a head level can affect many sublevel traces.

Omitting the SUB parameter for a head level that is defined with HEADOPTS=YES affects all sublevel traces with the LIKEHEAD attribute.

PARM=mem

Identifies a parmlib member that contains the options to be used for tracing. Using a parmlib member allows the operator to initiate the trace, change it, or stop it without a message prompting for options.

Any option specified on the TRACE command overrides the option specified in the parmlib member.

TT Specifies the transaction trace options. The system programmer will supply the trace parameters. To determine which trace parameters to use, see "Specifying TRACE TT Options." For further information, see "Transaction Trace" in *z/OS MVS Diagnosis: Tools and Service Aids*.

Specifying TRACE CT Options

In response to a TRACE CT,ON command without the PARM parameter, the system prompts you to specify the component trace options you want with message ITT006A. Use the REPLY command to respond to that message. You can specify each option, ASID, JOBNAME, OPTIONS, or WTR only if the component or application supports it. You can enter the DISPLAY TRACE command before entering a TRACE CT command to verify which options are supported.

The REPLY command syntax for specifying TRACE CT options is:

```
R id[,ASID=(nnnn[,nnnn]...)]
      [,JOBNAME=(name[,name]...)]
      [,OPTIONS=(name[,name]...)]
      [,WTR={membername|DISCONNECT}]
      [,CONT|,END ]
```

Note: When you specify CONT or END, it must be the last parameter on the input line.

id The identification number (0-9999), as specified on the prompting message.

ASID=(nnnn[,nnnn]...)

Specifies the address space identifiers (ASIDs) of address spaces to be used as a filter for tracing. Events in these ASIDs are to be recorded by the component trace.

The parameter contains a list of 0 to 16 hexadecimal ASIDs separated by commas. An empty ASID list, ASID=(), turns off filtering by address spaces. In the ASID parameter, list all address spaces to be traced; address spaces specified for previous traces will not be traced unless listed.

JOBNAME=(name[,name]...)

Specifies the names of jobs to be used as a filter for tracing. Events in these jobs are to be recorded by the component trace.

The parameter contains a list of 0 to 16 job names separated by commas. An empty job list, JOBNAME=(), turns off filtering by jobs. In the JOBNAME parameter, list all jobs to be traced; jobs specified for previous traces will not be traced unless listed.

TRACE Command

OPTIONS=(option[,option]...)

Specifies component-specific options for tracing. See *z/OS MVS Diagnosis: Tools and Service Aids* for the options for an IBM-supplied component that supports component tracing. Refer to the installation-supplied application for the options for the application.

The options for some IBM-supplied component traces can be changed while the trace is running; to change the options for others, stop the trace and restart it with the new options. An installation-supplied application trace defined with MOD=YES in the CTRACE DEFINE macro can be changed while running.

The options for a head level defined with HEADOPTS=NO cannot be changed. When you change a head level that was defined with HEADOPTS=YES, all of the sublevel traces currently in LIKEHEAD status will also be changed. Therefore, a change may cascade down a number of levels.

Omit OPTIONS to allow the component to use its default options.

WTR=membername

WTR=DISCONNECT

Connects or disconnects the component trace external writer and the trace. *membername* identifies the name of the member that contains the source JCL that invokes the external writer. The member can be a SYS1.PROCLIB catalogued procedure or a job. The *membername* in the WTR parameter must match the *membername* in a previous TRACE CT,WTRSTART command.

WTR=DISCONNECT disconnects the writer and the trace. The component continues tracing and placing the trace records in the address-space buffer, but stops passing trace records to the external writer.

You must also specify a TRACE CT,WTRSTART or TRACE CT,WTRSTOP command to start or stop the writer.

CONT or END

Specifies that the reply continues on another line. The system reissues the same prompting message. You then can continue the reply. You can repeat any parameters on the continuation line, except END. Repeated parameters are strung together. They do not overlay each other. You must specify END to complete the response. END identifies the end of the REPLY.

Example 1

To turn off system trace, enter:

```
TRACE ST,OFF
```

Example 2

System trace is off. The last time system trace was on, 16K of storage was set aside for system trace table entries for each processor. To turn on system trace, turn on the branch tracing function of system trace, and increase the space for system trace table entries to 250K for each processor, enter:

```
TRACE ST,250k,BR=ON
```

Example 3

To turn on master trace, if it is not already on, and to define a master trace table of 100K, enter:

```
TRACE MT,100k
```

Example 4

To modify component trace for the GRS component by specifying a different parmlib member, enter:

```
TRACE CT,ON,COMP=SYSGRS,PARM=CTYGRS02
```

Specifying TRACE TT Options

The TRACE TT command specifies the definition of a transaction trace (TTrace) filter set, definition or change in the TTrace processing options, removal of a TTrace filter set, or stopping of TTrace in the sysplex.

COLL | C=collection

specifies a collection name to be used as a filter for tracing. Collection name identifies the customer-defined name for a group of associated packages.

The collection name consists of 1 through 18 characters.

CON=connection type

specifies a connection type to be used as a filter for tracing. Connection type identifies the type associated with the environment creating the work request, which may reside anywhere within the network.

The connection type consists of 1 through 8 characters.

COR=correlation info

specifies correlation information to be used as a filter for tracing. Correlation information identifies the name associated with the user or program creating the work request, which may reside anywhere within the network.

The correlation information consists of 1 through 18 characters.

LU=logical unit name

specifies the LUNAME to be used as a filter, allowing work related to this logical unit to be traced.

The luname consists of 1 through 8 characters.

LVL=level

specifies the trace level indicator to be assigned to this filter set. Valid trace level indicators are 1, 2, 3, and 128-255.

The level consists of 1 through 3 numeric characters.

The default trace level value is 2.

Trace level indicators 0, and 4 through 127 are reserved.

NET=netid

specifies a NETID name to be used as a filter, allowing work related to this network ID to be traced.

The netid consists of 1 through 8 characters.

Specifying NET requires the specification of the LU keyword.

PKG=package

specifies a package name to be used as a filter for tracing. The package name identifies a set of associated SQL statements.

The package name consists of 1 through 8 characters.

PLAN|PL=plan name

specifies a name to be used as a filter for tracing. The plan name identifies the access plan name for a set of associated SQL statements.

TRACE Command

The plan name consists of 1 through 8 characters.

PRF=*perform*

specifies perform information to be used as a filter for tracing. Perform information identifies the performance group number (PGN) associated with the work request.

The perform information consists of 1 through 8 characters.

PROCIPR=*proc name*

specifies the proc name to be used as a filter for tracing. Proc name identifies the DB2-stored SQL Procedure name associated with the work request.

The proc name consists of 1 through 18 characters.

PRS=*process*

specifies process information to be used as a filter for tracing. Process information identifies the process name associated with the work request.

The process information consists of 1 through 32 characters.

SUB=*subsystem*

specifies subsystem information to be used as a filter for tracing. Subsystem information identifies character data related to the work request that is passed by the work manager for use in WLM classification.

The subsystem information consists of 1 through 18 characters.

TC=*transaction class*

specifies the transaction class to be used as a filter for tracing. Transaction class identifies a class name within the subsystem.

The transaction class information consists of 1 through 8 characters.

TRANIT=*transaction_name*

specifies a transaction name to be used as a filter for tracing.

Although the transaction name has specific meaning to many subsystems (those that exploit WLM classify), the transaction name identifies the name of the work request desired to be traced across the set of components that may handle it. See *z/OS MVS Planning: Workload Management* for information on transaction name and jobname.

The transaction name consists of 1 through 8 characters.

USERIU=*userid*

specifies a userid to be used as a filter for tracing.

userid is the name of a single userid for whom the tracing is to be done.

The userid consists of 1 through 8 characters.

WTR=*membername*|**STOP**

indicates that the external writer is to be started for recording transaction trace entries in a data set, or stopped if it was already started. This is an optional keyword.

membername

identifies the member that contains the source JCL that invokes the component trace external writer. The system opens the data sets that the writer uses. The member can be a SYS1.PROCLIB catalogued procedure or a job.

If WTR= is not specified, the trace data is saved in the TTRACE data space and can be obtained by dumping the data space.

The external writer may be started on the first invocation of the TRACE TT command or on any subsequent TRACE TT command prior to the TRACE TT,OFF command. Only one external writer may be started on any instance of transaction trace.

The external writer proc name **must** be the same on every system in the sysplex that is participating in transaction trace. Each system may define a proc with the same name or share a common writer proc.

If trace is already running, the writer may be started with a TRACE TT,WTR= command without specifying any filter parameters.

Trace entries are recorded in the external writer data set from the time the external writer is started until the external writer is stopped with a TRACE TT,WTR=STOP or trace is turned off with a TRACE TT,OFF,ALL. When the end of the data set is reached, it starts writing from the top of the data set, overlaying the oldest entries. Specifying a "membername" when transaction trace is not active (that is, no filter has been specified) results in an error message.

STOP

indicates that the external writer that was previously started needs to be stopped. No more trace entries are recorded in the external writer data set.

Any trace records currently in the buffers are written prior to stopping the external writer. An error message is issued if WTR=STOP is issued when no external writer is active.

BUFSIZ=nnnK or nnM

specifies the size of the TTrace data space the system is to use, in kilobytes (K) or megabytes (M).

nnnK is the buffer size in kilobytes, where nnn is a decimal number from 16 to 999.

nnM is the buffer size in megabytes, where nn is a decimal number from 1 to 32.

The buffer size may be changed anytime when trace is active. This includes when the initial TRACE TT command is issued. Specifying a buffer size when transaction trace is not active (that is, no filter has been specified) results in an error message. The default size is 1M.

The scope of this buffer size change affects TTrace processing until the buffer size is changed on a subsequent command or after a system IPL. That is, the buffer size persists if TTrace is turned off and turned on at a later time.

The buffer size is rounded up to a page boundary (that is, multiples of 4K).

This is an optional keyword.

LATENT=(YIN)

specifies whether latent transactions need to be traced.

A *latent transaction* is defined by *all* of the following:

- The transaction is currently active in the system.
- The transaction is marked for tracing.
- The filter value used to mark the transaction eligible for tracing is no longer active, that is, TRACE TT,OFF=nn was issued to turn that filter off.

Y specifies that latent transactions, if any, can continue to be traced. Y is the default value.

TRACE Command

N specifies to not trace latent transactions. Any current active transactions that do not have an active filter will not be traced from this point on.

The scope of the latent parameter setting affects TTrace processing until the parameter is changed on a subsequent command or after a system IPL. That is, the parameter value persists if TTrace is turned off and then turned on at a later time.

OFF=(nn|ALL)

turns the specified active transaction trace OFF across the sysplex.

nn specifies the filter set number of the trace to be turned off.

nn is an integer from 1-99 that identifies a particular transaction trace filter set. Use DISPLAY TRACE,TT to display the active TTrace filter sets.

ALL

specifies that all the active transaction traces should be turned off across the sysplex.

No additional trace entries are recorded. Any data in the trace buffers is written out to the data space and external writer data set. The external writer, if it was started, is also stopped.

A message is issued to the operator console to indicate that transaction trace is no longer active on the system and a DUMP command may be issued to access the trace in the data space.

TRACK Command

TRACK COMMAND NO LONGER SUPPORTED

The TRACK command is not supported on systems at z/OS V1R7 and higher.

Use the TRACK command to request a periodic display of job information on display consoles. Although you can enter this command from a display console or a non-display console, the system rejects the command when the TRACK output is directed to a non-display console or a console operating in message stream mode. The information requested is displayed at 30-second intervals unless you specify a different time interval with the CONTROL T command.

Note: To stop the TRACK display, use the STOPTR command, described earlier in this chapter.

Scope in a Sysplex

The TRACK command has sysplex scope only when you specify L=. See “Using Commands That Have Sysplex Scope” on page 1-11 for an explanation of sysplex scope.

Syntax

The complete syntax for the TRACK command is:

```
TR {TS|JOBS|J|A} [,LIST|,L] [,USERID=userid] [,L={a|name|name-a}]
```

Parameters

JOBS, J, TS, or A

When you specify **any** of the above operands, the system periodically displays all of the following overview information:

- The number of active batch jobs (MOUNT commands in execution are treated as active batch jobs)
- The number of started tasks
- The number of logged-on time-sharing users
- The maximum number of time-sharing users allowed to be logged on under TSO/VTAM
- The number of active APPC/MVS transaction programs
- The number of z/OS UNIX System Services address spaces

LIST or L

For TR JOBS or TR A, the system also periodically displays detailed information for active jobs. For TR TS or TR A, the system also periodically displays detailed information for each logged-on time-sharing user. See “Displaying System Activity” on page 4-149 for a description of the detailed information you get from LIST.

USERID=*userid*

A filter to show only the work running for *userid*. This userid can be specified on the *USER=* keyword in JCL or the userid that requested this transaction.

TRACK Command

Note: When the `USERID=userid` is specified on TRACK commands issued after a TRACK is already active, it only replaces the current *userid*. It does not add to a list of userids.

L=a, name, or name-a

The display area (*a*), console name (*name*), or both (*name-a*) of the active MCS or SMCS console where the requested display is to appear. TRACK is not valid for extended MCS consoles or system consoles. If you omit this operand, the display is presented in the first available display area or message area of the console through which you enter the command.

Example

To display periodically the overview information listed above for all units of work in the system, enter:

```
TR A
```

Because the L= operand is not specified, one of two situations must exist:

- The console issuing the command is a display console and is to receive the specified output.
- A MSGRT command has previously specified the routing instructions for TRACK command output.

UNLOAD Command

Use the UNLOAD command to unload mounted tape or DASD volumes.

In a JES3 complex, use the UNLOAD command carefully. When you request that a volume be removed from a JES3-managed direct access device, you must enter an UNLOAD command on each system in the complex that has access to the device. In addition, if you are requesting that a volume be removed from a JES3-managed unit, you must also enter a JES3 unload command (*MODIFY,S,U=...). See *z/OS JES3 Commands*.

Scope in a Sysplex

The UNLOAD command has sysplex scope only when you issue the command against an automatically switchable tape device. See “Using Commands That Have Sysplex Scope” on page 1-11 for an explanation of sysplex scope.

Syntax

The complete syntax for the UNLOAD command is:

```
U [/]devnum
```

Parameters

[/]devnum

The device number of the device to be unloaded. A device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/).

VARY Command

VARY Command

Table 4-43 summarizes the information that the VARY command provides. Use it to access the pages on which you can find details about a particular use of the VARY command.

Uses of the VARY command for TCPIP activity and functions are described in *z/OS Communications Server: IP Configuration Reference*.

Uses of the VARY command for VTAM network activity and functions are described in *z/OS Communications Server: SNA Operation*.

Note: To change the online or offline status of processors, channel paths, ESTOR elements, and central storage sections, see the CONFIG command.

Table 4-43. Summary of the VARY Command

Command:	Topic:
VARY CN,ACTIVATE	"Controlling Problem Determination Mode for the System Console" on page 4-618
VARY CN,DEACTIVATE	"Controlling Problem Determination Mode for the System Console" on page 4-618
VARY CN,AUTOACT=	"Controlling Problem Determination Mode for the System Console" on page 4-618
VARY CN	"VARY CN command" on page 4-621
VARY ...,CONSOLE	"VARY CONSOLE command" on page 4-626
VARY ...,HARDCPY	"Controlling Hardcopy Processing" on page 4-628
VARY conspec...,ONLINE VARY conspec...,OFFLINE	"Placing a Console Online or Offline" on page 4-631
VARY devspec...,AUTOSWITCH	"Defining a Tape Device as Automatically Switchable" on page 4-632
VARY devspec...,ONLINE VARY devspec...,OFFLINE	"Placing an I/O Device or a Range of I/O Devices Online or Offline" on page 4-633
VARY GRS	"Controlling a Global Resource Serialization Complex" on page 4-638
VARY PATH	"Placing an I/O Path or Paths Online or Offline" on page 4-639
VARY SMS CFCACHE VARY SMS CFVOL VARY SMS FALBACK VARY SMS,SHCDS VARY SMS,SHCDS CFRESET VARY SMS,SMSVSAM	"Changing the state of coupling facility cache structures and volumes" on page 4-642
VARY SMS,DRIVE VARY SMS,LIBRARY	"Placing an Optical Drive or Library Online or Offline" on page 4-645
VARY SMS,LIBRARY	"Placing a System-Managed Tape Library Online or Offline" on page 4-645
VARY SMS,PDSE,ANALYSIS	"Analyzing the State of the PDSE Subsystem" on page 4-645
VARY SMS,PDSE,FREELATCH	"Releasing PDSE Latches" on page 4-645

Table 4-43. Summary of the VARY Command (continued)

Command:	Topic:
VARY SMS,PDSE,MONITOR,...	"Modifying processing of PDSE monitor" on page 4-645
VARY SMS,PDSE,MONITOR	"Display current state of the PDSE monitor" on page 4-646
VARY SMS,PDSE1,TERMINATESERVER[,QUIESCE(duration 3),XQUIESCE(duration 5)] VARY SMS,PDSE1,ACTIVATE[,COMMONPOOLS(NEWIREUSE) VARY SMS,PDSE1,RESTART[,QUIESCE(duration 3),XQUIESCE(duration 5)] VARY SMS,PDSE1,RESTART[,QUIESCE(duration 3),COMMONPOOLS(NEWIREUSE)]	<i>z/OS DFSMSdfp Diagnosis</i>
VARY SMS,STORGRP VARY SMS,VOLUME	"Changing the SMS Status of a Storage Group or Volume" on page 4-646
VARY SMS,TRANVSAM	"Controlling DFSMStvs processing" on page 4-648
VARY SWITCH	"Placing a Switch Port Online or Offline" on page 4-658
VARY WLM,APPLENV	"Controlling an Application Environment" on page 4-659
VARY WLM,POLICY	"Activating a Service Policy" on page 4-662
VARY XCF	"Removing a System from the XCF Sysplex" on page 4-664

You cannot specify the names of extended MCS consoles or of a system console in the following VARY commands:

- VARY CONSOLE
- VARY OFFLINE
- VARY ONLINE

You cannot specify the names of SMCS consoles in the following VARY commands:

- VARY CONSOLE
- VARY ONLINE

Scope in a Sysplex

The following table describes the conditions under which the VARY command has sysplex scope. See "Using Commands That Have Sysplex Scope" on page 1-11 for an explanation of sysplex scope. If a command has no entry under "Conditions", then the command has sysplex scope under all circumstances and for all variations.

Table 4-44. Sysplex Scope for VARY Command

Command	Conditions
VARY CN	Has sysplex scope unless all of the following are true: <ul style="list-style-type: none"> • You issue VARY CN(<i>conspec</i>),ONLINE without specifying SYSTEM=. • You do not specify SYSTEM= in the CONSLxx member of Parmlib that defines this console. • The console has never been active in the sysplex.

VARY Command

Table 4-44. Sysplex Scope for VARY Command (continued)

Command	Conditions
VARY SMS, STORGRPIVOLUME	Has sysplex scope under these conditions only: <ul style="list-style-type: none">• You specify (<i>storgrp/volume,ALL</i>) and all systems in the sysplex are in the same SMS complex.• You specify (<i>storgrp/volume, system</i>) where <i>system</i> is a system group, and the system group exactly matches the sysplex (that is, none of the systems in the sysplex is explicitly defined to SMS).
VARY SWITCH	Logical partition cluster scope — see the “Intelligent Resource Director” chapter in <i>z/OS MVS Planning: Workload Management</i> for more information.
VARY XCF	All
VARY WLM	All

Syntax

The syntax for each variation of the VARY command is shown immediately preceding its respective parameter list.

VARY or V

Controlling Problem Determination Mode for the System Console

Use this form of the VARY command to control problem determination mode for the system console. If you are not sure of the system console names, enter the DISPLAY CONSOLES command with KEY=SYSCONS.

Problem determination mode allows you to receive messages and send commands on the system console to debug hardware and software problems. After initialization, the system console is NOT in problem determination mode and receives a minimal set of messages. You can place the system console in problem determination mode by entering the VARY CN,ACTIVATE command.

The first time the system console is placed into problem determination mode, it receives its attributes from its CONSOLE statement in CONSOLxx. On subsequent ACTIVATEs, it receives the attributes it had when it was last DEACTIVATED. (Use the DISPLAY CONSOLES command to see if the system console is in problem determination mode. If it is, the display will show COND=(A,PD).)

Use the DISPLAY CONSOLES command to see which attributes are in effect for the system console when it is in problem determination mode. Use the VARY, CONTROL, or MONITOR commands to change console attributes for the system console when in problem determination mode.

If the system console is already in problem determination mode, the system rejects the command.

VARY CN,ACTIVATE Command

The syntax of the VARY CN,ACTIVATE command is:

VARY CN(conspec1|*),{ACTIVATE|ACT}

CN(*conspec1 or **)

Change the mode of the specified system console to problem determination mode. *conspec1* is the system console name. An asterisk (*) indicates the system console from which you enter the command. You must enter the command from the system console that you want to activate.

ACTIVATE or ACT

The system is to activate problem determination mode for the specified system console.

Note: If you did not specify message level for the system console in CONSOLxx, the system console receives all messages except broadcast messages.

For information on the effect of issuing a VARY CN(syscons),ACTIVATE command while an AUTOACT group is defined, see “Suspending AUTOACT processing” on page 4-620.

Restrictions:

1. The VARY CN,ACTIVATE command affects only the system console on which you enter the command. You cannot enter multiple system console names.
2. If you enter an asterisk (*) on the CN parameter of the VARY CN,ACTIVATE command, do not specify system symbols in the command. The system will not substitute text for the system symbols.

VARY CN,DEACTIVATE Command

You can remove the system console from problem determination mode by entering the VARY CN,DEACTIVATE command. When the command takes effect, the system console receives a minimal set of messages. You cannot use the VARY, CONTROL, or MONITOR commands to change console attributes for the system console when it is NOT in problem determination mode.

If you issue the command for a system console that is NOT in problem determination mode, the command has no effect. The VARY CN,DEACTIVATE command can affect any system console in the sysplex. Thus, you can route the command from any console with the proper authority to the target system console.

For information on the effect of issuing a VARY CN(syscons),DEACTIVATE command while an AUTOACT group is defined, see “Suspending AUTOACT processing” on page 4-620.

The syntax of the VARY CN, DEACTIVATE command is:

```
VARY CN(conspec1|*),{DEACTIVATE|DEACT}
```

CN(*conspec1 or **)

Remove the specified system console from problem determination mode. *conspec1* is the system console name. An asterisk (*) indicates the system console on which the command is entered. You can enter the command from any authorized console (master authority) for the designated system console you want to deactivate.

DEACTIVATE or DEACT

The system is to deactivate problem determination mode for the specified system console.

VARY Command

VARY CN,AUTOACT= Command

You can configure the system console so that the commands VARY CN,ACTIVATE and VARY CN,DEACTIVATE are issued automatically to ensure that messages are sent to the system console when no other consoles are available. If you specify an automatic activate group (AUTOACT) for the system console, the VARY,CN AUTOACT= command uses the name of a console group from CNGRPxx.

If an AUTOACT group is defined and not suspended:

- When all of the consoles in AUTOACT are inactive, the system console will automatically be placed into PD mode.
- If any console in AUTOACT becomes active after the system console has automatically been placed into PD mode, the system console will then automatically be removed from problem determination (PD) mode.
- The consoles in the AUTOACT group can be of any type (MCS, SMCS or EMCS).
- In a sysplex environment, the consoles in the AUTOACT group can be attached to any system in the sysplex.

Suspending AUTOACT processing: If a VARY CN(syscons),ACTIVATE or a VARY CN(syscons),DEACTIVATE command is issued while an AUTOACT group is defined, and the command was NOT issued automatically by the AUTOACT processing, then AUTOACT processing will be suspended. This suspension means that the system console's mode will not be changed automatically by MVS. When the opposite (ACTIVATE or DEACTIVATE) command is issued, AUTOACT processing will resume.

The syntax of the VARY CN,AUTOACT= command is:

```
VARY CN(conspec1|*),{AUTOACT=console group name|*NONE*}
```

CN(conspec1 or *)

Add, change, or remove the specification of an automatic activate group for the specified system console. *conspec1* is the system console name. An asterisk (*) indicates the same console from which you enter the command. You can enter the command from any authorized console (master authority) for the designated system console you want to change.

Note: This command will not work if *conspec1* is not a system console.

console group name or *NONE*

console group name indicates the new or changed group name which is to be the automatic activate group for the system console. *console group name* must be defined in the current CNGRPxx. *NONE* indicates that the automatic activate group specification is to be removed from the system console. The system console will not be activated or deactivated automatically.

