z/OS

MVS System Commands

SA22-7627-03

z/OS

MVS System Commands

SA22-7627-03

Note

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

Fourth Edition, June 2002

This is a revision of SA22-7627-02.

This edition applies to Version 1 Release 3 of z/OS (5694-A01), and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

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

International Business Machines Corporation
Department 55JA, Mail Station P384
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: mhvrcfs@us.ibm.com
World Wide Web: http://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 book
- 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, 2002. 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 Book xvii
Who Should Use This Book xvii
How To Use This Book. xvii
Where To Find More Information xvii
Using LookAt to look up message explanations. xvii
Accessing licensed books on the Web 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
Setting the Time and Date 1-8
Starting and Specifying Parameters for the Job Entry Subsystem 1-8
Controlling the System 1-9
Displaying Current System Status. 1-9
Displaying the Status of Devices and Availability of Paths 1-10
Communicating Among Several Consoles 1-10
Sending Commands to Systems in a Sysplex 1-10
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 1-19
Activating a Workload Management Service Policy 1-20
Switching Workload Management Modes 1-20
Controlling Time-Sharing 1-21
Controlling Jobs 1-21
Starting a Job 1-22
Stopping a Job 1-22
Cancelling a Job 1-22
Passing Information to a Job 1-22
Modifying a Job's Performance Group 1-22
Modifying a Job's Service Class 1-22
Restarting a Job 1-22
Deferred Restart 1-23
Controlling Started Tasks 1-25
Controlling System Information Recording 1-25
System Management Facilities 1-25
System Trace 1-26
The Generalized Trace Facility 1-26
Master Trace 1-27
Component Trace 1-27
Logrec Recording Medium 1-27

© Copyright IBM Corp. 1988, 2002

Controlling Automatic Tape Switching 1-27
Defining Automatically Switchable Devices 1-29
Displaying Information About Automatically Switchable Devices 1-29
Interacting with System Functions 1-30
Device Allocation 1-30
Hot I/O Detection 1-32
Device Boxing 1-33
Command Flooding 1-34
Class M1 Commands 1-34
Class M2 Commands 1-34
Class C1 Commands 1-36
Class C2 Commands 1-36
Inline Commands 1-36
Responding to Failing Devices 1-37
Quiescing the System 1-38
Stopping the System 1-38

Chapter 2. Console Characteristics and Operations 2-1
General Characteristics of Display Consoles 2-1
Subsystem Use of Consoles. 2-1
Multiple-Console Configuration 2-2
Features on Display Consoles 2-3
Display Screen Areas 2-4
L= Operand 2-5
Special Screen Characters 2-6
Messages Sent to Display Consoles 2-6
Operations on Display Consoles in Full-Capability Mode 2-7
Performing Basic Keyboard Actions 2-7
How to Enter Commands 2-7
Entering Commands with the Keyboard 2-7
Entering Commands with Program Function Keys 2-9
Entering Commands with the Selector Pen 2-12
Changing Information in the Entry Area 2-13
Blanking the Entry Area 2-15
Handling Consoles in Error Conditions 2-15
System Errors 2-15
Console Hardware Errors 2-16
Responding to a Master Console Failure. 2-17
Responding to a No-Consoles Condition 2-18
Recovering Lost Messages. 2-19
Responding to Console Message Backups 2-19
Processing MVS Messages at the System Console During System Recovery 2-22
Placing a Console in Offline Status 2-23
Interchanging Your Consoles on a Control Unit 2-23

Chapter 3. Defining and Changing Console Characteristics 3-1
Using Operator Commands to Change CONSOLxx Statements 3-1
Potential Effects of Altering Console Attributes 3-1
The CONSOLE Statement of CONSOLxx 3-1
The INIT Statement in the CONSOLxx Member 3-2
The HARDCOPY Statement in the CONSOLxx Member 3-3
Displaying Information About Console Characteristics 3-4
Changing Console Characteristics 3-5
System Commands Grouped According to System Command Authority 3-5
Changing the Authorization of a Console 3-7

Changing the Master Console in a System or Sysplex 3-7
Console Groups 3-8
Switching Console Attributes from One Console to Another 3-9
Defining Console Use 3-10
Using a Full-Capability Console 3-11
Controlling System Messages and Commands 3-13
Defining Routing Codes for a Console 3-13
Defining Message Levels for a Console 3-14
Controlling the Format of Messages 3-16
Controlling the Message Processing Facility (MPF) 3-16
Controlling the Action Message Retention Facility 3-17
Activating WTO and WTOR Installation Exit Routines 3-18
Checking Message Processing, Retention, and Presentation Options 3-18
Defining Program Function Keys (PFKs) 3-19
Defining PFKs Using PFK Tables 3-19
Defining PFKs Using the CONTROL Command 3-20
The PFKTABxx and PFKs 3-21
Processing Hardcopy 3-22
The Hardcopy Message Set 3-22
The Hardcopy Medium 3-23