Controlling MCS and SMCS Consoles

Use the following forms of the VARY command to:

- Change the authority of a console
- Define or change the routing codes that send system messages to a console

The consoles specified on these commands must be defined as consoles in the CONSOLxx Parmlib member. Extended MCS consoles can also be accepted.

VARY CN command

The VARY CN command is used to set attributes for MCS, SMCS and extended MCS consoles. The syntax of the VARY CN command is:

```
V CN{(*|conspec1[,conspec1]... )}  
  [,AMSCOPE=([*][,name[,name]...])]  
  [,AUTH={ALL|INFO|MASTER|([SYS][,IO][,CONS])}]  
  [,AROUT=(rtcode[,rtcode]...)]  
  [,DMSCOPE=([*][,name[,name]...])]  
  [,DROUT=(rtcode[,rtcode]...)]  
  [,INTIDS={Y|N}]  
  [,LOGON={OPTIONAL|REQUIRED|AUTO|DEFAULT}]  
  [,LU={uname|*NONE*}]  
  [,MSCOPE={(*ALL)|{([*][,name[,name]...])}}]  
  [,OFFLINE[,FORCE]|,ONLINE[,SYSTEM=sysname][,FORCE]]  
  [,ROUT={ALL|NONE}(rtcode[,rtcode]...)]  
  [,UNKNIDS={Y|N}]
```

Notes:

1. You do not have to put a single AUTH=operand, or a single MSCOPE= operand, or a single routing code in parentheses. You must put a single console in parentheses.

CN

Change the indicated authority for the specified console name(s). When you are uncertain of the current console names, enter a DISPLAY CONSOLES command; see “Displaying Console Status Information” on page 4-112 for further information on this command.

conspec1

Represents one of the following:

nnnnnnnn The name of the console device

***** The console that you are currently issuing commands from.

AMSCOPE

Adds one or more system names to the list of systems from which the console can receive messages. If you specify more than one system name, separate the names with commas.

You can also specify an asterisk (*), which means that the console is to receive messages from the system to which it is attached. If the console is later attached to a different system, the console will start receiving message from that different system. For example, specifying AMSCOPE(SYS1,SYS2,*) causes a console to receive messages from systems SYS1, SYS2, and the system to which the console is attached.

AUTH=

Specifies the system command groups that the console is authorized to enter. This command requires MASTER authority and might be protected with a security product OPERCMDS resource class with the MVS.VARYAUTH.CN profile, which requires CONTROL authority. In addition, the MVS.VARY.CN profile, which requires UPDATE authority, must also be specified. Table 3-5 on page 3-24 lists the commands and their associated groups.

MASTER

One or more specified consoles are authorized to enter all system operator commands.

VARY Command

ALL

One or more specified consoles are authorized to enter INFO, SYS, IO, and CONS commands.

INFO

One or more specified consoles are authorized to enter only INFO commands.

(SYS, IO, CONS)

One or more specified consoles are authorized to enter INFO commands as well as SYS, IO, or CONS commands (depending on which commands you include). Enter them in any order.

AROUT=*rtcode*

The system is to add one or more specified routing codes or the routing codes in the specified range(s) to the routing codes already defined for the console. You can specify single routing codes (*rtcode*), ranges of routing codes (*rtcode-rtcode*), or a combination of single routing codes and ranges of routing codes on the same AROUT operand. For example, specifying AROUT(2,11-14,28) adds routing codes 2, 11, 12, 13, 14, and 28.

DMSCOPE

Deletes one or more system names from the list of systems from which the console can receive messages. If you specify more than one system name, separate the names with commas.

If the console currently is defined to receive messages from the system to which it is attached, you can also remove that definition. To do this, specify an asterisk on the DMSCOPE parameter.

For example, if you have issued VARY CN AMSCOPE(SYS1,SYS2,*), the console receives messages from SYS1, SYS2, and whatever system the console is currently attached to.

- If the console is attached to SYS1, it receives messages from SYS1 and SYS2. Issuing VARY CN DMSCOPE(*) has no effect on the console until the console is moved to a system other than SYS1 or SYS2. When it is moved to the other system, the console does not receive messages from that system.
- If the console is attached to SYS3, it receives messages from SYS3 as well as SYS1 and SYS2. When you issue VARY CN DMSCOPE(*), the console immediately stops receiving messages from SYS3.

DROUT=*rtcode*

The system is to remove the specified routing code(s) or the routing codes in the specified range(s) from the routing codes already defined for the console. You can specify single routing codes (*rtcode*), ranges of routing codes (*rtcode-rtcode*), or a combination of single routing codes and ranges of routing codes on the same DROUT operand. For example, specifying DROUT(2,11-14,28) deletes routing codes 2, 11, 12, 13, 14, and 28.

INTIDS=

Whether the specified console can receive messages which are directed to console id zero. These messages are usually the command responses for internally issued commands.

- Y** The specified console is to receive these messages.
N The specified console is not to receive these messages. This is the default value.

LOGON=value

This allows the operator to change the console's unique LOGON value of the specified console. *Value* is one of the following:

OPTIONAL

Indicates that this console does not require LOGON

REQUIRED

Indicates that this console requires an operator to logon before issuing commands

AUTO

Indicates that this console is automatically logged on

DEFAULT

Indicates that this console is to use the LOGON specification on the DEFAULT statement

OPTIONAL, REQUIRED, AUTO and DEFAULT all mean the same as they do on the LOGON keyword of the CONSOLE statement in CONSOLxx. See *z/OS MVS Initialization and Tuning Reference* for more information.

| This command requires MASTER authority and might be protected with a security product OPERCMDS resource class with the MVS.VARYLOGON.CN profile, which requires CONTROL authority. In addition, the MVS.VARY.CN profile, which requires UPDATE authority, must also be specified.

LU=

*luname OR *NONE**

Allows the operator to change or turn off the predefined LU name of a SMCS console. See *z/OS MVS Planning: Operations* for more information on predefined LUs. This command is only valid for SMCS consoles. If the command is issued for a non-SMCS console, then message IEE044I is issued. If the console is active, the only LU value that would be accepted is the LU that the console is active on or *NONE*. If a different LU is specified for an active console, then message IEE045I is issued. If the console is not active, any LU can be specified. This command requires MASTER authority. It might be protected with the RACF OPERCMDS resource class with the MVS.VARYLU.CN profile, which requires CONTROL authority. In addition, the MVS.VARY.CN profile, which requires UPDATE authority, must also be specified.

Value Range: Is from 2 to 8 characters. The first character must begin with the letters A through Z or with a #, \$, or @; the remaining characters can be A through Z, 0 through 9, or #, \$, or @.

OFFLINE

Deactivates an MCS or SMCS console and places the device in offline status (has the same effect as VARY device,OFFLINE for the device, when VARY device,OFFLINE is issued on the system where the console is attached). VARY CN(...),OFFLINE is sysplex-wide in scope, and does not have to be issued from the system where the console is active.

MSCOPE

In a sysplex, specifies which systems the console is to receive messages from.

***ALL**

The console will receive system messages from all active systems in the sysplex.

VARY Command

- * The console will receive system messages from the system it is attached to.

name

The console will receive system messages from the specified system name or system names in the sysplex.

ONLINE

Activates a device defined in the CONSOLxx parmlib member as an MCS console. MVS determines the system on which to activate the console in the following order:

1. The system specified on the SYSTEM keyword (if you specify SYSTEM on the VARY CN command)
2. The system on which this console was last active (if the console was previously active)
3. The system specified on the SYSTEM keyword in CONSOLxx (if you specified SYSTEM on the CONSOLE statement in CONSOLxx for this device)
4. The system on which the command is processed.

VARY CN(...),ONLINE does not accept extended MCS or SMCS console names as input.

If you specify a list of consoles, MVS attempts to process each console specified in the list:

- If a console is already active, MVS issues the same message (the DISPLAY C output - message IEE889I) as when activating a console.
- If a console is already active on another system (for example, if you specify SYSTEM=SYS1 and the console is already active on SYS2), then MVS issues message IEE605I.
- If the console cannot be activated, MVS issues a message appropriate to the situation. There are a number of such messages, including:
 - IEE025I (the console device has no logical paths)
 - IEE274I (the console name is not defined as a console in the sysplex)
 - IEE420I (the console is an extended MCS console)
 - IEE606I (the console name is not defined on this system)

FORCE

Activates an MCS console even when the console is being kept offline by a configuration manager. If you issue the VARY CN,ONLINE command and receive a message indicating that the device is being kept offline by a configuration manager, you can issue the VARY CN,ONLINE,FORCE command if necessary.

The FORCE keyword can only be specified if ONLINE is also specified.

When specified with the OFFLINE keyword, it acts just like a VARY dev,OFFLINE,FORCE command.

SYSTEM

Specifies the system on which the console is to be activated. SYSTEM overrides any default system specification (for example, the SYSTEM keyword on the CONSOLE statement). SYSTEM is valid only if you specify the ONLINE keyword.

The SYSTEM value applies to all values in the list of console names. If the system specified as the SYSTEM keyword value is not active in the sysplex, you will receive an error message.

ROUT=

The routing codes of messages the console(s) can receive. These codes replace those previously assigned. See Table 3-6 on page 3-30 for a list of routing codes.

You can specify single routing codes (*rtcode*), ranges of routing codes (*rtcode-rtcode*), or a combination of single routing codes and ranges of routing codes on the same ROUT operand. For example, specifying ROUT(2,11-14,28) assigns routing codes 2, 11, 12, 13, 14, and 28.

Note: The system processes the ROUT, AROUT, and DROUT operands in the order that you specify them in the command.

ALL

The console receives all system-to-operator messages.

NONE

The console does not receive any system-to-operator messages.

rtcode

The console receives all messages with the specified routing codes. You can specify single routing codes (*rtcode*) or ranges of routing codes (*rtcode-rtcode*) on the same ROUT operand.

UNKNIDS=

Whether the specified console can receive messages which are directed to "unknown" console ids. These ids are one-byte ids which the system cannot resolve.

Y The specified console is to receive these messages.

N The specified console is not to receive these messages. This is the default value.

Example 1

To assign master level authority to a console named REMOTE, enter:

VARY CN(REMOTE),AUTH=MASTER

Example 2

To limit console CON3 to receiving unsolicited messages from only systems SY3 and SY4, enter:

VARY CN(CON3),MSCOPE=(SY3,SY4)

Example 3

To add SY2 to the list of systems which will send unsolicited messages to console CON3, enter:

VARY CN(CON3),AMSCOPE=SY2

Example 6

To activate consoles FRED and STAN, enter:

V CN(FRED,STAN),ONLINE

Example 7

To activate consoles ABLE and BAKER for use on system SYS1, enter:

VARY Command

```
V CN(ABLE,BAKER),ONLINE,SYSTEM=SYS1
```

Example 8

To activate console FRED, and at the same time to authorize FRED to enter informational, console control and system control commands, and receive messages for the primary operator, the tape pool, and the tape library, enter:

```
V CN(FRED),ONLINE,AUTH=(CONS,SYS),ROUT=(1,3,5)
```

VARY CONSOLE command

The VARY CONSOLE command is used to activate and set attributes for MCS consoles. This command is not used for extended MCS or SMCS consoles.

The syntax of the VARY CONSOLE command is:

```
V {conspec2} (conspec2[,conspec2]...),CONSOLE  
  [,AROUT=(rtcode[,rtcode]...)]  
  [,AUTH={ALL|MASTER|INFO|([SYS][,IO][,CONS])}]  
  [,DROUT=(rtcode[,rtcode]...)]  
  [,ROUT={ALL|NONE|(rtcode[,rtcode]...)})]
```

Note: If you specify one device number, one AUTH=operand, or one routing code, you can omit the parentheses.

conspec2

conspec2 is one of the following:

[/]*devnum*

The device number of the console device.

O-[/]*devnum*

The device number of the console device preceded by the literal 'O-' to designate a device with output-only capability.

nnnnnnnn

The name of the console device as specified in the CONSOLxx Parmlib member CONSOLE statement (for example, 'TAPECNTL')

O-*nnnnnnnn*

The name of the console device as specified in the CONSOLxx Parmlib member CONSOLE statement preceded by the literal 'O-' and designates an output-only device (for example, 'O-TAPEPRNT').

The various types of *conspec* can be specified in any combination. If you specify only one *conspec*, you do not need to enter the parentheses.

A device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). You can precede the device number with a slash to prevent ambiguity between the device number and a console name.

CONSOLE

The unit is to be an active console.

AUTH=

Specifies the system command groups that the console is authorized to enter. This command requires MASTER authority and might be protected with a security product OPERCMDS resource class with the MVS.VARYAUTH.CONSOLE profile, which requires CONTROL authority. In

addition, the MVS.VARY.CONSOLE profile, which requires UPDATE authority, must also be specified. Table 3-5 on page 3-24 lists the commands and their associated groups.

ALL

The specified console(s) are authorized to enter INFO, SYS, IO, and CONS commands.

MASTER

The specified console(s) are authorized to enter all system operator commands.

INFO

The specified console(s) are authorized to enter only INFO commands.

(SYS, IO, CONS)

The specified console(s) are authorized to enter INFO commands as well as SYS, IO, or CONS commands (depending on which commands you include). Enter them in any order.

ROUT=

The routing codes of messages the console(s) can receive. These codes replace those previously assigned.

Note: The system processes the ROUT, AROUT, and DROUT operands in the order that you specify them in the command.

ALL

The console receives all system-to-operator messages.

NONE

The console does not receive any system-to-operator messages.

rcode

The console receives all messages with the specified routing codes. You can specify single routing codes (*rcode*) or ranges of routing codes (*rcode-rcode*) on the same ROUT operand.

AROUT=rcode

The system is to add the specified routing code(s) or the routing codes in the specified range(s) to the routing codes already defined for the console. You can specify single routing codes (*rcode*) or ranges of routing codes (*rcode-rcode*) on the same AROUT operand.

DROUT=rcode

The system is to remove the specified routing code(s) or the routing codes in the specified range(s) from the routing codes already defined for the console. You can specify single routing codes (*rcode*) or ranges of routing codes (*rcode-rcode*) on the same DROUT operand.

Example 1

To make consoles with device numbers 00C and 009 active, enter:

```
v (00c,009),console
```

Example 2

To make the console with device number 01FA an active console authorized to enter informational, system control, and console control commands and to receive messages for the primary operator, the tape pool, and the tape library, enter:

```
v /01fa,console,auth=(cons,sys),rout=(1,3,5)
```

VARY Command

Example 3

To remove routing codes 8, 31, 32, 33, 34, and 35 from the existing routing codes for the console with device number 2B0, enter:

```
v 2b0,console,drout=(8,31-35)
```

Example 4

To vary a console named TAPE online, add routing codes 7 through 12 to the current routing codes the console receives, enter:

```
VARY TAPE,CONSOLE,AROUT=(7-12)
```

Example 5

01F is to become an active console that is authorized to enter informational, system control, and console control commands and that receives messages for the primary operator, the tape pool, and the tape library.

```
V 01f,console,auth=(cons,sys),rout=(1,3,5)
```

Example 6

To remove routing codes 8, 31, 32, 33, 34, and 35 from the existing routing codes for console CON2, enter:

```
v con2,console,drout=(8,31-35)
```

Example 7

To vary a console named TAPE online, add routing codes 7 through 12 to the current routing codes the console receives, enter:

```
VARY TAPE,CONSOLE,AROUT=(7-12)
```

Controlling Hardcopy Processing

Use the HARDCPY form of the VARY command to do the following:

- Change the set of messages included in the hardcopy message set
- Assign either SYSLOG or OPERLOG to be the hardcopy medium
- Stop the hardcopy medium.

The system establishes hardcopy processing during system initialization based on the HARDCOPY statement in the CONSOLxx member of Parmlib. The ROUTCODE and CMDLEVEL parameters define messages that are included in the hardcopy message set.

The syntax of the VARY HARDCPY command is:

```
V [OPERLOG],HARDCPY[,CMDS|,NOCMDS|,STCMDS|,INCMDS]
    |SYSLOG
    [,AROUT=(rtcode[,rtcode]...)]
    [,DROUT=(rtcode[,rtcode]...)]
    [,ROUT={ALL|NONE
              }]
    { (rtcode[,rtcode]...)}
    [,OFF[,UNCOND]]
```

OPERLOG

The operations log is to be activated or deactivated.

When you omit the SYSLOG or OPERLOG operands, the system changes what goes into the hardcopy log, rather than the hardcopy log medium.

SYSLOG

The system log is to become the hardcopy medium.

HARDCPY

The system changes the hardcopy medium or the hardcopy message set, or both, depending on the options specified.

The following NOCMDS, INCMDS, STCMDS, and CMDS options correspond to the CMDLEVEL specifications of the HARDCOPY statement:

NOCMDS

The system is not to include operator commands or their responses in the hardcopy message set.

Note: If hardcopy support is required and you specify NOCMDS, the system will not allow NOCMDS and will choose CMDLEVEL=CMDS. (Hardcopy support is required when one or more display consoles are defined in a system.)

INCMDS

The system is to include operator commands and their responses, excluding any status displays, in the hardcopy message set.

STCMDS

The system is to include operator commands, their responses, and static status displays in the hardcopy message set.

CMDS

The system is to include operator commands, their responses, and all status displays (both static and dynamic) in the hardcopy message set.

Note: The following descriptor codes are associated with the above options:

Options	Descriptor Codes
NOCMDS	None
INCMDS	5
STCMDS	5, 8, 9
CMDS	5, 8, 9, 10

OFF

The system is to stop the hardcopy medium.

If you do not specify SYSLOG or OPERLOG, the system defaults to the hardcopy medium (SYSLOG) if it is active; otherwise, the system rejects the command. The system will not deactivate the operations log unless OPERLOG is specified. If you specify OPERLOG, the operations log must be active.

The system rejects this command if it would result in both the hardcopy log and the operations log becoming inactive.

When OFF is specified without UNCOND, it must be the last parameter.

UNCOND

Use UNCOND with OFF to specify that the system is to stop the hardcopy medium even if there is no other active hardcopy medium.

VARY Command

- If the hardcopy medium is SYSLOG, and it is the only hardcopy medium, the system saves messages for later hardcopy until the LOGLIM value is reached, after which the messages will be discarded.
- If the hardcopy medium is OPERLOG, no messages are saved.

Note: If the hardcopy medium is SYSLOG, and SYSLOG has never been activated in the system, messages will not be saved.

The use of UNCOND should be a temporary measure, and should be done only as a last resort in order to repair hardcopy functions. The installation might lose messages from hardcopy if too long a period elapses before the hardcopy medium is restored. The system issues message IEE012A when messages are no longer being saved.

When UNCOND is specified with OFF, UNCOND must be the last parameter.

ROUT=

The system is to include messages with the specified routing code or codes in the hardcopy message set. In addition to the routing codes you specify, the hardcopy message set also includes messages with the minimum set of routing codes (1,2,3,4,7,8,10, and 42) established at initialization by the HARDCOPY statement of CONSOLxx.

ALL

All routing codes (1-128) are used to select messages for the hardcopy message set.

NONE

No routing codes are used to select messages for the hardcopy message set.

rtcode

rtcode — *rtcode*

The specified routing code or codes are used to select messages for the hardcopy message set. *rtcode* is a decimal number from 1 to 128. You can specify a single routing code, a range of routing codes, or a combination of both.

AROUT

The system is to include messages with the specified routing code or codes in the hardcopy message set, in addition to any messages included because of prior routing code specifications.

rtcode

rtcode — *rtcode*

The specified routing code or codes, in addition to currently used routing codes, are used to select messages for the hardcopy message set. *rtcode* is a decimal number from 1 to 128. You can specify a single routing code, a range of routing codes, or a combination of both.

DROUT

The system is to stop including messages with the specified routing code or codes in the hardcopy message set.

rtcode

rtcode — *rtcode*

The specified routing code or codes are no longer used to select messages for the hardcopy message set. *rtcode* is a decimal number from 1 to 128. You can specify a single routing code, a range of routing codes, or a combination of both.

Note: At system initialization, processing of the HARDCOPY statement of the CONSOLxx member of Parmlib sets up a minimum set of routing codes (1,2,3,4,7,8,10, and 42) in addition to any other specified for the hardcopy message set.

The system processes the ROUT, AROUT, and DROUT operands in the order that you specify them.

Example 1

To include all operator commands, responses, and status displays (except dynamic status displays) in the hardcopy message set, enter:

```
V ,HARDCPY,STCMDS
```

Example 2

To have the hardcopy message set recorded on the system log, enter:

```
V SYSLOG,HARDCPY
```

Example 3

To add routing codes 11, 12, 13, 17, and 44 to the routing codes already defined for the hardcopy message set, enter:

```
V ,HARDCPY,AROUT=(11-13,17,44)
```

Example 4

To have the hardcopy message set recorded on the operations log, enter:

```
V OPERLOG,HARDCPY
```

Placing a Console Online or Offline

Use the following form of the VARY command to make a secondary console either online as an I/O device or offline. As a device, it will become inactive as a console. This command can be used to make an SMCS console offline, however, it cannot be used to make an SMCS console online.

```
V (conspec[,conspec]...),{OFFLINE|ONLINE}
```

conspec

conspec is the console device to be moved online or offline and is specified as one of the following:

[/]devnum

The device number of the console device.

O-[/]devnum

The device number of the console device preceded by the literal 'O-' to designate a device with output-only capability.

nnnnnnnnn

The name of the console device as specified in the CONSOLxx Parmlib member CONSOLE statement (for example, 'TAPECNTL')

O-nnnnnnnnn

The name of the console device as specified in the CONSOLxx Parmlib

VARY Command

member CONSOLE statement preceded by the literal ‘O’ to designate an output-only device (for example, ‘O-TAPEPRNT’).

[/]lowdevnum-[/]highdevnum

The *lowdevnum* is the device number of an input/output device that is the lower bound of a range of device numbers and *highdevnum* is the device number of an input/output device that is the upper bound of the range.

The various types of *conspec* can be specified in any combination. If you specify only one *conspec*, you do not need to enter the parentheses.

A device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). You can precede the device number with a slash to prevent ambiguity between the device number and a console name.

ONLINE

The system is to bring the specified device(s) online. If the specified device is a console, this command will de-activate it as a console.

OFFLINE

The system is to take the specified device(s) offline. If the specified device is a console, this command will de-activate it as a console.

Example 1

To take consoles 003 and 001 offline, enter:

V (003,001),OFFLINE

Example 2

To take consoles 3322,340A offline, enter:

V (/3322,/340A),OFFLINE

Defining a Tape Device as Automatically Switchable

Use the following form of the VARY command to change the automatically switchable characteristic (or the AUTOSWITCH attribute) of a tape device. The setting does not persist beyond the duration of the IPL.

VARY AUTOSWITCH is not valid for tape devices that are:

- Online. For a device that is online, issue the VARY OFFLINE command before you issue VARY AUTOSWITCH.
- Managed by JES3.

V {{(devspec[,devspec]...)},{[AUTOSWITCH|AS][,ON|OFF]}}

devspec

devspec is one of the following:

[/]devnum

The device is not assign capable.

lowdevnum-highdevnum

lowdevnum is the lower bound of a range of device numbers. *highdevnum* is the upper bound of the range.

A device number is 3 or 4 hexadecimal digits.

AUTOSWITCH or AS

The system is to turn on or off the AUTOSWITCH attribute of the tape device or range of tape devices you specified.

If you specify the VARY AUTOSWITCH command for a tape device that is online or managed by JES3, the system alerts you to the error.

- If you specify a list of devices (for example, VARY (281,282,283),AS,ON), the system returns a message for each device that is not valid.
- If you specify a range of devices (for example, VARY (281-283),AS,ON), you receive a DISPLAY command response that lists the status (including the AUTOSWITCH status) for devices in the specified range.

For more information about automatically switchable tape devices, see *z/OS MVS Setting Up a Sysplex*.

ON

The system is to turn on the AUTOSWITCH attribute for the device or devices you specified.

OFF

The system is to turn off the AUTOSWITCH attribute for the device or devices you specified.

Example 1

To turn on the AUTOSWITCH attribute for tape devices 282, 283, and 287, enter:

VARY (282,283,287),AS,ON

Placing an I/O Device or a Range of I/O Devices Online or Offline

Use the following form of the VARY command to place I/O devices online or offline.

```
V { (devspec[,devspec]...)},{ONLINE[,UNCOND][,FORCE]}
    {devspec}           |,SHR
                      |,RESET
    {OFFLINE[,FORCE]}  }
```

In a JES2 environment, use this command to specify that a cartridge tape device (such as a 3490) is to be shared among more than one system.

Use this form of the VARY command with care in a JES3 environment. For devices managed by JES3, issue a *VARY command instead of the MVS VARY command to change online or offline status. See "Placing Devices Online or Offline to JES3" in *z/OS JES3 Commands*.

devspec

devspec is one of the following:

[/]devnum

The device number of an I/O device.

O-[/]devnum

The device number of a console device preceded by the literal 'O-' to designate a device with output-only capability.

VARY Command

nnnnnnnnn

The name of a console device as specified in the CONSOLxx Parmlib member CONSOLE statement.

O-*nnnnnnnnn*

The name of a console device as specified in the CONSOLxx Parmlib member CONSOLE statement preceded by the literal 'O-' to designate an output-only device.

[I]*lowdevnum*-[I]*highdevnum*

lowdevnum is the device number of an I/O device that is the lower bound of a range of device numbers. *highdevnum* is the device number of an I/O device that is the upper bound of the range.

The various types of *devspec* can be specified in any combination, either a valid console name or a valid device number. The console name check will be made first and valid console names accepted. If you specify only one *devspec*, you do not need to enter the parentheses.

A device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). You can precede the device number with a slash to prevent ambiguity between the device number and a console name.

RESET, UNCOND and SHR keywords are ignored for console-capable devices. VARY of a range of devices is supported for console-capable devices.

ONLINE

The system is to bring the specified devices or ranges of devices online, that is, make the devices or ranges of devices available for allocation to problem programs and system tasks, if there is an online path to the devices. In a system-managed tape library, if the file tape drives within the library were placed offline with both the VARY device and VARY SMS commands, then you must issue both commands to place those devices online. If you bring a device online and you want the system to recognize a volume mounted while the device was offline, enter a MOUNT command for the device.

Notes:

1. If you specify a device that is not physically attached to its control unit, the system might consider the device operational and online. If an attempt is later made to allocate the device to a job, the attempt might fail, in which case the job would have to be canceled.
2. Bringing a device online cannot bring online I/O paths that have been taken offline with a VARY PATH command.
3. If you issue a VARY device ONLINE command for a device whose last path has been taken offline with a VARY PATH OFFLINE command, the system issues the following message:
IEE025I UNIT ddd HAS NO LOGICAL PATHS
4. If you specify a range of devices and any or all of them are not valid, you receive message IEE313I indicating the device numbers that are not valid.
5. For an automatically switchable tape device, the VARY ONLINE command brings the device online to the issuing system. This makes the device available to be allocated to that system. The device will be assigned when it is allocated.

UNCOND

The system is to bring the specified devices or range(s) of devices online, even if there are no paths to the devices or if the devices are pending offline and

boxed. The system ignores this operand if you specify it for a tape or direct access device that does not have a path. Use the UNCOND operand carefully because it causes inaccessible devices to appear accessible to some system components.

You can use the VARY bbbb,ONLINE,UNCOND command to correct problems when HyperPAV aliases are not properly used. Such an error condition occurs when non-FICON channels are configured online to a HyperPAV-capable control unit. Because MVS operates on base devices within that logical subsystem in base only mode, MVS cannot use alias devices for bases in the logical control unit. You can detect this condition by outstanding message IOS166E or by using the D M=DEV(bbbb) command, where bbbb is a base device in the logical control unit. MVS can use the HyperPAV aliases only after the non-FICON channels are removed from the HyperPAV configuration or these channels are configuring offline. However, it may be necessary under this or other conditions to force MVS to rediscover aliases that were not discovered previously. Use the VARY bbbb,ONLINE,UNCOND command where bbbb is an online base device in the logical control unit.

SHR

The system permits the sharable tape device you bring online to be shared among other processors. The system ignores the SHR keyword when specified for a device that is not assign capable. Share a tape device between processors only at the direction of the system programmer.

Do not use this keyword:

- For devices managed by JES3. JES3-managed devices are automatically sharable within the JES3 complex.
- For automatically switchable devices. If you use the SHR keyword, the system rejects the command because it is incompatible with automatic tape switching.

Do not confuse the sharing of tape devices (through the SHR keyword) with automatically switchable tape devices. The SHR keyword allows many systems sharing a key to access a single tape device at one time. Although automatically switchable devices are varied online to many systems at one time, only one system can actually access a device at one time.

RESET

The system is to bring online a device that is being kept offline because of a control-unit-initiated reconfiguration (C.U.I.R.).

OFFLINE

The system is to take the specified device(s) or range(s) of devices offline, that is, make the device(s) or range(s) of devices unavailable for allocation to application programs or system tasks. The system takes offline any device that is currently in use only after all the tasks to which it is allocated terminate.

Notes:

1. When you issue a VARY OFFLINE command, the system immediately places the specified device(s) in the "pending offline" state. A device in the pending offline state cannot be allocated — even if the job specifically requests the volume mounted on the device — unless the allocated/offline device installation exit is used to allow allocation or the operator selects the device in response to message IEF238D.
2. If a WTOR message IEF238D is outstanding, that is, has not yet been replied to, VARY OFFLINE activity cannot take place.

VARY Command

3. Also, while a message IEF238D remains outstanding, no other allocations can proceed for any devices in the same group as the device(s) waiting for the IEF238D response. For example, if a job is in allocation recovery trying to allocate a 3490 device (UNIT=3490), no other D/T3490 allocation will take place until the message IEF238D is satisfied. Similarly, if a job is trying to allocate a device in a device group named, say, CARTNY (UNIT=CARTNY), then no devices in CARTNY will be allocated until message IEF238D is satisfied.

The VARY device OFFLINE command takes effect immediately if the resources are available and the device is not allocated or when a system task starts.

When the specified device(s) is offline, you receive message IEF281I and, if you have not specified FORCE, the system rewinds and unloads all specified tape drives except for JES3-managed tape drives. All devices taken offline remain offline until you enter VARY device ONLINE commands for them or specify them in response to a system request for devices.

Note: When you specify VARY OFFLINE for a range of devices:

- If some or all of the devices are valid, you receive a status display of those devices in each range.
- If some or all of the devices are valid and are alternate path device numbers, or do not have device names assigned to their UCBs, you receive message IEE712I, stating that VARY processing has finished.
- If any of the devices are not syntactically valid, you receive message IEE313I indicating the device numbers that are invalid.

Make sure, when you specify a range of devices to be taken offline, that the range does not include any console device numbers.

CAUTION:

Never take any device offline if that device holds SYS1.DUMPxx data sets unless you first remove the SYS1.DUMPxx data sets from the system's list of SYS1.DUMPxx data sets with a DUMPDS DEL,DSN= command.

FORCE

You can specify FORCE with ONLINE or OFFLINE. The effect on the system is quite different, as described in the following.

FORCE specified with OFFLINE: When specified with the OFFLINE keyword, FORCE puts the specified device or devices immediately in pending offline status, even if they are currently active, allocated, reserved or assigned. The system stops I/O in progress on the devices and rejects future I/O requests to the devices as permanent I/O errors.

You can issue VARY device,OFFLINE,FORCE only from the master authority console.

If JES3 manages a device, VARY device,OFFLINE with FORCE still marks the device as pending offline to MVS. Any allocated device put into pending offline status with FORCE remains allocated to the user who owns it when you issue the VARY command. The system actually takes the device offline when all users have deallocated it. (The system does not allocate a device put into pending offline status with FORCE to any new job, regardless of how the job requests the device.)

You can also use VARY device,OFFLINE,FORCE to take any console.

When the system takes a device offline as a result of a VARY device,OFFLINE,FORCE command, you can usually bring the device online and make it available for I/O again by issuing a VARY device,ONLINE command. If, however, there are no physical paths to the device or the device is non-operational, you can place the device online only with a VARY device,ONLINE,UNCOND command. (Note that the system ignores the UNCOND operand if you specify it for a tape or direct access device.) A device brought online with UNCOND remains unavailable for I/O until you either supply it with a physical I/O path or make it fully operational again.

Notes:

1. Use VARY device,OFFLINE,FORCE only with great care in situations where the system is in serious trouble. Even if a specified device is already offline, the command immediately terminates all I/O in progress on the device; future I/O requests to the device are rejected as permanent I/O errors. Thus, the command might cause the loss of data, as well as a data integrity problem if the command prematurely releases a reserved device or unassigns an assigned device.
2. If you issue VARY device,OFFLINE,FORCE for a CTC adapter used by global resource serialization, be sure that you issue the command from both of the systems attached to that CTC adapter.
3. When you issue VARY ctc,OFFLINE,FORCE for a CTC adapter used by global resource serialization, you will be prompted by message ISG186D. Reply KEEP to take the CTC offline normally, allowing GRS to use the CTC when it is brought back online. Reply FREE to take the CTC away from GRS permanently. This will allow the installation to allocate the CTC to XCF signalling. Use LookAt (see “Using LookAt to look up message explanations” on page xviii) or use the *MVS System Messages* books to see more information about message ISG186D.

FORCE specified with ONLINE: When specified with the ONLINE keyword, FORCE places the specified device or devices online even if they are being kept offline by a configuration manager.

Example 1

To make devices 282, 283, and 287 available for system use, enter:

```
vary (282,283,287),online
```

Example 2

To take offline any devices in the range 283 through 287 and the range 130 through 135, enter:

```
V (283-287,130-135),OFFLINE
```

Example 3

To make device 282 available for system use even if there is no path to the device, enter:

```
V 282,ONLINE,UNCOND
```

Example 4

To terminate I/O to device 282, enter:

```
V 282,OFFLINE,FORCE
```

VARY Command

Message IEE800D asks you to confirm this command, which would cause the system to stop I/O in progress on device 282 and reject future I/O requests to the device as permanent I/O errors. Reply NO to message IEE800D to terminate the command and leave the status of the device unchanged. Reply YES to have the system stop I/O on the device, reject future I/O requests to the device, and mark the device pending offline (if device 282 is not already offline).

Controlling a Global Resource Serialization Complex

Use the VARY GRS command to:

- Temporarily remove a system from the global resource serialization ring complex.
- Restore a quiesced system to the global resource serialization ring complex.
- Remove a system from the global resource serialization ring complex.
- Rebuild a disrupted global resource serialization ring complex. (that is, a group of inactive or quiesced systems)

Note: Because the VARY GRS command is not valid when all of the systems in a global resource serialization complex are in the same sysplex, use the V XCF,sysname,OFFLINE command.

For more information on GRS and the sysplex, refer to *z/OS MVS Planning: Global Resource Serialization* and *z/OS MVS Setting Up a Sysplex*.

The following table shows the support for the VARY GRS commands during that time when an installation is migrating to a cross-system coupling facility sysplex.

Command	COMPLEX=SYSPLEX	Mixed Complex
VARY GRS,QUIESCE	Command rejected; message ISG153I issued	Command accepted
VARY GRS,RESTART	Command rejected; message ISG153I issued	Command accepted
VARY GRS,PURGE	Command rejected; message ISG153I issued	Command accepted:

Note: If the target system of a VARY GRS,PURGE command is a member of a sysplex with more than one system, it will be put into a non-restartable wait state.

Use the VARY GRS command mainly for recovery situations.

```
V GRS{{sysname}*|ALL},{RESTART|R}
      {{sysname}*},{QUIESCE|Q}
      {sysname},{PURGE|P}
```

sysname

The name of the system (specified on the SYSNAME system parameter). This name can be up to eight characters long and can contain any characters except commas and blanks.

- * The name of the current system (the system on which you enter the command). Specifying an asterisk means you want to change the current system's status in the global resource serialization ring.

ALL

You want to change the status of all systems in the global resource serialization ring.

RESTART or R

You want to restore a quiesced (or inactive) system to the global resource serialization ring or rebuild a global resource serialization ring that has been disrupted. (When rebuilding a disrupted ring, you can specify ALL with RESTART.) Once this command takes effect, the system processes all previously-suspended requests for global resources. Restarting a system or restarting the ring requires an active system. An inactive system can make itself active and restart the ring. If, however, all systems are quiesced, issuing VARY GRS with RESTART invokes the reactivate function. Reactivating the ring avoids a complex wide re-IPL but can introduce data integrity exposures. Allow the reactivate function to proceed only on instructions from your system programmer.

QUIESCE or Q

You want to temporarily remove a system from the global resource serialization ring. Requesters of global resources on the quiesced system hold on to all global resources they own and hold their position in the queues for those resources they do not own. Global resource serialization suspends processing of all new requests for global resources.

PURGE or P

You want to remove a quiesced system from the global resource serialization complex.

If the system specified on the purge command is active, global resource serialization issues messages that describe the situation. Depending on your response, GRS will quiesce the system and then continue with the purge. All global resources owned by the system you purge are released and all outstanding requests for global resources made by the system you purge are deleted. Use this option when a system is no longer running and needs a re-IPL.

Note: Indiscriminate use of the VARY GRS,PURGE command can cause resource integrity problems and can put the system in a non-restartable wait state. Notify the system programmer if the system you are purging holds any resources.

Placing an I/O Path or Paths Online or Offline

Use the following form of the VARY command to place online or offline a path to a device or a range of devices. A path is the logical route between a processor and a device. A path can be offline if:

- One or more of the path components is offline
- A VARY PATH OFFLINE command has been entered

Notes:

1. Path(s) taken offline with a VARY PATH command can only be brought online again with another VARY PATH command.
2. In a JES3 environment, if a device is being used by JES3, or allocated to a job by JES3, the system will not take offline the last path to that device.

VARY Command

3. When MVS takes the last path to a device offline, the device is also taken offline. In a JES3 environment, it also takes the device offline to JES3.
4. When MVS brings the first path to a device online, the device is also brought online (if it is not in use). In a JES3 environment, the device is also brought online to JES3.
5. The VARY PATH command cannot be used on paths that are defined as managed. To manipulate managed paths, use the VARY SWITCH command. (See “Placing a Switch Port Online or Offline” on page 4-658.)

```
V PATH
{([/]devnum,chp[,/]devnum,chp]...)
}
{(([/]devnum[,/]devnum]...),chp
 [,([/]devnum[,/]devnum]...),chp]...)
}
{(([/]lowdevnum-[/]highdevnum],[/]lowdevnum-[/]highdevnum]...),chp }
 [,([/]lowdevnum-[/]highdevnum[]),chp]...)
}
{(cfname,chp[,cfname,chp]...)
}
{((cfname[,cfname]...),chp
 [,cfname[,cfname]...],chp]...)
}
{,ONLINE[,FORCE]
{,OFFLINE[,UNCOND | ,FORCE]}
```

PATH

The system is to move the specified path(s) online or offline.

[/]devnum

The device number of a device associated with the path the system is to move online or offline.

[/]lowdevnum-[/]highdevnum

The device numbers of a range of devices associated with the paths the system is to move online or offline. The lower device number of each range is *lowdevnum* and the higher device number of each range is *highdevnum*.

cfname

The name of a single coupling facility associated with the path(s) the system is to logically move online or offline. *cfname* can be up to 8 alphanumeric characters long.

chp

The channel path associated with the path(s) the system is to move online or offline. You can specify for *chp* any number from 00 to FF.

Device numbers and coupling facility names can be specified in any combination. If you specify only one device number or coupling facility name, you do not need to enter the parentheses.

A device number is 3 or 4 hexadecimal digits, optionally preceded by a slash (/). You can precede the device number with a slash to prevent ambiguity between the device number and a coupling facility name.

ONLINE

The system is to bring the path(s) online.

FORCE

The system is to bring back online the path previously taken offline. If the path was taken offline by Enterprise System Connection Manager (ESCM), then use ESCM to bring the path back online. Use the VARY PATH,ONLINE,FORCE command only if ESCM is not available and the path is physically available. If the path is quiesced by Control Unit Initiated Reconfiguration (C.U.I.R.), the VARY PATH,ONLINE,FORCE command can be used to bring the path back online.

OFFLINE

The system is to take the specified path(s) offline. The system rejects this command if the specified path is the last available path to a device that is any one of the following:

- Online (either allocated or unallocated)
- Allocated (either online or offline)
- In use by the system
- A console
- Assigned to JES3
- A coupling facility

UNCOND

The system is to take the path offline. The system rejects the VARY PATH,OFFLINE,UNCOND command if the specified path is the last available path to a device that is any one of the following:

- Allocated
- In use by the system
- A console
- Assigned to JES3
- A coupling facility, and one or more structures are in use by an active XES connection on the system offline

In other words, adding the UNCOND keyword to the VARY PATH,OFFLINE command requests that the system take offline those last paths to devices that are online but unallocated.

FORCE

The system is to take the specified path(s) offline. If Dynamic Pathing (DPS) Validation is invoked, the system will issue DISBAND/REGROUP Set Path Group ID (SPID) commands on the paths that are remaining online, instead of RESIGN SPIDs on the path that is coming offline. The system rejects this command if the specified path is the last available path to a device that is any one of the following:

- Allocated (either online or offline)
- In use by the system
- A console
- Assigned to JES3
- A coupling facility

Example

Enter the following command to take offline the paths through channel path 2 leading to devices 130, 133, 134, 135, and 140.

```
V PATH((130,133-135,140),2),OFFLINE
```

The system issues a message describing the path status for each device.

Assume that channel path 2 represents the last paths to device 134, which is online and allocated, and device 135, which is online and unallocated. A VARY

VARY Command

PATH((134,135),2),OFFLINE,UNCOND command would remove the last path to device 135 but not 134. The jobs using device 134 must end or be terminated before the last path can be removed.

Changing the state of coupling facility cache structures and volumes

Use the VARY SMS command to control SMSVSAM processing. The scope of most of these commands is sysplex wide. Some, however, such as VARY SMS,SMSVSAM,ACTIVE is not sysplex wide. For more information, see *z/OS DFSMS Storage Administration Reference*.

The syntax of the VARY SMS command is:

```
V SMS,{CFCACHE(cachename),{ENABLE|E }
      {
          {QUIESCE|Q}
      }
      {CFVOL(volid),{ENABLE|E }
      {
          {QUIESCE|Q}
      }
      {MONDS(dsname[,dsname...]),{ON|OFF}
      {
          {SHCDS(shcdsname),{NEW
          {
              {NEWSPARE}
          }
          {DELETE
      }
      {SMSVSAM,{ACTIVE
      {
          {SPHERE(spherename),{ENABLE|E}
      }
      {FALLBACK
      {
          {TERMINATESERVER
      }
      {FORCEDELETELOCKSTRUCTURE
      }
```

CFCACHE(cachename)

To change the state of a cache structure, specify the name of the cache structure (structurename).

If you specify ENABLE, VSAM RLS data can be stored in cache structure. This is the normal state of operations and the state the coupling facility cache structure is in after sysplex IPL.

If you specify QUIESCE, you cannot store any VSAM RLS data in the cache structure.

The QUIESCE operation is not complete until the state of the volume is quiesced. Use the D SMS,CFVOL to determine the state of the volume.

CFVOL(volid)

To change the state of a volume as it relates to coupling facility cache structures, specify the volume (volid).

If you specify ENABLE, data contained on this volume can be stored in a coupling facility cache structure. This is the normal state of operations.

If you specify QUIESCE, you cannot store any data from the volume on the coupling facility cache structure.

Note: If you specify QUIESCE, SMS might still select the volume during data set allocation. To stop SMS from selecting this volume, see "Changing the SMS Status of a Storage Group or Volume" on page 4-646.

MONDS(dsname{,dsname...}),ONIOFF

To specify the data set name (dsname) or data set names (dsname{,dsname...}) you want to be eligible for coupling facility statistical monitoring, specify ON.

To indicate that the specified data set is no longer eligible for statistical monitoring, specify OFF.

Monitoring is tracked through SMF record 42 subtype 16.

You can specify a full or partial data set name with at least one high level qualifier. An asterisk cannot be followed by other qualifiers. You can specify up to 16 data set names with each command.

This command affects activity for the specified data sets across all systems in the sysplex.

SHCDS

To add or delete a sharing control data set (SHCDS), specify the name of the SHCDS.

If you specify NEW, a new active SHCDS named (shcdsname) will be added.

If you specify NEWSPARE, a new spare SHCDS named (shcdsname) will be added.

If you specify DELETE, a SHCDS named (shcdsname) will be deleted. This SHCDS can be either an active or a spare SHCDS.

Note: The sharing control data set (SHCDS) is identified by the dsname SYS1.DFPSCDSC.qualified.Vvolser. When specifying its name (shcdsname) in this command, do not use the fully-qualified name. Use only qualified.Vvolser as the shcdsname, without the SYS1.DFPSCDSC prefix.

SMSVSAM

To manage SMSVSAM data sets or the SMSVSAM server, specify one of the following parameters:

ACTIVE

Restarts the SMSVSAM server and re-enables the automatic restart facility for the server. This command will not function if the SMSVSAM address space was terminated with a FALBACK command.

SPHERE

Clears the VSAM-quiesced state for the specified sphere. Normally, this operation is done under application program control. This command is required only in rare circumstances.

FALBACK

Is used as the last step in the disablement procedure to fall back from SMSVSAM processing. For the SMSVSAM fallback procedure, see *z/OS DFSMS Storage Administration Reference*.

TERMINATESERVER

Terminates an SMSVSAM server. The server will not automatically restart after the termination. After some recovery action is complete, you can restart the SMSVSAM server with the V SMS,SMSVSAM,ACTIVE command.

After you issue the V SMS,SMSVSAM,TERMINATESERVER command, SMSVSAM attempts to release all system-related locks in the lock structure IGWLOCK00 of the coupling facility. If active locks still exist when SMSVSAM disconnects from IGWLOCK00, the system issues the message IGW413I.

Use this command for specific recovery scenarios that require the SMSVSAM server to be down and not to restart automatically. This command must be entered before issuing the V XCF, *sysname*,OFFLINE

VARY Command

command, in order to terminate the SMSVSAM address space before partitioning a system from the sysplex.

The TERMINATESERVER operation might be unable to terminate the SMSVSAM address space under some abnormal conditions. In such cases, you can use the FORCE SMSVSAM,ARM command to immediately terminate the SMSVSAM address space and automatically restart the server. If you do not want a restart, use the SET SMS=xx command or SETSMS RLSINIT(NO) command to activate the RLSINIT(NO) keyword in the IGDSMSxx parmlib member before you issue the FORCE SMSVSAM,ARM command.

FORCEDELETELOCKSTRUCTURE

Deletes the lock structure from the coupling facility and deletes any data in the lock structure at the time the command is issued.

You must reply to the confirmation message with the response FORCEDELETELOCKSTRUCTURESMSVSAMYYES before the command takes effect.

Use this command only in the event of a volume loss.

Example 1

The following command tells SMS not to allow allocation of new data sets from storage group SG1 on system MVS2:

```
VARY SMS,STORGRP(SG1,MVS2),DISABLE,N
```

Tip: This command works only if the specified system is defined explicitly to SMS. If the system is defined to SMS as part of a system group, the command fails.

Example 2

The following command tells SMS to allow allocation of both new and old data sets from storage group SG1 on all MVS systems:

```
VARY SMS,STORGRP(SG1,ALL),ENABLE
```

Tip: This command works only if the specified system is defined explicitly to SMS. If the system is defined to SMS as part of a system group, the command fails.

Example 3

The following command tells SMS to allow allocation of both new and old data sets from volume SMS001 on system MVS3, enter:

```
VARY SMS,VOLUME(SMS001,MVS3),ENABLE
```

Example 4

The following command tells a JES3 system to prevent scheduling a job that requires volume SMS001 on MVS3 and after a job is scheduled, tells SMS to select volume SMS001 on MVS3 for a new data set only if there are no other choices:

```
VARY SMS,VOLUME(SMS001,MVS3),QUIESCE
```

Tip: This command works only if the specified system is defined explicitly to SMS. If the system is defined to SMS as part of a system group, the command fails.

Placing an Optical Drive or Library Online or Offline

Use the VARY SMS command to vary optical drives and optical libraries online or offline.

For a detailed description of the VARY SMS,DRIVE command parameters, refer to the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

Placing a System-Managed Tape Library Online or Offline

Use the VARY SMS,LIBRARY command to place a system-managed tape library online or offline.

For a detailed description of the VARY SMS,LIBRARY command parameters, refer to the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*, and the *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Analyzing the State of the PDSE Subsystem

Use the VARY SMS,PDSE,ANALYSIS command to determine the state of the PDSE (partitioned data set extended) subsystem. You can run the analysis on all the PDSEs that are open, or you can specify a specific PDSE by dsname and optionally the volser.

Consult *z/OS DFSMSdfp Diagnosis* for specific information about how to use this command, including the command syntax.

Releasing PDSE Latches

Use the VARY SMS,PDSE,FREELATCH command to release a latch that the V SMS,PDSE,ANALYSIS command has determined is frozen.

Consult *z/OS DFSMSdfp Diagnosis* for specific information about how to use this command, including the command syntax.

Modifying processing of PDSE monitor

Use the following form of the VARY command to specify how the processing for the PDSE monitor should be modified:

```
V SMS,PDSE,MONITOR,ON[,interval[,duration]]
      OFF
      RESTART
```

ON | OFF

Turns monitor processing on or off.

RESTART

Resets the state at which the monitor, due to an unexpected error in its processing, is shut down.

interval

The number of seconds between successive scans of the monitor.

duration

The number of seconds an possible error condition must exist before it is treated as an error.

VARY Command

Note: The default values for interval and duration are not necessarily the optimum values for your system. Observation and adjustment will probably be necessary to attain the right values. Over time, even your chosen values might need to be re-adjusted.

Display current state of the PDSE monitor

Use the VARY SMS,PDSE,MONITOR command to display the current status of the PDSE monitor. See message IGW043I for details on the information displayed.

Changing the SMS Status of a Storage Group or Volume

Use the VARY SMS command to change the status of a storage group or volume that the storage management subsystem (SMS) controls. The command allows data sets in the storage group or volume to be allocated/accessed, or not allocated/accessed, to jobs. If the system you issue the command from is defined as part of a system group, but you did not specify the system or system group parameter, then the default is the current system and the command fails.

The possible states of a volume or storage group are:

ENABLE

Allows the system or system group to allocate and access data sets.

DISABLE

Prevents the system or system group from allocating or accessing existing data sets.

DISABLE,NEW

Prevents the system or system group from allocating new data sets; existing data sets can be accessed.

Note: For DISABLE and DISABLE,NEW, jobs needing the denied data sets can potentially fail during execution.

QUIESCE

Prevents JES3 from scheduling a job that creates new data sets or accesses existing data sets.

Once a job has been scheduled on either JES2 or JES3, SMS only selects the volume or storage group for a new data set if there are no other choices. There is no effect on the allocating or accessing of existing data sets.

Note: The QUIESCE state is not valid for object or object backup storage groups.

QUIESCE,NEW

Prevents JES3 from scheduling jobs that create new data sets or modify (DISP=MOD) existing data sets.

Once the job has been scheduled on either JES2 or JES3, SMS only selects the volume or storage group for a new data set if there are no other choices. There is no effect on the allocating or accessing of existing data sets.

Note: The QUIESCE state is not valid for object or object backup storage groups.

The syntax of the VARY SMS command is:

```
V SMS,{{STORGRP|SG}(storgrp,[*|ALL|system[,system]...]),{QUIESCE|Q}[,NEW|,N]
      {
        {{VOLUME|VOL}(volume,[*|ALL|system[,system]...]) } {DISABLE|D}[,NEW|,N]
```

STORGRP or SG(storgrp[,system,...])

Identifies the storage group and, optionally, the system or system group that the SMS status change is to affect. If you omit *system* or specify an *, the command affects only the system on which you issue the command. If the system you issue the command from is defined to SMS as part of a system group, then the command fails.

If you specify ALL, the command affects all systems and system groups in the complex. With ALL, the command is effective directly on the issuing system, and the sharing systems in the SMSPLEX see the updates in the COMMDS at the regular interval processing. Rapid use of several commands might overlap in updates of the COMMDS, increasing delay in the propagation through the sharing systems active configuration due to serialization timing.

To specify a storage group named “ALL”, you must enclose the name in parentheses ((ALL)) to distinguish it from *all* storage groups.

VOLUME or VOL(volume[,system,...])

Identifies the volume and, optionally, the system or system group that the SMS status change is to affect. If you omit *system* or specify an *, the command affects only the system on which you issue the command. If the system you issue the command from is defined to SMS as part of a system group, then the command fails.

If you specify ALL, the command affects all systems and system groups in the complex. With ALL, the command is effective directly on the issuing system, and the sharing systems in the SMSPLEX see the update in the COMMDS at the regular interval processing. Rapid use of several commands might overlap in updates of the COMMDS, increasing delay in the propagation through the sharing systems active configuration due to serialization timing.

To specify a volume named “ALL”, you must enclose the name in parentheses ((ALL)) to distinguish it from *all* volumes.

ENABLE or E

SMS is to permit allocation of new and old data sets from the specified storage group or volume on the designated system(s) or system group(s).

QUIESCE or Q[,NEW or ,N]

For QUIESCE, a JES3 system prevents the scheduling of jobs that create new data sets or accessing existing data sets from the specified storage group or volume. For QUIESCE,NEW a JES3 system prevents the scheduling of jobs that create new data sets or modify (DISP=MOD) existing data sets from the specified storage group or volume.

Once the job has been scheduled on JES2 or JES3, SMS only selects the volume or storage group for a new data set if there are no other choices. There is no effect on the allocating or accessing of existing data sets.

DISABLE or D[,NEW or ,N]

For DISABLE, SMS is *not* to allow allocation or accessing of existing data sets in the specified storage group or volume.

VARY Command

For DISABLE,NEW, SMS is not to select the volume or storage group for a new data set.

Note: For DISABLE and DISABLE,NEW, jobs needing the denied data sets can potentially fail during execution. You should specify these parameters *only* under the direction of your storage administrator.

Example 1

To tell SMS not to allow allocation of new data sets from storage group SG1 on system MVS2, enter:

```
VARY SMS,STORGRP(SG1,MVS2),DISABLE,N
```

Note: This command works only if the specified system is defined explicitly to SMS. If the system is defined to SMS as part of a system group, the command fails.

Example 2

To tell SMS to allow allocation of both new and old data sets from storage group SG1 on all MVS systems, enter:

```
VARY SMS,STORGRP(SG1,ALL),ENABLE
```

Note: This command works only if the specified system is defined explicitly to SMS. If the system is defined to SMS as part of a system group, the command fails.

Example 3

To tell SMS to allow allocation of both new and old data sets from volume SMS001 on system MVS3, enter:

```
VARY SMS,VOLUME(SMS001,MVS3),ENABLE
```

Example 4

To (1) tell a JES3 system to prevent scheduling a job required volume SMS001 on MVS3, and (2) tell SMS, once a job is scheduled, to only select volume SMS001 on MVS3 for a new data set if there are no other choices, enter:

```
VARY SMS,VOLUME(SMS001,MVS3),QUIESCE
```

Note: This command works only if the specified system is defined explicitly to SMS. If the system is defined to SMS as part of a system group, the command fails.

Controlling DFSMStvs processing

Use the VARY SMS command to control DFSMS Transactional VSAM Services (DFSMStvs) processing. For information about DFSMStvs, see *z/OS DFSMStvs Planning and Operating Guide* and *z/OS DFSMStvs Administration Guide*.

You can use the VARY SMS command to change the status for DFSMStvs in these ways:

- Change the state of a DFSMStvs instance or of all DFSMStvs instances in the sysplex
- Change the state of a log stream to which DFSMStvs has access

- Change the state of a data set for VSAM record-level sharing (RLS) and DFSMStvs access
- Start or stop peer recovery processing for a DFSMStvs instance

Restriction: You cannot use the VARY SMS command to change the state of a DFSMStvs instance while it is initializing. Any attempt to do so is suspended until the initialization completes.

The possible states of a DFSMStvs instance follow:

ENABLE

Enables DFSMStvs to begin accepting new units of recovery for processing.

DISABLE

Prevents DFSMStvs from processing new work requests. DFSMStvs does not process new work requests from units of recovery that are currently in progress.

QUIESCE

Prevents DFSMStvs from accepting any new units of recovery for processing. DFSMStvs completes the processing of any units of recovery in progress.

The possible states of a data set follow:

ENABLE

Unquiesces a data set for VSAM RLS and DFSMStvs access.

QUIESCE

Quiesces a data set for VSAM RLS and DFSMStvs access.

The syntax of the VARY SMS command for DFSMStvs follows.

```
V SMS,{TRANVSAM({tvname|ALL}){,{QUIESCE|Q}}}
      {,{ENABLE|E}}
      {,{DISABLE|D}}
      {LOG(logstreamid){,{QUIESCE|Q}}}
      {,{ENABLE|E}}
      {,{DISABLE|D}}
      {SMSVSAM,SPHERE(sphere){,{QUIESCE|Q}}}
      {,{ENABLE|E}}
      {TRANVSAM(tvname),PEERRECOVERY{,{ACTIVE|A}}}
      {,{ACTIVEFORCE}}
      {,{INACTIVE|I}}}
```

TRANVSAM({ tvname or ALL})

Enables, quiesces, or disables the specified DFSMStvs instance or all DFSMStvs instances in the sysplex. The command is routed to all systems in the sysplex and affects either all DFSMStvs instances or the DFSMStvs instance with the specified name.

QUIESCE or Q

DFSMStvs completes processing of any units of recovery that are in progress but does not accept any new ones. DFSMStvs completes its quiesce processing when the last data set that is open for DFSMStvs access is closed. Then DFSMStvs is unavailable until a VARY SMS,TRANVSAM,ENABLE command is issued.

VARY Command

DISABLE or D

DFSMStvs immediately stops processing new work requests, including units of recovery that are currently in progress. When the last data set that is open for DFSMStvs access is closed, DFSMStvs retains locks, unregisters with RRS, and is unavailable until a VARY SMS,TRANVSAM,ENABLE command is issued. No further DFSMStvs requests can complete until DFSMStvs is enabled. However, commit and backout requests that were already in progress at the time the disable command was received could be successful.

ENABLE or E

DFSMStvs begins accepting new units of recovery for processing.

LOG(*logstreamid*)

Enables, quiesces, or disables DFSMStvs access to the specified log stream (*logstreamid*). Quiescing the DFSMStvs undo or shunt log stream is equivalent to quiescing DFSMStvs processing.

Disabling the DFSMStvs undo or shunt log stream is equivalent to disabling DFSMStvs. Although the two log streams are physically separate, they are treated as a single entity by DFSMStvs logging services.

Quiescing or disabling the log of logs has no effect on DFSMStvs processing because records are written to the log of logs only as an optimization for forward recovery products. However, disabling the log of logs can cause a mismatch of tie-up records written at data set OPEN with file-close records.

Quiescing a forward recovery log stream will cause a quiesce of processing for any data sets that use that log stream. Disabling a forward recovery log stream causes all processing that attempts to use that log stream to fail. DFSMStvs will be unable to commit or back out any units of recovery that were using the log stream because it will be unable to write the necessary records to the log stream.

QUIESCE or Q

DFSMStvs completes the processing of any in-progress units of recovery using the log stream but does not accept any new ones that would require the log stream, with the exception of the log of logs. If the log is a DFSMStvs system log (undo or shunt log), it becomes quiesced when all units of recovery that are using DFSMStvs complete and any open data sets are closed. If the log is a forward recovery log, it becomes quiesced when the last data set that is open for output in DFSMStvs mode is closed. If the log is a log of logs, it becomes quiesced when the last forward-recoverable data set that is open for output in DFSMStvs mode, for which a tie-up record was written to the log of logs, is closed. New work can start, but DFSMStvs does not write tie-up records or file-close records to the log of logs.

DISABLE or D

DFSMStvs immediately stops using the log stream. This can prevent completion of commit or backout for units of recovery. Those units of recovery are shunted, as long as shunting them does not require reading or writing the now disabled log.

Recommendation: Do not use this command without first quiescing the log stream unless the log stream is damaged or errors occur that cannot be corrected.

If the log is a DFSMStvs system log (undo or shunt log), DFSMStvs does not allow any further work to be done. All OPENS and VSAM record

management requests are failed. The log becomes disabled when all units of recovery that are using DFSMStvs complete. DFSMStvs then retains locks, unregisters with RRS and the lock manager, and is unavailable until the log is enabled. No further DFSMStvs requests can complete until the system log is made available, including commit and backout requests.

If the log is a forward recovery log, any new OPENs that require the use of the log are failed. The log will become disabled when the last data set that uses it, and is OPEN for output in DFSMStvs mode, is closed.

If the log is a log of logs, it will become disabled when the last forward recoverable data set that is open for output in DFSMStvs mode, for which a tie-up record is written to the log of logs, is closed. New work can start, but DFSMStvs does not write tie-up records or file-close records to the log of logs.

ENABLE or E

DFSMStvs begins accepting new units of recovery that use the log stream for processing. If DFSMStvs work was left incomplete when DFSMStvs processing was stopped, DFSMStvs completes that work as part of its restart processing.

SMSVSAM,SPHERE(*sphere*)

Quiesces or unquiesces the data set *sphere* for VSAM RLS and DFSMStvs access. Use this command to ensure that users do not access the data set while it is being recovered. A data set can be quiesced to allow it to be accessed in a mode other than VSAM RLS or DFSMStvs. Before attempting to quiesce a data set, ensure that all jobs that were accessing the data set using DFSMStvs are either finished or canceled.

When you specify data sets for a VARY SMS,SMSVSAM,SPHERE command, they are not necessarily quiesced in the order in which you specified them or in any other order. The same is true for asterisk notation (*); the data sets are not necessarily quiesced alphabetically or in any other order. So, when the last data set you specified is quiesced, you cannot assume that the other data sets you specified have been quiesced.

TRANVSAM(*tvsname*),PEERRECOVERY

Starts or stops peer recovery processing for a failed instance of DFSMStvs. This command applies only to the system on which it is issued. That system is responsible for performing all peer recovery processing for the failed DFSMStvs instance.

ACTIVE or A

This system should begin peer recovery processing on behalf of the specified failed instance of DFSMStvs. If the failed instance of DFSMStvs was not disabling or disabled due to an operator command, the system will perform the necessary initialization and then start tasks to perform any work that was left incomplete by a system failure. Because a large amount of work could be outstanding, the system will start tasks in groups of ten and then begin more work as those tasks complete. Controlling the amount of work in progress at any given time allows a quiesce of peer recovery processing by varying it INACTIVE, in the event that the failed system comes back up.

ACTIVEFORCE

The system begins peer recovery processing on behalf of the specified failed instance of DFSMStvs, regardless of the failed instance's status.

VARY Command

INACTIVE or I

This system should stop processing peer recovery work on behalf of the specified instance of DFSMStvs. This command does not take affect immediately. Instead, peer recovery processing that is already in progress is allowed to complete before peer recovery processing stops.

Example 1

This example show how to vary a DFSMStvs instance and quiesce it. At the time this command was issued, no jobs were using DFSMStvs services, nor were there any active DFSMStvs opens. As a result, DFSMStvs was able to transition from quiescing to quiesced.

```
V SMS,TRANVSAM(001),Q
```

This will result in:

```
V SMS,TRANVSAM(001),Q
IGW471I DFSMS VSAM RLS REQUEST TO QUIESCE 834
TRANSACTIONAL VSAM INSTANCE IGWTVO01 IS ACCEPTED
QUIESCE REASON: VARY SMS TRANSACTIONAL VSAM OPERATOR COMMAND
IGW471I DFSMS VSAM RLS COMMAND PROCESSOR 835
ON SYSTEM: SYSTEM1
IS WAITING FOR A RESPONSE
    FROM TRANSACTIONAL VSAM: IGWTVO01
COMMAND REQUESTED:

    QUIESCE TRANSACTIONAL VSAM: IGWTVO01
    IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 836
    ON SYSTEM: SYSTEM1
    IS WAITING FOR A RESPONSE FROM TRANSACTIONAL VSAM: IGWTVO01
    COMMAND REQUESTED: DISCONNECT FROM LOGSTREAM: IGWTVS1.LOG.OF.LOGS
    IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 837
    ON SYSTEM: SYSTEM1
    IS WAITING FOR A RESPONSE FROM TRANSACTIONAL VSAM: IGWTVO01
    COMMAND REQUESTED: QUIESCE LOGSTREAM: IGWTVO01.IGWLOG.SYSLOG
    IGW474I DFSMS VSAM RLS IS DISCONNECTING FROM 838
    TRANSACTIONAL VSAM LOGSTREAM IGWTVO01.IGWLOG.SYSLOG
    SYSTEM NAME: SYSTEM1
    TRANSACTIONAL VSAM INSTANCE NAME: IGWTVO01
    IGW474I DFSMS VSAM RLS IS DISCONNECTING FROM 839
    TRANSACTIONAL VSAM LOGSTREAM IGWTVO01.IGWSHUNT.SHUNTLG
    SYSTEM NAME: SYSTEM1
    TRANSACTIONAL VSAM INSTANCE NAME: IGWTVO01
    IGW474I DFSMS VSAM RLS IS DISCONNECTING FROM 840
    TRANSACTIONAL VSAM LOGSTREAM IGWTVS1.LOG.OF.LOGS
    SYSTEM NAME: SYSTEM1
    TRANSACTIONAL VSAM INSTANCE NAME: IGWTVO01
    IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 841
    ON SYSTEM: SYSTEM1
    HAS BEEN POSTED BY TRANSACTIONAL VSAM: IGWTVO01
    COMMAND REQUESTED: QUIESCE LOGSTREAM: IGWTVO01.IGWLOG.SYSLOG
    IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 842
    ON SYSTEM: SYSTEM1
    HAS BEEN POSTED BY TRANSACTIONAL VSAM: IGWTVO01
    COMMAND REQUESTED: QUIESCE LOGSTREAM: IGWTVO01.IGWSHUNT.SHUNTLG
    IGW471I DFSMS VSAM RLS COMMAND PROCESSOR 843
    ON SYSTEM: SYSTEM1
    HAS BEEN POSTED BY TRANSACTIONAL VSAM: IGWTVO01
    COMMAND REQUESTED:

        QUIESCE TRANSACTIONAL VSAM: IGWTVO01
        IGW471I DFSMS VSAM RLS COMMAND PROCESSOR 844
```

ON SYSTEM: SYSTEM1
 HAS CALLED THE DFSMS COMMAND COMPLETE PROCESSOR
 COMMAND REQUESTED:

QUIESCE TRANSACTIONAL VSAM: IGWTVO01
 IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 845
 ON SYSTEM: SYSTEM1
 IS WAITING FOR A RESPONSE FROM TRANSACTIONAL VSAM: IGWTVO01
 COMMAND REQUESTED: QUIESCE LOGSTREAM: IGWTVO01.IGWSHUNT.SHUNTLOG
 IGW471I DFSMS VSAM RLS REQUEST TO QUIESCE 846
 TRANSACTIONAL VSAM INSTANCE IGWTVO01 COMPLETED.

TRANSACTIONAL VSAM INSTANCE IGWTVO01 IS NOW QUIESCED.

TRANSACTIONAL VSAM LOGSTREAM IGWTVO01.IGWLOG.SYSLOG IS NOW QUIESCED
 HAS BEEN POSTED BY TRANSACTIONAL VSAM: IGWTVO01
 COMMAND REQUESTED:

QUIESCE TRANSACTIONAL VSAM: IGWTVO01
 IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 844
 ON SYSTEM: SYSTEM1
 HAS CALLED THE DFSMS COMMAND COMPLETE PROCESSOR
 COMMAND REQUESTED:

QUIESCE TRANSACTIONAL VSAM: IGWTVO01
 IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 845
 ON SYSTEM: SYSTEM1
 IS WAITING FOR A RESPONSE FROM TRANSACTIONAL VSAM: IGWTVO01
 COMMAND REQUESTED: QUIESCE LOGSTREAM: IGWTVO01.IGWSHUNT.SHUNTLOG
 IGW471I DFSMS VSAM RLS REQUEST TO QUIESCE 846
 TRANSACTIONAL VSAM INSTANCE IGWTVO01 COMPLETED.

TRANSACTIONAL VSAM INSTANCE IGWTVO01 IS NOW QUIESCED.

TRANSACTIONAL VSAM LOGSTREAM IGWTVO01.IGWLOG.SYSLOG IS NOW QUIESCED

Example 2

This example shows how to vary a DFSMStvs instance disabled. At the time this command was issued, no jobs were using DFSMStvs services, nor were there any active DFSMStvs opens. As a result, DFSMStvs was able to transition from disabling to disabled.

This will result in:

V SMS,TRANVSAM(002),D
 IGW471I DFSMS VSAM RLS REQUEST TO DISABLE 849
 TRANSACTIONAL VSAM INSTANCE IGWTVO02 IS ACCEPTED.

DISABLE REASON: VARY SMS TRANSACTIONAL VSAM OPERATOR COMMAND
 IGW471I DFSMS VSAM RLS REQUEST TO DISABLE 850
 TRANSACTIONAL VSAM INSTANCE IGWTVO02 IS COMPLETED.

TRANSACTIONAL VSAM INSTANCE IGWTVO02 IS NOW DISABLED.

TRANSACTIONAL VSAM LOGSTREAM IGWTVO02.IGWLOG.SYSLOG IS NOW DISABLED
 TRANSACTIONAL VSAM LOGSTREAM IGWTVO02.IGWSHUNT.SHUNTLOG IS NOW DISABLED

Example 3

VARY Command

This example shows how to vary all DFSMStvs instances and enable them. At the time this command was issued, the two DFSMStvs instances in the sysplex were quiesced and disabled, respectively.

This will result in:

```
V SMS,TRANVSAM(ALL),E
IGW472I DFSMS VSAM RLS REQUEST TO ENABLE 853
TRANSACTIONAL VSAM INSTANCE IGWTVO01 ACCEPTED.

ENABLE REASON: VARY SMS TRANSACTIONAL VSAM OPERATOR COMMAND
IGW472I DFSMS VSAM RLS REQUEST TO ENABLE 854
TRANSACTIONAL VSAM INSTANCE IGWTVO02 ACCEPTED.

ENABLE REASON: VARY SMS TRANSACTIONAL VSAM OPERATOR COMMAND
IGW471I DFSMS VSAM RLS COMMAND PROCESSOR 855
ON SYSTEM: SYSTEM1
IS WAITING FOR A RESPONSE
FROM TRANSACTIONAL VSAM: IGWTVO01
COMMAND REQUESTED:

ENABLE TRANSACTIONAL VSAM: IGWTVO01
IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 856
ON SYSTEM: SYSTEM1
IS WAITING FOR A RESPONSE FROM TRANSACTIONAL VSAM: IGWTVO01
COMMAND REQUESTED: ENABLE LOGSTREAM: IGWTVO01.IGWLOG.SYSLOG
IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 857
ON SYSTEM: SYSTEM1
IS WAITING FOR A RESPONSE FROM TRANSACTIONAL VSAM: IGWTVO01
COMMAND REQUESTED: ENABLE LOGSTREAM: IGWTVO01.IGWSHUNT.SHUNTLG
IGW860I TRANSACTIONAL VSAM HAS SUCCESSFULLY REGISTERED WITH RLS
IGW848I 10312000 13.38.51 SYSTEM UNDO LOG IGWTVO01.IGWLOG.SYSLOG 859
INITIALIZATION HAS STARTED
IXL014I IXLCONN REQUEST FOR STRUCTURE TVS_LOG001 860
WAS SUCCESSFUL. JOBNOME: IXGLOGR ASID: 0015
CONNECTOR NAME: IXGLOGR_SYSTEM1 CFNAME: FACIL02
IXL015I STRUCTURE ALLOCATION INFORMATION FOR 861
STRUCTURE TVS_LOG001, CONNECTOR NAME IXGLOGR_SYSTEM1
CFNAME ALLOCATION STATUS/FAILURE REASON
-----
FACIL02 STRUCTURE ALLOCATED
FACIL01 PREFERRED CF ALREADY SELECTED
TRANSACTIONAL VSAM LOGSTREAM IGWTVO01.IGWLOG.SYSLOG
SYSTEM NAME: SYSTEM1
TRANSACTIONAL VSAM INSTANCE NAME: IGWTVO01
IGW848I 10312000 13.39.21 SYSTEM UNDO LOG IGWTVO01.IGWLOG.SYSLOG 869
INITIALIZATION HAS ENDED
IGW848I 10312000 13.39.21 SYSTEM SHUNT LOG IGWTVO01.IGWSHUNT.SHUNTLG
INITIALIZATION HAS STARTED
IGW474I DFSMS VSAM RLS IS CONNECTING TO 877
TRANSACTIONAL VSAM LOGSTREAM IGWTVO01.IGWSHUNT.SHUNTLG
SYSTEM NAME: SYSTEM1
TRANSACTIONAL VSAM INSTANCE NAME: IGWTVO01
IGW848I 10312000 13.39.50 SYSTEM SHUNT LOG IGWTVO01.IGWSHUNT.SHUNTLG
INITIALIZATION HAS ENDED
IGW848I 10312000 13.39.50 LOG OF LOGS IGWTVS1.LOG.OF.LOGS 879
INITIALIZATION HAS STARTED
IGW474I DFSMS VSAM RLS IS CONNECTING TO 886
TRANSACTIONAL VSAM LOGSTREAM IGWTVS1.LOG.OF.LOGS
SYSTEM NAME: SYSTEM1
TRANSACTIONAL VSAM INSTANCE NAME: IGWTVO01
IGW848I 10312000 13.40.18 LOG OF LOGS IGWTVS1.LOG.OF.LOGS 887
INITIALIZATION HAS ENDED
IGW865I TRANSACTIONAL VSAM INITIALIZATION IS COMPLETE.
```

```

IGW471I DFSMS VSAM RLS COMMAND PROCESSOR 897
ON SYSTEM: SYSTEM1
HAS BEEN POSTED BY TRANSACTIONAL VSAM: IGWTVO01
COMMAND REQUESTED:

    ENABLE TRANSACTIONAL VSAM: IGWTVO01
IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 898
ON SYSTEM: SYSTEM1
HAS BEEN POSTED BY TRANSACTIONAL VSAM: IGWTVO01
COMMAND REQUESTED: ENABLE LOGSTREAM: IGWTVO01.IGWLOG.SYSLOG
IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 899
ON SYSTEM: SYSTEM1
HAS BEEN POSTED BY TRANSACTIONAL VSAM: IGWTVO01
COMMAND REQUESTED: ENABLE LOGSTREAM: IGWTVO01.IGWSHUNT.SHUNTLOG
IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 900
ON SYSTEM: SYSTEM1
HAS CALLED THE DFSMS COMMAND COMPLETE PROCESSOR
COMMAND REQUESTED: ENABLE LOGSTREAM: IGWTVO01.IGWSHUNT.SHUNTLOG
IGW886I 0 RESTART TASKS WILL BE PROCESSED DURING TRANSACTIONAL VSAM
RESTART PROCESSING
IGW472I DFSMS VSAM RLS REQUEST TO ENABLE 902
TRANSACTIONAL VSAM INSTANCE IGWTVO01 IS COMPLETED.

TRANSACTIONAL VSAM INSTANCE IGWTVO01 IS NOW ENABLED.

TRANSACTIONAL VSAM LOGSTREAM IGWTVO01.IGWLOG.SYSLOG IS NOW ENABLED.

TRANSACTIONAL VSAM LOGSTREAM IGWTVO01.IGWSHUNT.SHUNTLOG IS NOW ENABLED.

TRANSACTIONAL VSAM IGWTVO01 WILL NOW ACCEPT NEW WORK
IGW866I TRANSACTIONAL VSAM RESTART PROCESSING IS COMPLETE.

IGW467I DFSMS TVSNAME PARMLIB VALUE SET DURING 904
SMSVSAM ADDRESS SPACE INITIALIZATION ON SYSTEM: SYSTEM1
TVSNAME: IGWTVO01
CURRENT VALUE: ENA-ED 1
IGW467I DFSMS TVS UNDO LOG PARMLIB VALUE SET DURING 905
SMSVSAM ADDRESS SPACE INITIALIZATION ON SYSTEM: SYSTEM1
UNDO LOGSTREAM NAME: IGWTVO01.IGWLOG.SYSLOG
CURRENT VALUE: ENA-ED 1
IGW467I DFSMS TVS SHUNT LOG PARMLIB VALUE SET DURING 906
SMSVSAM ADDRESS SPACE INITIALIZATION ON SYSTEM: SYSTEM1
SHUNT LOGSTREAM NAME: IGWTVO01.IGWSHUNT.SHUNTLOG
CURRENT VALUE: ENA-ED 1
IGW472I DFSMS VSAM RLS REQUEST TO ENABLE 907
TRANSACTIONAL VSAM INSTANCE IGWTVO02 IS COMPLETED.

TRANSACTIONAL VSAM INSTANCE IGWTVO02 IS NOW ENABLED.

TRANSACTIONAL VSAM LOGSTREAM IGWTVO02.IGWLOG.SYSLOG IS NOW ENABLED.

TRANSACTIONAL VSAM LOGSTREAM IGWTVO02.IGWSHUNT.SHUNTLOG IS NOW ENABLED.

TRANSACTIONAL VSAM IGWTVO02 WILL NOW ACCEPT NEW WORK

```

Example 4

This example shows how to vary a log stream quiesced. At the time this command was issued, no jobs were using the log stream, nor were there any active DFSMStvs opens. As a result, DFSMStvs was able to transition the log stream from quiescing to quiesced.

VARY Command

This results in the following output:

```
V SMS,LOG(IGWTVS.FR.LOG001),Q
IGW473I DFSMS VSAM RLS REQUEST TO QUIESCE 910
TRANSACTIONAL VSAM LOGSTREAM IGWTVS.FR.LOG001      IS ACCEPTED
QUIESCE REASON: VARY SMS TRANSACTIONAL VSAM OPERATOR COMMAND
IGW473I DFSMS VSAM RLS REQUEST TO QUIESCE 911
TRANSACTIONAL VSAM LOGSTREAM IGWTVS.FR.LOG001      COMPLETED.

TRANSACTIONAL VSAM LOGSTREAM IGWTVS.FR.LOG001      IS NOW QUIESCED
```

If any data sets were open in DFMSMStvs mode, transitioning the log stream from quiescing to quiesced could take much longer. DFMSMStvs would need to wait until the data sets were closed.

Example 5

This example shows how to vary a log stream disabled. At the time this command was issued, no jobs were using the log stream, nor were there any active DFMSMStvs opens. As a result, DFMSMStvs was able to transition the log stream from disabling to disabled.

This will result in:

```
V SMS,LOG(IGWTVS1.LOG.OF.LOGS),D
IGW473I DFSMS VSAM RLS REQUEST TO DISABLE 917
TRANSACTIONAL VSAM LOGSTREAM IGWTVS1.LOG.OF.LOGS      IS ACCEPTED
IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 918
ON SYSTEM: SYSTEM1
IS WAITING FOR A RESPONSE FROM TRANSACTIONAL VSAM: IGWTV001
COMMAND REQUESTED: DISABLE LOGSTREAM: IGWTVS1.LOG.OF.LOGS
IGW474I DFSMS VSAM RLS IS DISCONNECTING FROM 919
TRANSACTIONAL VSAM LOGSTREAM IGWTVS1.LOG.OF.LOGS
SYSTEM NAME: SYSTEM1
TRANSACTIONAL VSAM INSTANCE NAME: IGWTV001
IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 920
ON SYSTEM: SYSTEM1
HAS BEEN POSTED BY TRANSACTIONAL VSAM: IGWTV001
COMMAND REQUESTED: DISABLE LOGSTREAM: IGWTVS1.LOG.OF.LOGS
IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 921
ON SYSTEM: SYSTEM1
HAS CALLED THE DFSMS COMMAND COMPLETE PROCESSOR
COMMAND REQUESTED: DISABLE LOGSTREAM: IGWTVS1.LOG.OF.LOGS
IGW473I DFSMS VSAM RLS REQUEST TO DISABLE 922
TRANSACTIONAL VSAM LOGSTREAM IGWTVS1.LOG.OF.LOGS      IS COMPLETED.

TRANSACTIONAL VSAM LOGSTREAM IGWTVS1.LOG.OF.LOGS      IS NOW DISABLED
```

Example 6

This example shows how to vary a log stream enabled.

```
V SMS,LOG(IGWTVS1.LOG.OF.LOGS),E
IGW474I DFSMS VSAM RLS REQUEST TO ENABLE 929
TRANSACTIONAL VSAM LOGSTREAM IGWTVS1.LOG.OF.LOGS      IS ACCEPTED
IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 930
ON SYSTEM: SYSTEM1
IS WAITING FOR A RESPONSE FROM TRANSACTIONAL VSAM: IGWTV001
COMMAND REQUESTED: ENABLE LOGSTREAM: IGWTVS1.LOG.OF.LOGS
IGW848I 10312000 13.44.20 LOG OF LOGS IGWTVS1.LOG.OF.LOGS 931
INITIALIZATION HAS STARTED
IGW474I DFSMS VSAM RLS IS CONNECTING TO 938
TRANSACTIONAL VSAM LOGSTREAM IGWTVS1.LOG.OF.LOGS
```

```

SYSTEM NAME:           SYSTEM1
TRANSACTIONAL VSAM INSTANCE NAME:  IGWTV001
IGW848I 10312000 13.44.48 LOG OF LOGS IGWTVS1.LOG.OF.LOGS 939
INITIALIZATION HAS ENDED
IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 940
ON SYSTEM:  SYSTEM1
HAS BEEN POSTED BY  TRANSACTIONAL VSAM:  IGWTV001
COMMAND REQUESTED:  ENABLE LOGSTREAM:  IGWTVS1.LOG.OF.LOGS
IGW473I DFSMS VSAM RLS COMMAND PROCESSOR 941
ON SYSTEM:  SYSTEM1
HAS CALLED THE DFSMS COMMAND COMPLETE PROCESSOR
COMMAND REQUESTED:  ENABLE LOGSTREAM:  IGWTVS1.LOG.OF.LOGS
IGW474I DFSMS VSAM RLS REQUEST TO ENABLE 942
TRANSACTIONAL VSAM LOGSTREAM IGWTVS1.LOG.OF.LOGS      IS COMPLETED.

TRANSACTIONAL VSAM LOGSTREAM IGWTVS1.LOG.OF.LOGS      IS NOW ENABLED

```

Example 7

This example shows how to vary a VSAM sphere quiesced. At the time this command was issued, no jobs were using the data set, nor were there any active opens for the data set. As a result, DFSMStvs was able to transition the data set from quiescing to quiesced.

```

V SMS,SMSVSAM,SPHERE(SYSPLEX.SHCD.SUND01.KSDS01),Q
IGW522I SMSVSAM QUIESCE REQUEST FOR SPHERE 945
SYSPLEX.SHCD.SUND01.KSDS01 IS COMPLETED.

```

If any jobs were using the data set while it is open in DFSMStvs mode, transitioning the data set from quiescing to quiesced could take much longer. DFSMStvs would need to wait until the jobs were complete.

Example 8

This example shows how to vary a VSAM sphere enabled.

```

V SMS,SMSVSAM,SPHERE(SYSPLEX.SHCD.SUND01.KSDS01),E
IGW522I SMSVSAM ENABLE REQUEST FOR SPHERE 948
SYSPLEX.SHCD.SUND01.KSDS01 IS COMPLETED.

```

Controlling CICSVR processing

Use the VARY SMS command to control CICS VSAM Recovery services (CICSVR) processing. For information about CICSVR, see *CICSVR V3R1 Implementation Guide* and *CICSVR V3R1 User's Guide and Reference*.

You can use the VARY SMS command to:

- Control the CICSVR address space startup and shutdown.

- Add or delete a Recovery Control Data Set (RCDS) in the CICSVR address space.

The syntax of the VARY SMS command to control CICSVR startup and shutdown follows.

V SMS,CICSVR{,ACTIVE ,TERMINATESERVER}

ACTIVE

The command creates or activates the CICSVR address space and allows work

VARY Command

to be started. This command only executes on the system where the command was entered or on the target system as a result of the MVS ROUTE command.

TERMINATESERVER

The command terminates the CICSVR address space on the system where the command was entered or on the target system as a result of the MVS ROUTE command.

The syntax of the VARY SMS command to add or delete the RCDS follows.

```
V SMS,CICSVR,RCDS(rcds_name),{ADD|DELETE}
```

rcds_name

The name of the RCDS, which follows the DDNAME syntax rules.

ADD

The command adds the RCDS specified to the CICSVR address space.

DELETE

The command deletes the RCDS specified from the CICSVR address space.

Placing a Switch Port Online or Offline

Use the VARY SWITCH(ssss,pp[-pp][,pp[-pp]]...,{DCM=OFFLINE[,UNCOND]!ONLINE}) command to place a switch port online or offline to dynamic channel path management. Invoking this command for a switch port will also cause the specific managed device paths to be varied online or offline. An offline request will cause the managed channel paths to be removed from the control units connected to the managed CHPIDs at the specified ports. The VARY SWITCH command is routed to all systems in the logical partition cluster to ensure that all systems run with the same configuration of managed channel paths.

This command affects only managed device paths. Nonmanaged paths must be varied online or offline separately.

Note: If you specify DCM=OFFLINE or =ONLINE for a logical partition cluster spanning multiple logical channel subsystems, you must issue the command from a system at the z990 V1R4 exploitation support level or higher. For example, if systems A and B have z990 V1R4 compatibility support and are running in CSS 0, while systems C and D have exploitation support and are running in CSS 1, you must issue the DCM= command from system C or system D.

The format of the VARY SWITCH command is:

```
V SWITCH(ssss,pp[-pp][,pp[-pp]]...),DCM=OFFLINE[,UNCOND]  
      ,DCM=ONLINE
```

ssss

specifies the switch device number.

pp[-pp] [,pp[-pp]]

specifies the switch port address or address list.

DCM=OFFLINE[,UNCOND]

specifies that the switch port is to be varied offline to dynamic channel path management. If UNCOND is specified, then the UNCOND option will be passed

to the VARY PATH commands that are invoked as a result of this VARY SWITCH command. See “Placing an I/O Path or Paths Online or Offline” on page 4-639.

DCM=ONLINE

specifies that the switch port is to be varied online to dynamic channel path management.

Example 1

To vary port 60 of switch B000 offline on the two systems in a logical partition cluster, enter on system MVS1:

```
VARY SWITCH(b000,60),DCM=OFFLINE
```

The response from this command will show how it ran on both systems, MVS1 and MVS2:

```
MVS1 IEE633I SWITCH B000, PORT 60, DCM STATUS=OFFLINE
ATTACHED NODE = 003990.OCC.IBM.XG.000000000006
THE FOLLOWING DEVICE PATHS ARE ONLINE THROUGH THIS PORT:
(0220,58)
```

```
MVS2 IEE633I SWITCH B000, PORT 60, DCM STATUS=OFFLINE
ATTACHED NODE = 003990.OCC.IBM.XG.000000000006
THE FOLLOWING DEVICE PATHS ARE ONLINE THROUGH THIS PORT:
(0220,58)
```

These messages show that the command ran on both system MVS1 and MVS2. The fact that these messages are identical shows that these systems are configured identically — the preferred configuration when using dynamic channel path management.

Controlling an Application Environment

Use the VARY WLM,APPLENV=*applenvname*, or VARY WLM,DYNAPPL=*applenvname* command to control an application environment. You can perform the following:

- Request that the server address spaces for an application environment be terminated and start new ones in their place (REFRESH). This is useful if you have updated resources such as load modules which might have been cached by the servers.
- Request that the server address spaces for an application environment be terminated and that any additional work requests be queued but not selected (QUIESCE or Q).
- Restart the server address spaces for an application environment that was previously quiesced, or was stopped by workload management when it detected an error condition (RESUME).

You can use the DISPLAY WLM command to check the status of the VARY WLM command for the application environments that are affected. See the “Defining Application Environments” chapter in *z/OS MVS Planning: Workload Management* for more information on operational considerations for application environments and the role of the VARY WLM,APPLENV=*applenvname*, or VARY WLM,DYNAPPL=*applenvname* command.

Command Scope

The VARY WLM,APPLENV=*applenvname* command has a sysplex scope and so affects all servers of an application environment on all the systems in the sysplex. For subsystems that can have multiple instances, all the subsystem instances are affected.

If you need to stop application environment activity on just one system in a sysplex, use the subsystem-specific interface to stop the activity on that system.

The VARY WLM,APPLENV command is processed on the coordinator system, which is determined dynamically and may not be the local system where the command was issued. The command output is routed to the console from which the command has been issued, but the command hardcopy is logged on the system on which the command has been processed. Therefore the message may not appear in the SYSLOG on the system where the command was entered.

The VARY WLM,DYNAPPL=*applenvname* has a single system scope and affects only servers of an application environment on the system where the command is issued or to where it is routed.

Workload Management Mode Considerations

Important

Beginning with z/OS V1R3, WLM compatibility mode is no longer available. Accordingly, the information below that pertains specifically to WLM compatibility mode is no longer valid. It has been left here for reference purposes, and for use on backlevel systems.

You can enter the VARY WLM,APPLENV=*applenvname* command on any system in the sysplex in either compatibility or goal mode. However, if a system is in compatibility mode then that system does not start or terminate servers. In this case, the installation is responsible for starting and terminating servers. For example, if VARY WLM,APPLENV=*applenvname*,QUIESCE is issued, the application environment enters the quiescing state on all systems in the sysplex. However, if any servers exist on a compatibility mode system, they must be terminated with the MVS CANCEL command or through a subsystem-specific interface before the QUIESCE is considered complete.

Note that workload management starts servers on goal mode systems only if a JCL procedure has been defined in the service definition for the application environment. This is known as “automatic” control for the application environment. If a JCL procedure is not defined, the application environment is under “manual” control, and the installation is responsible for starting the servers.

Table 4-45 on page 4-661 summarizes how the QUIESCE, RESUME, and REFRESH options of the VARY WLM,APPLENV command are acted on by systems:

- Under “Automatic” control, that is, goal mode when a JCL procedure is defined for the application environment
- Under “Manual” control, that is, goal mode when a JCL procedure is *not* defined for the application environment
- In Compatibility mode

Table 4-45. Goal Mode and Compatibility Mode Actions for VARY WLM,APPLENV

Operation	Goal Mode Systems		Compatibility Mode Systems
	Automatic Control: JCL Procedure Defined to WLM	Manual Control: No JCL Procedure Defined to WLM	
QUIESCE	WLM stops the server address spaces.	WLM stops the server address spaces.	Operator must cancel the server address spaces.
RESUME	WLM starts the server address spaces.	Operator must start the server address spaces.	Operator must start the server address spaces.
REFRESH	WLM stops the server address spaces, and starts new ones.	WLM stops the server address spaces. The operator must start new ones.	Operator must cancel the server address spaces and start new ones.

Command Syntax

The syntax of the VARY WLM,APPLENV command is:

```
V WLM,APPLENV=applenvname,{REFRESH}
{QUIESCE|Q}
{RESUME}
```

WLM,APPLENV=applenvname

Specifies the 1 to 32 character name of the application environment for the command.

The VARY WLM command is rejected if the named application environment does not exist in the workload management service definition. To list all the application environment names, use the DISPLAY WLM,APPLENV=* command.

WLM,DYNAPPL=applenvname

Specifies the name (length=1–32 characters) of the dynamic application environment for the command. The VARY WLM,DYNAPPL command is rejected if the named dynamic application environment was not defined to the system. To list all the dynamic application environment names, use the DISPLAY WLM,DYNAPPL=* command. The following keywords are valid:

SNODE=nodename

When SNODE=nodename is used, the VARY WLM,DYNAPPL command specifically applies to the dynamic application environments with the specified node name.

SNAME=subsystemname

When SNAME=subsystemname is used, the VARY WLM,DYNAPPL command specifically applies to the dynamic application environments with the specified subsystem name.

STYPE=subsystemtype

When STYPE=subsystemtype is used, the VARY WLM,DYNAPPL command specifically applies to the dynamic application environments with the specified subsystem type.

REFRESH

Specifies that the application environment server address spaces be terminated after completion of the currently executing request and new ones started in their place.

VARY Command

QUIESCE | Q

Specifies that the application environment server address spaces be terminated after completion of the currently executing request. No new server address spaces can be started for the application environment by either WLM or an operator. Additional work requests for an application environment that supports queueing, are queued but not selected.

When an application environment is quiesced, changes can be made to libraries, procedures, and other items for the application environment.

To restart the application environment, use the VARY
WLM,APPLENV=*applenvname*,RESUME command; any other VARY
WLM,APPLENV action is rejected by the system.

To restart a dynamic application environment, use the VARY
WLM,DYNAPPL=*applenvname*,RESUME command; any other VARY
WLM,DYNAPPL action is rejected by the system.

RESUME

Specifies that the application environment be restarted. After this command is executed, server address spaces are allowed to start. Work requests that are queued are eligible for selection.

Example 1

To quiesce application environment db2pay, enter:

```
V WLM,APPLENV=db2pay,Q
```

The system responds with:

```
IWM032I VARY QUIESCE FOR DB2PAY COMPLETED
```

Example 2

To quiesce dynamic application environment websphere in node *testnode*, enter:

```
V WLM,APPLENV=websphere,snode=testnode,Q
```

The system responds with:

```
IWM032I VARY QUIESCE FOR WEBSPHERE COMMAND COMPLETED
```

Activating a Service Policy

Use the VARY WLM command to activate a named service policy for a sysplex. The service policy must be defined in the workload management service definition and must have been previously installed on the WLM couple data set. In addition, each system in the sysplex must have connectivity to the WLM couple data set in order to participate in the service policy activation.

To activate a new policy, issue a VARY WLM command with the name of the policy you want to make active. This in effect makes the previous policy inactive.

You can also activate a workload management service policy by using the online ISPF administrative application. Refer to *z/OS MVS Planning: Workload Management* for more information or see your service administrator.

Important

Beginning with z/OS V1R3, WLM compatibility mode is no longer available. Accordingly, the following information that pertains specifically to WLM compatibility mode is no longer valid. It has been left here for reference purposes, and for use on backlevel systems.

You can enter the VARY command on any system in the sysplex. This command activates the named service policy on all systems in the sysplex, regardless of the workload management mode in effect on a system. However, only systems operating in workload management goal mode will manage towards that service policy. If there is an active service policy on a system running in compatibility mode, and you issue the MODIFY command to switch that system into goal mode, workload management manages the system using the service policy you activated. The VARY command does not change the workload management mode in effect on any system. Use the MODIFY command to change workload management modes.

You can use the DISPLAY WLM command to check the service policy currently active for the sysplex. For example, before you activate a service policy, check which, if any, policy is active using the DISPLAY WLM command. After activating a service policy using the VARY command, you can confirm that the VARY command has taken effect by using the DISPLAY WLM command.

If you routinely activate service policies based on time-of-day or day-of-week, you can update automation packages with the commands.

Only one service policy can be in effect throughout all systems in a sysplex at any one time.

```
V WLM,POLICY=policyname[,REFRESH]
```

WLM,POLICY=*policyname*

Specifies the 1 to 8 character name of the service policy to be activated.

REFRESH

Specifies that WLM is to discard historical workload characterization data, reset to begin data collection anew, and activate the named policy.

Note: Use REFRESH only when directed to do so by IBM Level 2 personnel.

The VARY WLM command is rejected if the named service policy does not exist. Contact your service administrator to determine the name of the desired service policy.

Example 1

To activate a service policy named SHIFT1, enter:

```
V WLM,POLICY=shift1
```

The system responds with:

```
IWM001I WORKLOAD MANAGEMENT POLICY SHIFT1 NOW IN EFFECT
```

VARY Command

Example 2

If you activate a service policy that does not exist, the command is rejected. If service policy WEEKEND does not exist and you enter:

```
V WLM,POLICY=weekend
```

The system responds with:

```
IWM003I VARY WLM FAILED, POLICY NAME WEEKEND NOT DEFINED
```

Removing a System from the XCF Sysplex

Use the following form of the VARY command to remove a system from the XCF sysplex.

```
V XCF,systemname,{OFFLINE|OFF} [,RETAIN={YES|NO}] [,FORCE]
```

XCF,systemname,OFFLINE or OFF

Specifies the name of a system that XCF is to remove from the sysplex. The system that is removed is put into a wait state. The system to be brought offline should be shut down completely, including the issuance and completion of the HALT EOD command, before the VARY XCF,sysname,OFFLINE command is issued.

Note that the V SMS,SMSVSAM,TERMINATESERVER command must be issued to terminate the SMSVSAM address space prior to partitioning a system from the sysplex. If a VARY XCF,sysname,OFFLINE is issued while the SMSVSAM address space is active, miscellaneous abend code X'0F4's can be issued as a result.

RETAIN=YES or NO

Indicates whether or not XCF, on the remaining systems in the sysplex, is to retain the signalling path resources used to communicate with the system that's removed. If you specify YES, the XCF signalling paths used to communicate with the removed system remain allocated. They are reinitialized so that they are ready to reestablish communications with a new system if the removed system joins the sysplex or another system takes its place. If you specify NO, XCF stops the signalling path to stop the XCF paths that had communicated with the removed system.

If a replacement for the removed system later joins the sysplex, after RETAIN=NO, you must issue the SETXCF START path command on the remaining systems. This procedure guarantees that each signalling path can communicate with the replacement system.

FORCE

Indicates that XCF will immediately remove the specified system from the sysplex. The FORCE option is only accepted after XCF has failed to remove the system with the VARY command. The VARY command with the FORCE option must be issued on the same MVS image where the original VARY command was issued.

Notes:

1. Use FORCE only at the direction of the system programmer.
2. Before using FORCE: to avoid damage to sysplex resources ensure that the target system has been through a SYSTEM RESET.

WRITELOG Command

Use the WRITELOG command to control the system log. Using WRITELOG, you can start, stop, or print the system log, or modify the output class of the system log.

Syntax

The complete syntax for the WRITELOG command is:

```
W [class|CLOSE|START]
```

Note: Specifying WRITELOG without any operands schedules the system log to be printed with the default output class specified at system installation. If no default output class was specified at system installation, specifying WRITELOG without any operands schedules the system log to be printed with output class A.

Parameters

class

The one-character output class (A-Z, 0-9) to be used when printing the contents of the system log. This command is in effect only for the current scheduling of the system log output. All subsequent scheduling is to the default output class unless the class parameter is again entered.

CLOSE

The system log is closed and the log function is discontinued. This command is rejected if the system log is the hardcopy medium.

START

The system log is to be restarted.

Example

To schedule the system log to the class D output queue, enter:

```
WRITELOG d
```

WRITELOG Command

Appendix. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
 - Operate specific or equivalent features using only the keyboard
 - Customize display attributes such as color, contrast, and font size
-

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

www.ibm.com/servers/eserver/zseries/zos/bkserv/

Notices

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States and/or other countries:

- CICS
- DB2
- DFSMS/MVS
- ES/3090
- ES/9000
- ESCON
- FICON
- GDPS
- Hiperspace
- HyperSwap
- IBM
- IBMLink
- IMS
- Language Environment
- MVS
- MVS/ESA
- MVS/SP
- NetView
- OS/390
- Parallel Sysplex
- PR/SM
- RACF
- RMF
- SAA
- SOM
- SOMobjects
- Sysplex Timer
- System Object Model
- VTAM
- z/Architecture
- z/OS
- z/OS.e
- zSeries
- 3090
- 400

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

Special characters

- * (asterisk)
 - specified on identifier of system command 4-310
 - specified on jobname.identifier of system command 4-21, 4-591
 - specified on parameter of system command 1-17, 4-154, 4-312, 4-313, 4-314, 4-339, 4-341

Numerics

- 3277-2 display station
 - in message stream mode 3-27
 - illustration 3-27
- PFK
 - displaying number 4-60
 - selector pen 3-12
- 3380 Direct Access Storage
 - Model AA4
 - recovery 1-57
 - 3490 tape device 4-633
- 3880 Storage Control
 - controlling 3380 1-57
- 3990-3
 - SETSMS command
 - DINTERVAL parameter 4-465

A

- A (RELEASE command) 4-2
- ACBNAME parameter in APPCPMxx 4-397
- accessibility A-1
- ACDS parameter
 - SETSMS command 4-459
- ACR (alternate CPU recovery) 1-43
 - machine check 4-305
- action
 - action for a SLIP trap 4-514
- action message
 - backup 3-27
 - deleting
 - eventual action 4-58
 - immediate action 4-58
 - retained immediate 4-58
 - displaying information 3-33
 - retained
 - deleting 4-58
 - retained by action message retention facility 3-29, 3-33
- retention facility 4-58
 - activating 4-63
 - cancelling 4-63
 - changing 4-63
 - deactivating 4-63
 - deleting 4-59
 - displaying status 4-63
- action message retention facility
 - cancelling 3-33
- action message retention facility (*continued*)
 - deactivating 3-33
 - description 3-33
 - displaying action messages not retained 3-34
 - retrieving an action message 3-33
- ACTION parameter
 - SLIP command 4-528
- ACTIVATE command 4-15
 - example 4-18
 - specifying FORCE 4-17
 - summary table 4-2
 - syntax 4-15
- ACTIVATE parameter
 - SETSSI command 4-473
- ACTIVATE=SERVICE on DISPLAY command 4-177
- activating a workload management service policy
 - controlling 1-20
- activation
 - action message retention facility 3-33
 - general WTO installation exit IEAVMXIT 3-34
- ACTIVE parameter
 - DISPLAY SMS command 4-218
- ADD dump mode 4-25
- ADD parameter
 - SETSSI command 4-472
- address
 - direct 4-508
 - indirect 4-508
 - wraparound 4-555
- address compare in SLIP
 - definition 4-540
- address space
 - qualifying direct or indirect address 4-511
- Advanced Program-to-Program Communication/MVS (APPC/MVS)
 - address space information
 - changing 4-382
- AFTER parameter
 - in PROGxx parmlib member 4-449
- AKP parameter
 - SET SMS command 4-387
 - SETSMS command 4-459
- alternate nucleus
 - load 1-5
- alternate path 4-51, 4-636
 - recovery 1-55
- ALTLU parameter in APPCPMxx 4-399
- APF (authorized program facility)
 - listing authorized libraries 4-391
- APF list
 - updating 4-441
- apostrophe
 - use 4-1
- ARM (automatic resource management)
 - switching couple data containing ARM information 4-485
- ASIDIA on DISPLAY command 4-175

assign
 command group 4-621, 4-627
 MCS (multiple support console) console 4-620
 routing code 4-621, 4-627
assignment
 message level 3-31
 routing code 3-30
ATBOTTOM parameter
 in PROGxx parmlib member 4-449
ATTOP parameter
 in PROGxx parmlib member 4-449
authorization
 changing console 3-26
authorization to system commands 3-24
AUTOACT 4-620
automatic
 AVR (automatic volume recognition) 1-31
automatic message deletion
 assigning 4-72
automatic restart 1-23
automatic restart management
 DISPLAY XCF command 4-256
automatically switchable tape devices
 change the AUTOSWITCH attribute 4-632
 definition 1-27
 obtain the status of 1-28, 4-241
 operational considerations 1-27
automation
 effects of altering console attributes 3-20
auxiliary storage
 adding 4-349
 deleting 4-351
 draining 4-351
 replacing 4-351
AVR (automatic volume recognition) 1-31

B

base address 4-509
BASE parameter in APPCPMxx 4-398
base/displacement shorthand in SLIP command 4-509
binary compare in SLIP
 definition 4-540
blank screen
 error response 3-15
 response 3-16
BMFTIME parameter
 SETSMS command 4-465
box a device 1-32
BPXPRMxx parmlib member
 changing, with SETOMVS 4-427
branch
 SLIP trap 4-520
BRL on DISPLAY command 4-175
broadcast
 message
 level 4-78
 routing 4-78
broadcast data set 1-21
broadcast message
 description 3-31

buffer shortage 3-16, 3-19

C

C command
 See CANCEL command

cache
 subsystem 4-349

CACHE parameter
 DISPLAY SMS command 4-218

CACHETIME parameter
 SETSMS command 4-465

cancel
 job 4-19
 status display for an inactive console 3-15

CANCEL action
 how to perform 3-8
 perform 3-7, 3-11, 3-13, 3-14

CANCEL command 1-22, 4-19
 example 4-22
 summary table 4-2
 syntax 4-20

CAPSIC on DISPLAY command 4-176

catalog address space 4-309
 information
 displaying 4-321
 service
 requesting 4-321

CD command
 See CHNGDUMP command

central storage
 job 1-23
 moving online or offline 4-48
 physical view 2-4
 reconfiguring 2-10
 specifying configuration 1-2
 specifying RSU parameter
 example 2-7

CF command
 See CONFIG command

CFCACHE parameter
 DISPLAY SMS command 4-218

CFRM (coupling facility resource management)
 switching couple data containing CFRM
 information 4-485

change
 dump mode and option 4-26
 status
 of secondary console 4-631
 system commands a console can receive 4-620

change information in the entry area 3-13

changing authorization
 console 3-26

changing resource states
 MODIFY command 4-337

channel path
 alert condition 1-49
 initialized condition 1-49
 outstanding reserve 1-47
 permanent error 1-49
 placing online or offline 4-49

channel path (*continued*)
 reconfiguring 2-10, 4-49
 recovery 1-47
 terminating error 1-47
 channel subsystem
 error 1-46, 1-51
 missing interrupt 1-50
 monitoring facility
 recovery 1-49
 characters
 valid command 4-1
 checkpoint/restart 1-22, 1-24
 effect on SLIP PER trap 4-518
 CHNGDUMP command 4-25
 DEL operand
 syntax 4-26
 RESET operand
 syntax 4-32
 scope in a sysplex 4-26
 SET operand
 syntax 4-34
 summary table 4-2
 CICSVR
 VARY SMS command 4-657
 CICSVR,RCDS
 VARY SMS command 4-658
 class C1 commands 1-37
 class C2 commands 1-38
 class M1 commands 1-35
 class M2 commands 1-36
 clock
 setting 4-357
 CLOCKxx member of SYS1.PARMLIB
 OPERATOR PROMPT parameter 1-8
 CLOCKxx parmlib member
 OPERATOR PROMPT parameter 4-358
 CLPA (create link pack area)
 option 4-349
 CNS example
 SETGRS 4-410
 COFDLFxx member
 keyword
 MAXEXPB 4-331
 PCTRETB 4-331
 color
 description 3-3
 color options
 displayed 3-34
 command
 check PFK command 3-9
 enter
 using PFK (program function key) 3-9
 using selector pen 3-12
 enter with selector pen 3-12
 entry error correction 3-8
 formats 4-13
 how to enter
 using keyboard 3-7
 MODIFY
 example 4-67
 routing 4-367

command (*continued*)
 select for hardcopy message set 3-38
 syntax conventions 4-12
 command authority 3-24
 command characters
 valid 4-1
 command delimiter 3-8
 command entry 3-8
 command entry error correction 3-8
 command example
 CONTROL A 4-58
 CONTROL C,A 4-59
 CONTROL C,E 4-59
 CONTROL C,I 4-59
 CONTROL D 4-61
 CONTROL M,AMRF 3-33, 4-63
 CONTROL M,LOGLIM 4-64
 CONTROL M,MLIM 4-65
 CONTROL M,RLIM 4-65
 CONTROL M,UEXIT 3-34, 4-67
 CONTROL N,PKF 3-36
 CONTROL Q 4-72
 CONTROL S 3-32, 4-75
 CONTROL V,LEVEL 3-32
 CONTROL V,USE 3-28
 DEVSERV P 4-90
 DEVSERV SMS 4-91
 DISPLAY 4-149, 4-192, 4-193, 4-194, 4-244, 4-248, 4-249
 DISPLAY PFK 3-36
 DUMP 4-274
 FORCE 1-26, 4-293
 IEE094D 4-275
 LOG 4-299
 LOGOFF 4-300
 MODE 4-306, 4-307
 MODIFY 4-313, 4-319, 4-334, 4-338, 4-341
 MODIFY CATALOG 4-330
 MONITOR 4-344
 MOUNT 4-346
 MSGRT 4-348, 4-349
 PAGEADD 4-350
 PAGEDEL 4-352
 REPLY 4-356
 RESET 4-365
 ROUTE 4-372
 SEND 4-376, 4-379, 4-381
 SET 3-36, 4-393, 4-395
 SETCEE 4-402
 SETDMN 4-406
 SETETR 4-407
 SETGRS 4-410
 SETIOS 4-417
 SETLOAD 4-420
 SETLOGR 4-424
 SETPROG EXIT 4-443, 4-446
 SETPROG LNKLST 4-450
 SETSMF 4-456
 START 4-579, 4-582, 4-589
 STOP 4-592
 STOPMN 4-596

command example (*continued*)
 STOPTR 4-597
 SWAP 4-600
 TRACE 4-608
 TRACK 4-614
 VARY 3-30, 3-38, 3-40, 4-627, 4-632, 4-641, 4-662, 4-663, 4-664
 VARY HARDCPY 3-39, 4-631
 WRITELOG 4-665
 command flooding 1-35
 command group 3-24
 assigning 4-621, 4-627
 figure showing 3-26
 summary 3-24
 COMMNDxx parmlib member 4-507
 common storage tracking
 SET command 4-392
 communication
 operator-to-operator 4-375
 terminal user 4-377
 time-sharing user 4-376
 component trace 1-27, 4-605
 option
 replying 4-359
 specifying 4-359
 CONFIG command 4-46
 example 4-51, 4-54, 4-55
 MSGRT command routing 4-347
 reconfiguration 2-1
 summary table 4-2
 configuration
 changing 1-23
 CONFIGxx member
 SYS1.PARMLIB
 reconfiguration 2-1
 CONS command authority 3-24
 console
 activity
 lack 3-15
 activity, lack 3-15
 changing authorization 3-26
 console use 3-26
 control command 3-26
 description 3-2
 device 3-1
 error condition 3-14
 hardware at IPL 1-2
 hardware error 3-15
 inactivity
 cause 3-15
 response 3-15
 MCS (multiple console support)
 definition 3-2
 use 3-2
 MCS at IPL 1-2
 NIP at IPL 1-2
 operation 3-1
 recovery 3-15
 response to an inactive 3-15
 varying offline 3-19
 console area
 L= operand 3-5
 console assignment
 making 3-24
 console attributes
 alteration and effects on automation 3-20
 console characteristics
 changing 3-20
 defining 3-20
 displaying 3-23
 console in error condition 3-14
 console inactivity
 cause 3-15
 console message backup
 response 3-16
 console operating mode
 definition 3-26
 setting up message stream console 3-26
 setting up status display console 3-26
 console operation
 full-capability mode
 changing 4-77
 displaying 4-77
 output-only mode
 changing 4-77
 displaying 4-77
 console operation mode
 effect on display screen 3-3
 console screen
 information
 removing 4-62
 console specification
 altering 4-72
 changing 4-72
 current 4-72
 displaying 4-72
 console-to-console
 status display
 routing 4-347
 sending 4-347
 CONSOlx member of Parmlib
 CMDLEVEL parameter 4-628
 ROUTCODE parameter 4-628
 CONSOlx member of SYS1.PARMLIB
 AMRF parameter 3-21
 example 3-33
 AREA parameter 3-20
 AUTH parameter 3-20
 CMDELIM parameter 3-21
 CMDLEVEL parameter 3-22
 CON parameter 3-20
 DEL parameter 3-20
 DEVNUM parameter 3-22
 general description 3-2
 HARDCOPY statement 3-22
 INIT statement 3-21
 LEVEL parameter 3-20
 example 3-31
 MFORM parameter 3-20
 MLIM parameter 3-16, 3-21

CONSOLxx member of SYS1.PARMLIB (*continued*)
 MONITOR parameter on the CONSOLE
 statement 3-20
 MONITOR parameter on the INIT statement 3-21
 MPF parameter 3-21
 PFK (program function key) parameter 3-21
 PFKTAB parameter 3-20, 3-37
 relationship with CONTROL command 3-20
 RLIM parameter 3-16, 3-21
 RNUM parameter 3-20
 ROUTCODE parameter 3-20, 3-22
 RTME parameter 3-20
 SEG parameter 3-20
 UEXIT parameter 3-21
 USE parameter 3-20
 UTME parameter 3-20
 CONSOLxx parmlib member
 MLIM parameter 3-18
 MSGRT parameter 4-347
 RLIM parameter 3-18
 contents compare in SLIP
 definition 4-540
 control
 message format 3-6, 3-32
 message processed by message processing facility
 (MPF) 3-32
 status displays on a status display console 3-26
 system message 3-29
 CONTROL command 4-56
 A
 example 4-58
 syntax 4-57
 C,A
 example 4-59
 syntax 4-58
 C,CE
 syntax 4-58
 C,D
 syntax 4-60
 C,E
 example 4-59
 C,I
 example 4-59
 syntax 4-58
 D
 example 4-61
 syntax 4-60
 duplicate line
 K Q 4-72
 E
 example 4-62
 syntax 4-62
 example 3-12
 extended MCS console
 limitation 4-56
 M,AMRF
 example 3-33, 4-63
 syntax 4-63
 M,LOGLIM
 example 4-64
 syntax 4-64

CONTROL command (*continued*)
 M,MLIM
 example 4-65
 syntax 4-64
 M,RLIM
 example 4-65
 M,UEXIT
 example 3-34, 4-67
 syntax 4-67
 MSGRT command routing 4-347
 N,PFK
 example 3-36, 4-71
 syntax 4-70, 4-71
 Q
 example 4-72
 syntax 4-72
 relationship to CONSOLE statement in
 CONSOLxx 3-20
 S
 example 3-32, 4-75
 syntax 4-72
 scope in a sysplex 4-57
 SMCS APPLID
 SMCS 4-68
 status display
 framing 4-60
 summary table 4-2
 T
 syntax 4-76
 V,LEVEL
 example 3-32, 4-79
 syntax 4-78, 4-79
 V,USE
 example 3-28, 3-29
 syntax 4-77
 CONTROL Q command
 example 3-15
 CONTROL T command
 example 3-15
 controlling
 system 1-9
 conversational mode
 enter command assigned to PFK 3-10
 enter command with selector pen 3-12
 message deletion 4-72
 canceling 4-73
 PKF 4-70
 PKF (program function key) 3-35, 3-36
 COPYFROM parameter
 in PROGxx parmlib member 4-447
 coupling facility
 reconfiguration 2-11
 coupling facility resource management
 DISPLAY XCF command 4-256
 CPU
 error 1-41
 hard 1-42
 soft 1-41
 terminating 1-43
 hard error 1-42

CPU (*continued*)
 recovery
 hard error 1-42
 restart 1-59
 soft error 1-41
 terminating error 1-43
 restart 1-59
 soft error 1-41
 terminating error 1-43
 critical action
 message
 level 4-78
 routing 4-78
 critical eventual action
 message
 level 4-78
 routing 4-78
 critical eventual action message
 description 3-31
 cross system coupling facility (XCF)
 displaying information 4-251
 cross-system coupling facility (XCF) 4-483
 active configuration
 saving 4-483
 alternate data set couple
 defining 4-483
 data set couple
 controlling 4-483
 SETXCF command 4-491
 signal path
 modifying 4-483
 starting 4-483
 stopping 4-483
 transport class
 defining 4-483
 deleting 4-483
 modifying 4-483
 XCF PR/SM policy 4-491
 CRW (channel report word)
 describing error 1-46
 cursor
 description 3-3
 move 3-7
 cursor detect action
 perform 3-7
 CVIO (clear virtual I/O)
 option 4-349

D

D command
 See DISPLAY command

DAE (dump analysis and elimination)
 parameter 4-382, 4-385
 suppression of the dump 4-530

DASD (direct access storage device)
 maintenance 1-58
 recovery 1-58

data lookaside facility (DLF)
 starting data lookaside facility (DLF) 4-587

data removal
 data from entry area 3-14

data set
 clearing 4-285
 dump
 adding 4-280, 4-281
 allocation of, automatic 4-280
 clearing 4-280
 deleting 4-285
 disable automatic allocation 4-284
 enable automatic allocation 4-284
 name-pattern 4-287

DATASET parameter in APPCPMxx 4-401

date
 setting 4-357, 4-384

DD (data definition) statement 1-29

DDR (dynamic device reconfiguration) 1-39, 4-599
 failing device
 recovery 1-54
 operator-requested DDR 4-599

DEACTIVATE parameter
 SETSSI command 4-473

dedicated tape device
 definition 1-27

DEFAULT statement 3-23

deferred restart 1-23

define
 routing codes for console 4-625, 4-627

definition
 command for PFK (program function key) 3-35, 3-36
 routing codes for a console 3-30
 status display console 3-26

degradation interruption
 machine check
 controlling recording 4-303
 recording 4-303

delete
 SLIP trap 4-574

DELETE parameter in CUNUNIx 4-481

deletion
 message
 from message stream console 3-27

delimiter of command 3-8

device
 allocation 1-29
 assignment 1-30
 boxing 1-32
 failing
 recovery 1-54
 hung
 recovery 1-53
 I/O device online or offline 4-633
 place online or offline 4-633
 supported and not supported by DDR 1-39

device status
 displaying 1-10

Device Support Facilities
 See ICKDSF

DEVSERV command 4-80

DEVSERV command 4-80 (*continued*)
 channel path status
 displaying 4-80
 complete syntax 4-80
 device status
 displaying 4-80
 example 4-90
 SMS-managed device status
 displaying 4-80
 summary table 4-2
 use 1-10

DFSMStvs
 VARY SMS command 4-648, 4-649

DINTERVAL parameter
 SETSMS command 4-465

direct address in indirect addressing 4-508

disability A-1

disable status
 meaning 4-646
 DFSMStvs instance 4-649

disabled loop 1-18

displacement in indirect addressing 4-508

display
 action messages awaiting action 3-33
 action messages not retained 3-34
 active job 1-23, 1-24
 color options 3-34
 console characteristics 3-23
 console operating mode 3-23
 console specifications 3-23
 console status 3-23
 current system status 1-10
 device status 1-10
 hardcopy information 3-40
 highlighting options 3-34
 information about system log 3-31
 intensity options 3-34
 job information 1-10, 4-613
 logged-on TSO/E (Time Sharing Option/Extension)
 users 1-21
 message deletion specifications 3-23
 message levels not assigned to any console 3-31
 message processing facility (MPF) message
 processing 3-34
 number of allowed WTO message buffers 3-23
 number of allowed WTOR message buffers 3-23
 PFK (program function key) definition 3-23
 routing codes not assigned to any console 3-31
 status of installation exit 3-34
 system activity 1-10
 time and date 3-15

DISPLAY command 4-97
 ACTIVATE=SERVICE operand 4-177
 ASIDIA operand 4-175
 BRL operand 4-175
 CAPSIC operand 4-176
 CONSOLES
 example 3-31, 3-40
 displaying active service policy 4-245
 displaying general IPL information used by the
 system 4-148

DISPLAY command (*continued*)
 displaying IPL information 4-148
 displaying PARMLIB information 4-192
 displaying RTLS information 4-211
 displaying the logical parmlib setup for a
 system 4-192
 displaying workload management mode in
 effect 4-245
 example 4-244, 4-248, 4-249
 EXCEPTIONIE operand 4-176
 FILEIF operand 4-175
 LIMITSIL operand 4-176
 M 1-10
 MEMLIMIT operand 4-176
 MSGRT command routing 4-347
 NAMEIN operand 4-175
 OMVS operand 4-174
 OPTIONSIO operand 4-176
 OWNERIO operand 4-176
 PF
 example 3-36
 PFK (program function key)
 example 3-10
 PID operand 4-175
 R
 example 4-59
 RESET operand 4-177
 scope in a sysplex 4-99
 SER operand 4-177
 summary table 4-2
 SUMMARYIS operand 4-175
 TYPEIT operand 4-176
 U
 use 1-10
 U operand 4-175
 use 1-10
 VSERVERIV operand 4-176

DISPLAY Command
 MIDAW facility status 4-144

display console
 features 3-3

DISPLAY CONSOLES command
 example 3-17
 sample 3-18
 sample output 3-18

DISPLAY IPLINFO command
 example 4-149

DISPLAY PARMLIB command
 example 4-192, 4-193, 4-194

display screen area
 on console 3-3

DISPLAY SMS command
 ACTIVE parameter 4-218
 CACHE parameter 4-218
 CFCACHE parameter 4-218
 description 4-215
 SHCDS parameter 4-222
 SMS parameter 4-218
 STATUS parameter 4-222

DISPLAY SSI command
 ALL parameter 4-233

DISPLAY SSI command (*continued*)

 DYNAMIC parameter 4-233

 FUNC parameter 4-233

 LIST parameter 4-233

 SSI parameter 4-233

 STATUS parameter 4-233

 SUBSYS parameter 4-234

 syntax 4-232

DISPLAY WLM command

 syntax 4-245

DISPLAY XCF command

 syntax 4-251

 displaying device status and allocation

 the IPL volume 4-241

 Displaying SMS information 4-215

DISSTIMEOUT parameter

 SETSMS command 4-465

DLF (data lookaside facility)

 changing 4-331

 defined 4-587

 displaying 4-332

 modifying 4-331

 stopping address space 4-593

DLF (data lookaside facility) address space

 stopping 4-593

DLF (data lookaside facility) data object

 starting 4-587

DLOG (JES3 system log) 3-39

domain description table

 changing 4-396, 4-403, 4-405

DS command

 See **DEVSERV command**

DSNAME parameter

 in PROGxx parmlib member 4-448

 SETSMS command 4-468

dump

 clearing 4-285

 data set

 adding 4-280, 4-281

 clearing 4-280

 deleting 4-285, 4-287

 disable automatic allocation 4-284

 enable automatic allocation 4-284

 name-pattern 4-287

 mode and option 4-25

 option

 specifying 4-263

 resources

 adding 4-281

 deleting 4-285, 4-286

DUMP command

 example 4-274

 option

 replying 4-359

 specifying 4-359

 summary table 4-2

 syntax

 complete 4-262

dump options list

 for SYSABEND dump type 4-25

 for SYSMDUMP dump type 4-25

dump options list (*continued*)

 for SYSUDUMP dump type 4-25

DUMPDS command

 summary table 4-2

dynamic device allocation 1-29

dynamic display

 time interval

 changing 4-76

 displaying 4-76

dynamic exits

 display 4-198

 update 4-443, 4-446

dynamic I/O configuration 2-1

dynamic LPA

 managing 4-451

Dynamic LPA module information

 display 4-202

dynamic parmlib concatenation

 dynamic Changing

 dynamic changing 4-420

dynamic pathing support

 array out-of-sync

 recovery 1-58

dynamically activating maintenance

 z/OS Unix System Services 4-335

E

E command

 See **RESET command**

enable status

 meaning 4-646

 data set 4-649

 DFSMStvs instance 4-649

enable/disable switch

 3380 1-57

enablement policy

 changing 4-382

 display 4-195

ENQMAXA example

 SETGRS 4-411

ENQMAXU example

 SETGRS 4-410

ENTER action

 perform 3-7

enter command

 using keyboard 3-7

entry area

 blank out 3-14

 change information 3-13

 character insertion 3-13

 description 3-4

 error correction 3-13

error

 CPU 1-41

 storage 1-45

error code 1-6

error event

 trap 4-514

error message

 response 3-15

error recovery 1-29
 in SMF 1-25
 switch data set 1-26
 error response
 blank screen 3-15
 console error condition 3-14
 error message 3-15
 system error 3-15
 ES/3090
 storage element size 2-6
 ETR (External Time Reference)
 machine check 1-42
 eventual action
 message
 deleting 4-58, 4-59
 eventual action message
 description 3-31
 example of command
 CONFIG 4-51, 4-54, 4-55
 CONTROL D command 3-12
 CONTROL E command 3-12
 DISPLAY CONSOLES 3-18
 DISPLAY PFK 3-10
 DISPLAY T 3-15
 SEND command 4-1
 EXCEPTIONIE on DISPLAY command 4-176
 extended highlight capability
 description 3-3
 external writer
 data set selection
 specifying criteria 4-338
 MODIFY command
 data set selection 4-338
 starting 4-588

F

F command
 See MODIFY command
 fetch
 SLIP trap 4-520
 FILEIF on DISPLAY command 4-175
 flagged
 message
 deleting 4-62
 FORCE command 1-22
 caution 4-290
 example 1-26, 4-293
 summary table 4-2
 syntax
 complete 4-291
 using 4-290
 FORCE operand of CONFIG command
 cautions about using 1-32, 4-50
 FORCE operand of VARY command
 cautions about using 1-32
 cautions about using FORCE with OFFLINE 4-637
 full-capability console
 console 3-27
 defined 3-27

full-capability mode
 cursor on console 3-3
 display screen for, illustration 3-4
 keyboard on console 3-3
 operation on display console 3-7

G
 G command
 See SWAP command
 general trace service 1-27
 GETMAIN/FREEMAIN/STORAGE tracing
 SET command 4-392
 global resource serialization
 removing a system from complex 4-638
 VARY GRS command 4-638
 global resource serialization (GRS)
 resource name list
 changing 4-382
 GMT (Greenwich mean time) 4-357
 GRNAME parameter in APPCPMxx 4-400
 GRS (global resource serialization)
 resource name list
 selecting 4-390
 GRSQ example
 SETGRS 4-411
 GTF (generalized trace facility) 1-26
 starting 4-581

H

halt
 TRACK display 4-597
 HALT command 1-40, 4-295
 complete syntax 4-295
 summary table 4-2
 hard error
 CPU 1-42
 storage 1-45
 hard interruption
 machine check
 controlling recording 4-303
 monitoring 4-303
 hardcopy medium
 changing status 4-629
 controlling in CONSOLEX 3-22
 device, bypassing 3-19
 displaying information 3-40
 message 4-78
 hardcopy message set
 minimum routing codes for 4-630
 selecting a command 3-38
 selecting messages 3-38
 hardcopy processing
 controlling 4-628
 defaults used for 3-23
 varying 4-628
 hardware
 CPU error 1-41
 problem
 recovery 1-40

hardware (*continued*)

 recovery 1-40

hardware configuration definition

 See HCD (hardware configuration definition)

hardware configuration definition (HCD)

 defining a device 3-19

hardware console

 definition 1-2

 to process MVS message traffic 3-19

hardware malfunction 1-31

hardware unit

 deselecting 2-1

 logical reconfiguration 2-2

HCD (hardware configuration definition)

 device definition 3-1

high speed buffer

 error 1-46

highlighting of message 4-172

 displaying options in effect 3-34

hot I/O

 recovery 1-51

hot I/O detection 1-31

I

I command

 See SWITCH command

I/O configuration

 loading 1-5

I/O control command 3-26

I/O device

 error 1-50

 hung

 recovery 1-53

 place online or offline 4-633

I/O interrupt

 hot I/O 1-51

 missing 1-50

IBM Health Checker for z/OS 4-309

ICKDSF (Device Support Facilities) 1-58

ICRF (Integrated Cryptographic Feature)

 CPU with ICRF

 bringing online 2-3

 reconfiguring 2-3

 taking offline 2-4

 removing last ICRF 2-4

IEA101A message 1-8

IEA116A message 4-361

IEA347A message 1-7

IEA886A message 4-357

IEA888A message 4-358

IEA903A message 4-358

IEACMDxx parmlib member 4-507

IEASLPxx parmlib member 4-507

IEASYSxx member of SYS1.PARMLIB 1-8

IEAVMXIT installation exit

 description 3-34

 displaying status 3-34

 status

 activating 4-67, 4-72

 changing 4-67, 4-72

IEAVMXIT installation exit (*continued*)

 status (*continued*)

 displaying 4-67, 4-72

IECIOSxx member

 Parmlib

 I/O device time interval 1-50

IEE094D message 4-359

IEE100E message 4-50

IEE131D message 4-50, 4-51

IEE360I message 1-26

IEE507D message 4-50, 4-51

IEE508E message 4-51

IEE828E message 3-31

IEE922I message 4-77

IEE980I message 1-26

IEF225D message 1-23

IEF238D message 1-32

IEFAUTOS structure

 on coupling facility 1-28

IFASMFDP dump program 1-25

immediate action

 message

 deleting 4-58

 level 4-78

 routing 4-78

immediate action message

 deleting 4-59

IMSI 1-4

inactive console

 characteristic 3-15

 response 3-15

increment

 storage 2-5

indirect address 4-508

indirect addressing

 DATA parameter 4-510

 LIST parameter 4-511

 RANGE parameter 4-511

 REFAFTER parameter 4-510

 REFBEFOR parameter 4-510

 SUMLIST parameter 4-511

 TRDATA parameter 4-511

indirection indicator in indirect addressing 4-508

INFO command authority 3-24

informational

 message

 level 4-78

 routing 4-78

informational command 3-26

informational message

 description 3-31

initialization message suppression indicator (IMSI) 1-4

initialization of system software 1-7

inline commands 1-38

installation exit routine

 description of use in processing a message 3-34

instruction fetch

 SLIP trap 4-520

instruction line

 description 3-4

Integrated Cryptographic Feature (ICRF)
 bringing online 4-47
 taking offline 4-47
intensity
 description 3-3
interception
 of system event 4-506
interruption
 machine check
 controlling recording 4-303
 monitoring 4-303
INTERVAL parameter
 SETSMS command 4-465
IO command authority 3-24
IOACTION command
 summary table 4-2
IOS,MIH command options and functions
 ASM 4-148
IPL (initial program loader) 1-6
 procedure 1-6
 program 1-6
IPS (installation performance specification)
 changing 1-18
IRA200I message 4-349
IRA201I message 4-349
ISG334I 4-410
IXL126I message 4-51
IXL127I message 4-51

J

JES event trace facilities 1-25
JES2
 message routing code 3-30
 procedure for starting 1-9
JES3
 message routing code 3-30
 procedure for starting 1-9
 using MCS console 3-1
job
 canceling 1-22
 controlling 1-22
 displaying active 1-23, 1-24
 passing information 1-22, 4-309
 restart 1-22
 start 1-22
 stop 1-22

K

K command
 See CONTROL command
keyboard A-1
keyboard action 3-7

L

L command
 See LOG command
L= operand
 L= description 3-5

LIBRARY command 4-298
 summary table 4-2
library lookaside
 displaying information 4-160
library lookaside (LLA)
 defined 4-583
 starting LLA address space 4-583
LIMITSIL on DISPLAY command 4-176
LIMMSG parameter in BPXPRMxx 4-432
link pack area expansion 1-24
LLA (library lookaside)
 directory
 modifying 4-332
 stopping address space 4-593
LLA (LNKLST lookaside)
 LNKLST directory
 modifying 4-67
 LLA address space
 starting 4-583
 stopping 4-593
LNKLST concatenation
 display 4-200
LNKLST directory
 modifying 4-67
LNKLST set
 activating 4-449
 locating a routine 4-449
load
 alternate nucleus 1-5
 I/O configuration 1-5
 IML (initial microprogram) 1-2
 parameter field 1-5
 process 1-6
load function 1-6
LOAD parameter 1-7, 1-8
locked keyboard
 response 3-16
LOG command 3-39, 4-299
 complete syntax 4-299
 example 4-299
 summary table 4-2
LOG parameter
 VARY SMS command 4-650
logger
 See system logger
logical reconfiguration
 hardware unit 2-2
LOGOFF command
 complete syntax 4-300
 example 4-300
 summary table 4-2
LOGON command 1-6
 complete syntax 4-301
 LOGON panel 4-301
 summary table 4-2
logrec
 action for a SLIP trap 4-514
 recording by SLIP trap 4-530
LookAt message retrieval tool xviii
loop
 disabled 1-18

lowercase character
use 4-1
LU (logical unit)
base LU 4-398
NOSCHED LU 4-105
system base LU 4-398
LUADD statement in APPCPMxx 4-396
LUDEL command in APPCPMxx 4-400

M

M command
See MOUNT command
machine check
accompanying data 1-41
controlling status
monitoring 4-306
recording 4-306
CPU
hard error 1-42
soft error 1-41
terminating error 1-43
degradation interruption
controlling recording 4-303
displaying status
monitoring 4-308
recording 4-308
ETR (External Time Reference) 1-42
hard interruption
controlling recording 4-303, 4-304
monitoring 4-303, 4-304
interruption
controlling recording 4-303, 4-306
monitoring 4-303, 4-306
monitoring status
controlling recording 4-306
system recovery
controlling recording 4-303
master catalog
specifying an alternate 1-7
MASTER command authority 3-24
master trace 1-26, 4-604
match
for SLIP trap 4-514
MAXCFFEATURELEVEL parameter
SETSMS command 4-463
MCS (multiple console support)
consoles at IPL 1-2
MCS (multiple console support) console
assigning 3-24
use 3-2
MCS console
at IPL 1-2
attribute definition 3-2
defining in CONSOLxx member of
SYS1.PARMLIB 3-2
subsystems' use 3-1
MEMLIMIT on DISPLAY command 4-176
MEMLIMIT on SETOMVS command 4-435

message
automatic deletion
assigning 4-72
controlling format 3-32
deleting 4-58
SYS1.BRODCAST notice section 4-381
error response 3-15
eventual action
deleting 4-59
flagged
deleting 4-62
format 3-6
MPF-processed
controlling 4-382
MPF-suppressed 4-382
number
controlling 4-60
deleting 4-62
operator-to-operator 4-375
processed by an installation exit 3-34
processed by IEAVMXIT 3-34, 4-67
processed by message processing facility
(MPF) 3-32
queue
deleting 4-72
roll rate
setting 4-72
routing
broadcast 4-78
critical action 4-78
critical eventual action 4-78
descriptor code 4-78
immediate action 4-78
informational 4-78
instruction 4-347
message level 4-78
system failure 4-78
write to operator 4-78
SYS1.BRODCAST
saving 4-379
terminal user 4-377
time-sharing user 4-376
message area
description 3-4
message buffer shortage 3-18
message deletion
automatic mode
assigning 4-72
defining 4-72
conversational mode 4-72
nonconversational mode 4-72
roll mode
assigning 4-72
defining 4-72
roll-deletable mode
assigning 4-72
defining 4-72
specification
defining 4-72
displaying 4-72
verifying 4-72

message display
 changing
 color 4-385
 highlighting 4-385
 intensity 4-385
 color
 changing 4-385
 control
 MPFLSTxx member 4-385
 foreign messages
 handling 4-385
 handling
 foreign messages 4-385
 highlighting
 changing 4-385
 message format
 changing 3-32
 controlling 3-6, 3-32
 message identifier 3-6
 message level
 assigning 4-79
 assigning to a console 3-31
 console
 assigning 4-78
 displaying message levels not assigned to any
 console 3-31
 message number
 deleting 4-62
 message presentation
 controlled by MPFLSTxx member 3-32
 displaying options in effect 3-34
 message processing facility (MPF) 4-172
 changing 4-385
 controlling message presentation 3-32
 description 3-32
 displaying status 3-34
 ending message suppression 4-386
 message queue
 deleting 4-72
 message retrieval tool, LookAt xviii
 message roll rate
 setting 4-72
 message routing
 cancelling 4-349
 stopping 4-349
 message routing code 3-30
 message stream console
 defined 3-26
 message stream mode
 deleting messages from a console 3-27
 display screen for, illustration 3-4
 message traffic
 lack 3-15
 MIH
 MIH time options and functions 4-144
 MIH (missing interruption handler)
 time interval
 dynamic changing 4-412
 missing I/O interrupt
 recovery 1-50
 missing interruption handler (MIH)
 time interval
 changing 4-382
 MMS (MVS message service)
 allocated resource
 freeing 4-389
 processing
 ending 4-389
 MN command
 See MONITOR command
 MODE command
 example 4-306, 4-307
 machine check
 interruption 4-303
 machine check interruption
 controlling recording 4-306
 summary table 4-2
 modify
 SLIP trap 4-573
 MODIFY CATALOG command
 example 4-330
 MODIFY command 1-21, 4-309
 BPXOINIT 4-319
 catalog address space
 information 4-321
 service 4-321
 changing resource states 4-337
 DLF
 changing 4-331
 DLF parmlib member
 replacing 4-331
 DLF status
 displaying 4-332
 example 4-67, 4-313, 4-338, 4-341
 external writer
 causing pause 4-340
 specifying data set selection 4-338
 fork() service
 restart 4-314, 4-333
 shut down 4-314, 4-333
 jobname,APPL 4-313
 LLA directory
 replacing 4-332
 OAM
 managing 4-333
 OMVS 4-334
 pass information
 z/OS UNIX System Services application 4-313
 passing information to a job 1-22
 passing information to a job or started task 4-309
 stopping a PFS through an LFS interface 4-336
 stopping a TFS (temporary file system) running in a
 colony address space 4-336
 summary table 4-2
 TSO/VTAM time sharing
 modifying 4-314
 z/OS UNIX initiators
 shut down 4-314, 4-333
 z/OS Unix System Services 4-335
 z/OS UNIX Threads
 terminate 4-314, 4-333

MONITOR command
 complete syntax 4-343
 example 4-344
MSGRT command routing 4-347
 scope in a sysplex 4-343
 summary table 4-2
 to track terminal user 1-21
 tracking a job 1-10
MONITOR command> 4-343
monitoring facility
 recovery
 channel subsystem 1-49
MOUNT attribute 1-30
MOUNT command 1-30
 complete syntax 4-345
 example 4-346
 scope in a sysplex 4-345
 summary table 4-2
MOUNT command> 4-345
mount tape drive
 with a partitionable processor 4-49
move
 online or offline
 central storage sections 4-46
 channel path 4-46, 4-50
 processor 4-46
MPFLSTxx member of SYS1.PARMLIB 3-34
MR command
 See MSGRT command
MSGRT command 4-347
 command routing
 CONFIG 4-347
 CONTROL 4-347
 MONITOR 4-347
 STOPMN 4-347
 complete syntax 4-347
 example 4-348
 extended MCS console
 limitation 4-347
 message routing
 cancelling 4-349
 stopping 4-349
ROUTE *ALL and ROUTE systemgroupname commands
 limitation 4-347
routing instruction
 removing 4-347
status display
 routing 4-347
multiple-console
 configuration 3-2
 support
 for MCS console 3-2
multiprocessor
 non-partitionable
 RSU parameter 2-8
RSU parameter
 recommended value 2-8
terminating error
 CPU 1-43

MVS message service (MMS)
 changing 4-389
parameter
 changing 4-382
 SET command 4-389
 starting 4-389
 stopping 4-389

N

NAME parameter
 in PROGxx parmlib member 4-447
NAMEIN on DISPLAY command 4-175
NIP (nucleus initialization program) completion 3-18
NIP console 1-7
 definition 1-2
no-match
 for SLIP trap 4-514
NODUMP dump mode 4-25
nonconversational mode
 enter command with selector pen 3-12
 entering command assigned to PFK 3-11
 message deletion 4-72
 cancelling 4-73
PFK 4-70
 PFK (program function key) 3-35, 3-36
NOSCHED parameter in APPCPMxx 4-397
NQN/NONQN parameter in APPCPMxx 4-400
nucleus
 alternate
 loading 1-5
 identifier 1-5
 primary 1-5
 standard 1-5

O

OAM (object access method)
 managing 4-333
 starting 4-583
 starting OAM 4-583
 starting OAM address space 4-583
OAM (Object Access Method)
 stopping address space 4-594
OAM address space
 stopping 4-594
offline
 varying console 3-19
offline device 3-1
OLTEP (online test executive program) 1-30
online device 3-1
operations log
 purpose 3-40
operator 1-6
operator information area
 description 3-4
operator load function 1-6
operator-to-operator
 communication 4-375
 message 4-375
OPERLOG (operations log) 3-40

options for CHNGDUMP command 4-29
 OPTIONSIO on DISPLAY command 4-176
 out of line display area
 specification
 altering 4-57
 changing 4-57
 displaying 4-57
 out-of-sync condition
 3380
 recovery 1-57
 recovery 1-58
 output-only mode
 defined 3-26
 OVER dump mode 4-25
 OWNERIO on DISPLAY command 4-176

P

P command
 See STOP command
PA command
 See PAGEADD command
page data set
 adding 4-349
 deleting 4-351
 draining 4-351
 replacing 4-351
PAGEADD command 4-349
 complete syntax 4-350
 example 4-350
 summary table 4-2
PAGEDEL command 4-351
 auxiliary storage
 deleting 4-351
 draining 4-351
 replacing 4-351
 complete syntax 4-352
 example 4-352
 summary table 4-2
PAGNUM system parameter 4-349
parallel access volume
 boxed alias of a parallel access volume 1-32
parameter
 job entry subsystem 1-2
 on SLIP command 4-515
parmlib
 ADYSETxx member 4-385
 APPC/MVS member
 changing 4-382
 ASCH member
 changing 4-382
 CEEPRM member
 changing 4-382
 CNGRP member
 changing 4-382
 CONSOLxx member
 AMRF parameter 4-63
 AREA parameter 4-57
 CON parameter 4-72
 CONSOLE statement MONITOR
 parameter 4-343

parmlib (continued)
 CONSOLxx member (*continued*)
 DEL parameter 4-72
 INIT statement MONITOR parameter 4-343
 LEVEL parameter 4-78
 LOGLIM parameter 4-64
 MFORM parameter 4-72
 MLIM parameter 4-64
 PFK parameter 4-386
 RLIM parameter 4-64
 RNUM parameter 4-72
 RTME parameter 4-72
 SEG parameter 4-72
 UEXIT parameter 4-67
 USE parameter 4-77
 UTME parameter 4-76
 DIAGxx member
 selecting through DIAG=xx 4-392
 GRSRNLxx member
 selecting GRS resource name list 4-390
 IEAOPTxx member 4-385
 IECIOSxx member 4-412
 IGDSMSxx member
 selecting through SMS=xx 4-386
 MMSLSTxx member
 selecting through MMS=xx 4-389
 PFKTABxx member
 changing 4-386
 PROGxx member 4-441
 selecting through PROG=xx 4-391
 SMFPRMxx member 4-385, 4-456
Parmlib IECIOSxx member
 I/O device time interval 1-50
path
 varying online or offline 4-639
PD command
 See PAGEDEL command
PER (program event recording) event
 trap 4-514
PER trap
 dynamic PER traps 4-518
 monitor with checkpoint/restart 4-518
 set 4-515
performance group 4-362
PFK (program function key)
 assigning 4-71
 command
 changing 4-70, 4-71
 defining 4-70, 4-71
 conversational mode 4-70
 defining a command 3-35, 3-36
 definition error 3-9
 display command defined for 3-9
 display line on console 3-4
 display on 3277-2 display station 3-12
 displaying number
 3277-2 display station 4-60
 in conversational mode 3-35, 3-36
 initial definition 3-9
 keyboard 3-3
 nonconversational mode 4-70

PFK (program function key) (*continued*)

 system installation definition 3-9

 table

 assigning 4-71

 defining 4-70

 use to enter command 3-9

PFK (program function key) parameter 3-37

PFK (program function key) table

 defining 3-37

 definition 3-35, 3-37

 description 3-37

PFK (program function key) table 3-37

PFK display line

 example 3-4

PFK table

 display definition 3-9

PFKTABxx member of **SYS1.PARMLIB**

 changing 3-37

 description 3-37

 example 3-10

PFK (program function key) parameter 3-37

PFS (physical file system)

 stopping a PFS through an LFS interface 4-336

physical reconfiguration

 hardware unit 2-2

physically partitioned mode

 RSU parameter

 recommended value 2-8

PID on **DISPLAY** command 4-175

place

 channel path online or offline 4-49

 processor online or offline 4-47

 storage online or offline 4-47, 4-48

pointer in indirect addressing 4-508

PPT (program properties table)

 effects of the **SET SCH** command 4-391

 modifying 4-391

previous command

 retrieve 3-7

primary nucleus 1-5

primary SMF data set 1-25

print the system log 3-39, 3-40

procedure

 started task 1-24

process resource / system manager (PR/SM)

SETXCF command 4-491

 XCF PR/SM policy 4-491

processor

 placing online or offline 4-47

 reconfiguring 4-47

product enablement policy

 changing 4-382, 4-392

 display 4-195

program function keyboard

 description 3-3

PRSMOPOLICY option

SETXCF command 4-491

PSTIMER parameter in **APPCPMxx** 4-398

PSW (program status word) 1-6, 1-18

PT command

 See **STOPTR** command

Q

quiesce

 global resource serialization system 4-639

 system 1-39

QUIESCE command 1-18, 1-39

 complete syntax 4-354

 summary table 4-2

quiesce status

 meaning 4-646

 data set 4-649

DFSMStvs instance 4-649

R

R command

 See **REPLY** command

REALSTORAGE parameter in **CUNUNIxx** 4-481

reconfiguration

 central processor 2-2

 central storage 2-4, 2-10

 channel path 2-10

 coupling facility 2-11

 failing device

 recovery 1-54

hardware unit

 adding 2-1

 logical 2-2

 physical 2-2

 removing 2-1

I/O device 2-11

ICRF (Integrated Cryptographic Feature) 2-3

processor type 2-2

reconfigure

 channel path 4-49

 DDR (dynamic device reconfiguration) 1-39

I/O device 4-633

I/O path 4-639

processor 4-47

storage 4-47, 4-48

record

 system event 1-26

recovery

 action 1-59

 error 1-29

 from inactive console condition 3-15

 hardware 1-40

 hot I/O 1-31

 operating system 1-40

Recovery

I/O

 hung 1-53

registered product

 display 4-195

remove

 channel path 4-50

 last path to allocated, TP, in-use, or reserved

 device 4-50

 last path to allocated, TP, in-use, reserved device, or

 coupling facility 4-50

 last path to coupling facility 4-50

remove (*continued*)

 tape or DASD volume 4-615

REPLY command 4-355

 complete syntax 4-356

 dump option

 specifying 4-263

 example 4-356

 scope in a sysplex 4-356

SET command 4-357

 short form 4-356

 specifying system symbols 4-355

 summary table 4-2

 to invoke automatic restart 1-23

reserve a volume 1-30

reset

 dump options to initial value 4-32

 dump type to ADD 4-32

 system dump options list 4-26

RESET command 4-362

 changing performance group 4-363

 changing service class 4-363

 example 4-365

 forcing hung MCS console offline 4-362

 quiesce problem job or address space 4-363

 resume quiesced job or address space 4-363

 scope in a sysplex 4-362

 summary table 4-2

RESET on DISPLAY command 4-177

RESET parameter

 SET command 4-385

resource allocation 1-29

resource states

 changing with MODIFY command 4-337

resources

 dump

 adding 4-281

 deleting 4-285, 4-286

response

 blank screen 3-16

 console message backup 3-16

 error message 3-15

 hardware error 3-15

 inactive console condition 3-15

 locked keyboard 3-16

 WTO buffer shortage 3-16

 WTOR buffer shortage 3-18

restart

 automatic 1-23

 deck 1-23

 deferred 1-23

 global resource serialization ring 4-639

 job 1-22

 SMF (system management facilities) 1-26

 system 1-19

 using 1-18

RESTART key 1-19

restore

 screen after hardware error 3-16

restrictions

 Define PC command 4-367

 DFSMStvs, VARY SMS command 4-649

restrictions (*continued*)

 DISPLAY SMS command

 LISTDRI keyword 4-220

 GRSRNL= parameter 4-390

retained action

 message

 deleting 4-58

retained immediate

 message

 deleting 4-58

retention facility

 action message

 activating 4-63

 cancelling 4-63

 changing 4-63

 deactivating 4-63

 deleting 4-59

 displaying status 4-63

retrieve previous command 3-7

RLSABOVETHEBARMAXPOOLSIZE parameter

 SETSMS command 4-464

RLSFIXEDPOOLSIZE parameter

 SETSMS command 4-464

RLSINIT parameter

 SETSMS command 4-464

RMAX value

 effect on short form of REPLY command 4-356

roll mode

 message deletion

 assigning 4-72

 defining 4-72

roll-deletable mode

 message deletion

 assigning 4-72

 defining 4-72

roll-deletable mode of message deletion

 on message stream console 3-27

ROUTE command

 complete syntax 4-367, 4-370

 example 4-372

 specifying system symbols 4-369

 summary table 4-2

 system-to-system

 directing 4-367

routing

 message 3-30

 by routing code 3-30, 4-620

routing code

 assigning to a console 3-30, 4-620

 description 3-30

 displaying routing codes not assigned to any

 console 3-31

 for hardcopy message set 4-630

routing codes 4-115

 listing 3-30

rRequirements

 SETSMS parameter 4-458

RRS (resource recovery services)

 starting 4-584

RSU parameter

 example 2-7

RSU parameter (*continued*)
recommended value 2-8

S

S command
 See START command
SCDS parameter
 SETSMS command 4-460
SCCHED parameter in APPCPMxx 4-397
SCHEDEXX parmlib member 4-604
scratch volume 1-30, 1-31
screen
 erasing 4-62
screen area
 L= operand 3-5
 on console of different operating mode 3-3
screen character
 description 3-6
SD command
 See SETDMN command
SDUMP dump type 4-25
SE command
 See SEND command
secondary console
 placing online or offline 4-631
secondary SMF data set 1-25
selector pen
 defining a command 3-35
 description 3-3
 enter command 3-12
 enter command in conversational mode 3-12
 enter command in nonconversational mode 3-12
 use 3-7
send
 message to console 4-620
SEND command
 example 4-1, 4-376, 4-379, 4-381
 for time-sharing communication 1-21
operator-to-operator
 communication 4-375
 message 4-375
scope in a sysplex 4-375
summary table 4-2
terminal user
 communication 4-377
 message 4-377
time-sharing user
 communication 4-376
 message 4-376
SEND command> 4-375
SER on DISPLAY command 4-177
service class 4-362
service processor
 damage
 recovery 1-44
 permanent failure 1-44
 temporary failure 1-44
set
 dump mode and option 4-34
SET command 4-382

SET command (*continued*)
 complete syntax 4-384
 DIAG parameter 4-392
 example 3-33, 3-36, 4-393
 GRSRNL parameter 4-390
 MEMLIMIT operand 4-435
 MMS parameter 4-389
 PROG parameter 4-391
 relationship to CONSOLE statement in
 CONSOLOxx 3-21
 REPLY command 4-357
 RESET parameter 4-385
 RLS_MAXCFFEATURELEVEL parameter 4-389
 scope in a sysplex 4-383
 SMS parameter 4-386
 summary table 4-2
SET SMS command
 AKP parameter 4-387
set time and date 1-18
SETAPPC command 4-396
 See also SETCON command
 complete syntax 4-396
SETCEE command
 complete syntax 4-402
 example 4-402
SETCON command 4-403
 See also SETCON command
 complete syntax 4-403
 summary table 4-2
SETDMN command 4-405
 complete syntax 4-405
 example 4-406
 summary table 4-2
SETETR command
 complete syntax 4-407
 enabling 4-407
 example 4-407
 PORT=n 4-407
 summary table 4-2
SETGRS
 ENQMAXA 4-411
 ENQMAXU 4-410
 GRSQ 4-411
SETGRS command
 CNS example 4-410
 complete syntax 4-408
 example 4-410
 MODE=STAR 4-408
 summary table 4-2
SETIOS command
 complete syntax 4-412
 example 4-417
 summary table 4-2
SETLOAD command
 complete syntax 4-420
 example 4-420
 summary table 4-2
SETLOGR command
 complete syntax 4-422
 example 4-424
 summary table 4-2

SETLOGRC command
 changing the recording medium 4-425
 summary table 4-2
SETOMVS command 4-427
 syntax 4-427
SETPROG command
 summary table 4-2
 syntax 4-441
SETPROG EXIT command
 example 4-443, 4-446
SETPROG LNLST command
 example 4-450
SETRRS CANCEL command
 complete syntax 4-454
 example 4-454
SETRRS command
 summary table 4-2
SETRRS SHUTDOWN command
 complete syntax 4-454
 example 4-455
SETSMF command
 complete syntax 4-456
 example 4-456
 summary table 4-2
SETSMS command
 ACDS parameter 4-459
 AKP parameter 4-459
 BMFTIME parameter 4-465
 CACHETIME parameter 4-465
 comparison with SET SMS= 4-458
 complete syntax 4-457
 DINTERVAL parameter 4-465
 DSSTIMEOUT parameter 4-465
 INTERVAL parameter 4-465
 MAXCFFEATURELEVEL parameter 4-463
 parameter
 incorrect combination 4-458
 SCDS parameter 4-460
 scope in a sysplex 4-459
 summary table 4-2
SETSSI command
 ACTIVATE parameter 4-473
 ADD parameter 4-472
 CONSNAME parameter 4-474
 DEACTIVATE parameter 4-473
 INITPARM parameter 4-474
 INITRTN parameter 4-474
 SUBNAME parameter 4-473
 summary table 4-2
 syntax 4-472
SETUNI command
 summary table 4-2
 syntax 4-475
SETXCF command
 controlling XCF 4-483
 PRSMPOLICY option 4-491
 scope in a sysplex 4-483
 summary table 4-2
 syntax 4-483
 XCF PR/SM policy 4-491
 shared device
 operator action 1-55
 recovery 1-55
SHCDS parameter
 DISPLAY SMS command 4-222
shortcut keys A-1
shorthand for base/displacement in SLIP
 command 4-509
SIDEINFO statement in APPCPMxx 4-401
single-image mode
 RSU parameter
 recommended value 2-8
SL command
 See **SLIP command**
SLIP command 4-506
 complete syntax 4-506
 error event SLIP trap 4-519
 IF SBT PER trap 4-520
 indirect addressing 4-508
 non-PER trap 4-519
 parameter relationship 4-515
 SA PER trap 4-521
 SAS PER trap 4-521
 set effective PER trap 4-515
 SET example 4-569
 summary table 4-2
 syntax for a PER trap 4-520
SLIP trap
 instruction fetch 4-520
 SAS PER 4-521
 storage alteration 4-521
 storage alteration by STURA 4-521
 successful branch 4-520
SMF (system management facilities) 4-215
SMF (system management facility)
 option
 changing 4-360
 replying 4-360
SMS (storage management subsystem)
 ACDS
 replacing 4-457
 active configuration
 saving 4-457
 BMF record 4-465
 cache control unit 4-465
 CF processing 4-642
 changing status of storage group or volume 4-642,
 4-646
 Display PDSE monitor state 4-646
 Modifying processing of PDSE monitor 4-645
 new configuration
 activating 4-457
 PDSE subsystem 4-645
 placing a system-managed tape library online or
 offline 4-645
 Releasing PDSE latches 4-645
 restarting 4-386
 starting 4-386
 synchronization interval
 changing 4-457
 trace option 4-466

SMS (storage management subsystem) (*continued*)

 varying an optical drive online or offline 4-645

 varying an optical library online or offline 4-645

SMS (Storage Management Subsystem)

 changing status of DFSMStvs 4-648

SMS parameter

 DISPLAY SMS command 4-218

SMVSAM SPHERE parameter

 VARY SMS command 4-651

soft error

 CPU 1-41

 storage 1-45

SOM

 START 4-585

special screen character

 description 3-6

SRM (system resource manager)

 parameter

 changing 4-382

SSI parameter

 DISPLAY SSI command 4-233

standard nucleus 1-5

start

 Advanced Program-to-Program Communication/MVS

 (APPC/MVS) 4-579

 ASCH (APPC/MVS transaction scheduler) 4-580

 DLF (data lookaside facility) 4-587

 external writer 4-588

 GTF (generalized trace facility) 4-581

 job 4-575

 LLA address space 4-583

 OAM (object access method) 4-583

 RRS 4-584

 starting 4-579, 4-580

 starting APPC/MVS address space 4-579

 starting APPC/MVS transaction scheduler 4-580

 terminal control address space (TCAS) 4-586

 TSO/Virtual Telecommunications Access Method

 (TSO/VTAM) 4-586

 VLF (virtual lookaside facility) 4-587

START command 1-21, 1-22, 1-26, 4-575

 example 4-579, 4-582, 4-585, 4-589

 summary table 4-2

START RRS command

 complete syntax 4-584

 example 4-585

started task

 coding 1-24

 controlling 1-24

 modifying 1-24

 passing information 1-22, 4-309

 starting 1-24

 starting from console 4-575

started tasks

 displaying status 4-158, 4-159

status command

 controlled by TRACK command 4-613

status display

 console-to-console

 routing 4-347

 status display (*continued*)

 console-to-console routing

 controlling 4-347

 controlled by CONFIG command 4-46

 controlled by DISPLAY command 1-10

 controlling 3-26

 halting 4-60

 printing

 halting 4-60

 updating 4-60

status display console

 defined 3-26

status display mode

 display screen for, illustration 3-4

 setting up a console 3-26

STATUS parameter

 DISPLAY SMS command 4-222

STEPNAME parameter

 SETSMS command 4-468

stop

 I/O on a channel path quickly with FORCE operand

 of CONFIG 4-50

 monitoring of terminal use 1-21

 OAM address space 4-594

 system 1-40

 TFS (temporary file system) running in colony

 address space 4-595

 TRACK display 4-597

 VLF (virtual lookaside facility) 4-595

STOP command 1-21, 1-22, 1-26, 4-590

 complete syntax 4-593

 example 4-592

 summary table 4-2

stop routine 1-22

STOPMN command

 complete syntax 4-596

 example 4-596

 MSGRT command routing 4-347

 stopping monitoring of terminal use 1-21

 summary table 4-2

STOPTR command 4-597

 complete syntax 4-597

 example 4-597

 scope in a sysplex 4-597

 summary table 4-2

storage

 central

 reconfiguration 2-4

 error 1-45

 hard 1-45

 soft 1-45

 placing online or offline 4-47, 4-48

 reconfiguring 4-47, 4-48

storage element

 size

 ES/3090 processor complex 2-6

storage group

 varying status 4-647

storage increment

 size 2-5

Storage Management Subsystem (SMS)
 displaying information 4-215
 storage sub-increment
 size 2-5
 sub-increment
 storage 2-5
 subchannel
 recovery 1-49
 subsystem cache 4-349
 successful branch
 SLIP trap 4-520
 summary of system command 4-2
 SUMMARYIS on DISPLAY command 4-175
 SWAP command 1-39, 4-599
 example 4-600
 summary table 4-2
 SWAP operator command
 failing device
 recovery 1-54
 swapping device of unlike density 1-39
 SWITCH
 varying online or offline 4-658
 SWITCH command 1-26, 4-601
 complete syntax 4-601
 scope in a sysplex 4-601
 summary table 4-2
 switching workload management modes
 controlling 1-21
 symbolic group name 1-31
 symbolic links
 in a sysplex 4-440
 symbols
 in commands 1-14
 syntax conventions
 how to read 4-12
 SYS command authority 3-24
 SYS1.BRODCAST 4-375
 contents 1-21
 message
 saving 4-379
 notice section
 deleting message 4-381
 listing 4-379
 SYS1.BRODCAST data set
 managing with the SEND operator command 4-375
 SYS1.DUMPxx 4-280
 SYS1.NUCLEUS 1-7
 SYS1.PARMLIB
 CONSOLxx member
 MSGRT parameter 4-347
 SYSABEND dump type 4-25
 SYSLOG (system log) 3-18, 3-39
 SYMDUMP dump type 4-25
 sysplex
 BPXPRMxx VERSION statement 4-440
 scope
 CHNGDUMP command 4-26
 CONTROL command 4-57
 definition 1-11
 DISPLAY command 4-99
 MONITOR command 4-343

sysplex (*continued*)
 scope (*continued*)
 MOUNT command 4-345
 REPLY command 4-356
 RESET command 4-362
 SEND command 4-375
 SET command 4-383
 SETSMS command 4-459
 SETXCF command 4-483
 STOPTR command 4-597
 summary of commands 1-12
 SWITCH command 4-601
 TRACK command 4-613
 UNLOAD command 4-615
 VARY command 4-617
 SETOMVS AUTOMOVE paramater 4-430
 SETOMVS FILESYS paramater 4-430
 SETOMVS FILESYSTEM paramater 4-430
 SETOMVS FROMSYS parameter 4-430
 SETOMVS MOUNTPOINT parameter 4-430
 SETOMVS SYSNAME parameter 4-439
 sysplex failure management
 DISPLAY XCF command 4-256
 switching couple data containing sysplex failure
 management information 4-485
 sysplex scope
 CHNGDUMP command 4-26
 CONTROL command 4-57
 definition 1-11
 DISPLAY command 4-99
 MONITOR command 4-343
 MOUNT command 4-345
 REPLY command 4-356
 RESET command 4-362
 SEND command 4-375
 SET command 4-383
 SETSMS command 4-459
 SETXCF command 4-483
 STOPTR command 4-597
 summary of commands 1-12
 SWITCH command 4-601
 TRACK command 4-613
 UNLOAD command 4-615
 VARY command 4-617
 system
 control command 3-26
 controlling 1-9
 dump options list 4-25
 error condition 3-15
 information recording
 controlling 1-25
 IODF (I/O definition file) 1-4
 IPL volume 1-3
 monitoring 1-29
 operator frame 1-19
 record activity 3-38
 recovery
 controlling recording 4-306
 machine check interruption 4-306
 recovery processing 3-19
 residence volume 1-3, 1-6, 1-39

system (*continued*)

 response, lack 3-15

 restart 1-19

 stop 1-40

 trace 1-26

system command group 3-26

See also command group

 assigning 4-621, 4-627

system console

 definition 1-2

system control frame 1-3

system failure

 message

 level 4-78

 routing 4-78

system information request

 replying 4-356

 responding 4-356

system log

 control 3-38

 displaying information 3-31

 message appearing 3-31

 printing 4-665

 purpose 3-39

 queuing 4-665

 starting 4-665

 stopping 4-665

system logger

 DISPLAY XCF command 4-256

 displaying information 4-161

 must be active for START RRS command 4-584

 switching couple data containing system logger

 information 4-485

system management facilities (SMF)

 description 1-25

system message

 controlling 3-29

system parameter 1-18

 entering 4-360

 replying 4-360

 specifying 4-360

system parameter change 1-18

system parameters 1-8

system recovery

 WTOR message

 replying 4-357

system request

 WTOR message

 replying 4-357

System Security WTORS

 replying 4-357

system symbols

 in commands 1-14

system task

 starting from console 4-575

system trace 4-604

system-initiated DDR 4-600

 switching a data set 4-601

system-to-system

 ROUTE command

 directing 4-367

SYSUDUMP dump type 4-25

T

T command

See SET command

tape

 blank 1-30

 label 1-30

 label verification 4-346

 unlabeled 1-30

tape device

 define automatically switchable 4-632

 in a sysplex 1-27

tape drive

 mounting with a partitionable processor 4-49

tapemark 1-30

TCAS (terminal control address space) 4-314

terminal control address space (TCAS)

 start 4-586

terminal user

 communication 4-377

 message 4-377

terminate

 job 4-19

 MOUNT command 4-19

 time-sharing user 4-19

 z/OS UNIX process 4-19

terminating error

 CPU 1-43

 multiprocessor 1-43

TFS (temporary file system)

 stopping TFS running in colony address

 space 4-336, 4-595

time

 setting 4-357

time interval

 IBM-supplied default value

 changing 1-50

time zone constant 4-385

time-sharing user

 communication 4-376

 message 4-376

TOD (time-of-day) clock

 setting 1-8, 4-357

TPDATA parameter in APPCPMxx 4-399

TPLEVEL parameter in APPCPMxx 4-399

TR command

See TRACK command

trace

 action for a SLIP trap 4-514

 component 1-27

 master 1-26

 SMS trace option 4-466

 system 1-26

TRACE command 1-27

 complete syntax 4-603

 example 4-608

 summary table 4-2

trace table 1-26

TRACK command 4-613

TRACK command (*continued*)
 complete syntax 4-613
 example 4-614
 scope in a sysplex 4-613
 summary table 4-2
 TRANVSAM parameter
 VARY SMS command 4-649
 TRANVSAM PEERRECOVERY parameter
 VARY SMS command 4-651
 trap
 of system event 4-506
 TSO/E (Time Sharing Option/Extension) 1-21
 controlling 1-21
 TSO/Virtual Telecommunications Access Method (TSO/VTAM)
 start 4-586
 TSO/VTAM
 time sharing
 modifying 4-314
 TSO/VTAM (Time Sharing Option/Virtual Telecommunications Access Method)
 modifying 1-21
 TYPEIT on DISPLAY command 4-176

U

U command
 See UNLOAD command
 U on DISPLAY command 4-175
 unconditional reserve
 recovery 1-55
 uniprocessor
 RSU parameter
 recommended value 2-8
 UNIX System Services
 switching couple data containing UNIX System Services information 4-485
 unlabeled tape 1-30
 unload
 tape or DASD volume 4-615
 UNLOAD command 1-30, 4-615
 complete syntax 4-615
 scope in a sysplex 4-615
 summary table 4-2
 uppercase character
 use 4-1
 USE attribute 1-30
 USERVER parameter in APPCPMxx 4-400
 UTC (Coordinated Universal Time) 4-357

V

V command
 See VARY command
 value compare in SLIP
 definition 4-540
 vary
 console 4-620
 console offline 3-19
 devices defined as 3270 model X 3-19
 global resource serialization system 4-638

vary (*continued*)
 I/O device 4-633
 I/O path online or offline 4-639
 secondary console 4-631
 SMS (DFSMStvs) 4-648
 SMS (storage group or volume) 4-642, 4-646
 tape device as automatically switchable 4-632
 VARY CN command
 description 4-621
 VARY command 1-32, 4-616
 complete syntax 4-665
 CONSOLE
 example 3-30
 example 4-627, 4-631, 4-632, 4-641, 4-662, 4-663, 4-664
 HARDCOPY
 example 3-38, 3-39, 3-40
 limitations with extended MCS console 4-617
 reconfiguration 2-1
 relationship to HARDCOPY statement in CONSOLxx 3-22
 scope in a sysplex 4-617
 SMS (storage management subsystem) 4-642, 4-646
 SMS (storage management subsystem) Display PDSE monitor state 4-646
 SMS (storage management subsystem) Modifying processing of PDSE monitor 4-645
 SMS (storage management subsystem) PDSE subsystem 4-645
 SMS (storage management subsystem) Releasing PDSE latches 4-645
 SMS (storage management subsystem) storage group or volume 4-645
 SMS placing a system-managed tape library online or offline 4-645
 summary table 4-2
 SWITCH 4-658
 VARY CN,AUTOACT= 4-620
 VARY CONSOLE command
 description 4-626
 VARY SMS command
 CICSVR 4-657
 CICSVR,RCDS 4-658
 DFSMStvs 4-648, 4-649
 LOG parameter 4-650
 SMVSAM SPHERE parameter 4-651
 TRANVSAM parameter 4-649
 TRANVSAM PEERRECOVERY parameter 4-651
 VATLSTxx parmlib member 1-30
 virtual lookaside facility (VLF)
 starting virtual lookaside facility (VLF) 4-587
 Virtual Telecommunications Access Method (VTAM) 4-309
 TCAS
 modifying 4-314
 VIRTUAL=REAL (V=R) 1-23
 VLF (virtual lookaside facility)
 defined 4-587
 starting 4-587
 stopping 4-595

VLF (virtual lookaside facility) (*continued*)
 stopping VLF (virtual lookaside facility) 4-595
VOLSELMSG parameter
 SETSMS command 4-467
volume
 mounting 1-30, 4-345
 varying volume status 4-647
VSERVERIV on DISPLAY command 4-176

W

W command
 See WRITELOG command
wait state
 not valid 1-18
 restartable 1-18, 1-19, 1-31
warning line
 description 3-4
wildcard
 * (asterisk)
 specified on identifier of system command 4-310
 using in operator commands 1-17
WLM (workload management)
 DISPLAY XCF command 4-256
 switching couple data containing WLM
 information 4-485
WRITELOG command 3-39, 4-665
 complete syntax 4-665
 example 4-665
 summary table 4-2
 to force printing of SYSLOG 3-40
WTL (write to log)
 buffer 4-64
 SYSLOG buffer 4-64
WTL buffer
 number allowed
 changing 4-64
WTL macro 3-39
WTL SYSLOG buffer
 number allowed
 changing 4-64
 displaying 4-64
WTO (write to operator)
 buffer
 shortage 4-72
 message
 level 4-78
 routing 4-78
 message buffer 4-64
WTO (write to operator) buffer 3-16
WTO (write-to-operator) buffer
 shortage 3-16
WTO installation exit
 IEAVMXIT
 changing 4-67
 displaying status 4-67
WTO message
 buffer limit 3-16
 controlled by installation exit IEAVMXIT 3-34

WTO message buffer
 number allowed
 changing 4-64
 displaying 4-64
WTOR (write to operator with reply)
 buffer 4-64
WTOR (write to operator with reply) buffer
 shortage 3-16
WTOR message
 buffer limit 3-16
WTOR message buffer
 number allowed
 changing 4-64
 displaying 4-64

X

XCF (cross-system coupling facility)
 switching couple data set containing XCF
 information 4-485
XCF PR/SM policy
 SETXCF command 4-491

Z

Z command
 See HALT command
z/OS UNIX System Services
 dynamically activating maintenance 4-335

Readers' Comments — We'd Like to Hear from You

**z/OS
MVS System Commands**

Publication No. SA22-7627-15

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send your comments via e-mail to: mhvrcfs@us.ibm.com

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

E-mail address

Readers' Comments — We'd Like to Hear from You
SA22-7627-15



Cut or Fold
Along Line

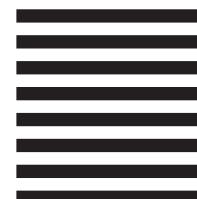
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P181
2455 South Road
Poughkeepsie, NY
12601-5400



Fold and Tape

Please do not staple

Fold and Tape

SA22-7627-15

Cut or Fold
Along Line



Program Number: 5694-A01, 5655-G52

Printed in USA

SA22-7627-15