Chapter 4. MVS System Commands Reference 4-1
Command Syntax Notation 4-12
System Command Formats 4-13
Typical Format 4-13
A Second Format 4-13
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 SYSMDUMP 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 4-32
Setting the Dump Modes and Options 4-40
CMDS Command 4-43
Syntax 4-43
Parameters 4-43
CONFIG Command 4-45
Syntax 4-45
Reconfiguring the System Directly 4-46
Reconfiguring the System with a CONFIGxx Parmlib Member 4-54
Reconfiguring the System in Response to a Configuration Display 4-55
CONTROL Command 4-57
Scope in a Sysplex 4-58
Syntax 4-58
Changing Out Of Line Display Area Specifications 4-58

Deleting Retained Action Messages 4-59
Halting the Printing or the Display of a Status Display 4-61
Controlling Displays in Areas 4-61
Removing Information From the Screen 4-63
Activating, Deactivating, or Displaying the Status of the Action Message Retention Facility 4-64
Changing or Displaying the Number of Allowed WTL SYSLOG Buffers 4-65
Changing or Displaying the Number of Allowed WTO and WTOR Message Buffers 4-66
Changing the Time the System Waits for ROUTE Command Responses 4-66
Increasing the Maximum Number of Reply IDs 4-67
Changing or Displaying the Status of WTO Installation Exit IEAVMXIT 4-68
Displaying the SMCS APPLID of the current system and VTAM generic name for SMCS 4-69
Setting the APPLID of the System 4-69
Setting or Turning Off the VTAM Generic Name for SMCS 4-70
Changing a PFK Definition 4-71
Rerouting Message Queues 4-73
Changing or Displaying Message Deletion and Format Specifications 4-75
Changing or Displaying Time Intervals for Dynamic Displays 4-78
Changing the Operating Mode of a Console 4-79
Selecting the Message Levels for a Console 4-80
DEVSERV Command 4-83
Syntax 4-85
Parameters 4-85
DISPLAY Command 4-97
Scope in a Sysplex 4-98
Syntax 4-100
Displaying APPC/MVS Information 4-100
Displaying ASCH Configuration Information 4-106
Displaying Page Data Set Information 4-108
Displaying CONTROL Command Functions 4-109
Displaying Attached Coupling Facility Information 4-110
Displaying Console Group Definitions 4-110
Displaying Console Status Information 4-112
Displaying Data Lookaside Facility Information 4-116
Displaying the Domain Description Table 4-119
Displaying Dump Options or Dump Data Set Status 4-120
Displaying Extended MCS Information 4-124
Displaying the Timer Synchronization Mode and ETR Ports 4-128
Displaying Global Resource Serialization Information 4-130
Displaying TSO/E Parmlib Information 4-139
Displaying I/O Configuration Information 4-141
Displaying Dynamic Channel Path Management Information 4-141
Displaying IOS Group Information 4-141
Displaying MIH and I/O Timing Limits 4-142
Displaying the Devices Stopped by the IOACTION Command 4-145
Displaying IPL Information 4-146
Displaying System Activity 4-148
Displaying Started Task Status 4-156
Displaying Library Lookaside Information 4-158
Displaying the System Logger and its Log Streams 4-160
Displaying the Logrec Recording Medium 4-162
Displaying System Configuration Information 4-164
Displaying MVS Message Service Status and Languages 4-169

Displaying Message Suppression, Retention, Color, Intensity, and Highlighting Options 4-170
Displaying z/OS UNIX System Services Status 4-172
Displaying Sysplex-Wide Operator Information 4-181
Displaying PARMLIB Information 4-182
Displaying Commands Defined for PFKs 4-185
Displaying Registered Products 4-186
Displaying Entries in the List of APF-Authorized Libraries 4-187
Displaying Dynamic Exits 4-188
Displaying LNKLST Information 4-191
Displaying LPA Information 4-192
Displaying System Requests 4-194
Displaying RTLS Information 4-199
Displaying SLIP Trap Information 4-202
Displaying SMF Data 4-203
Displaying Storage Management Subsystem Information 4-205
Displaying Information about All Subsystems 4-211
Displaying Static System Symbols 4-213
Displaying the Local and Greenwich Mean Time and Date 4-214
Displaying Component or Transaction Trace Status 4-214
Displaying Device Status and Allocation 4-218
Displaying Workload Manager Information 4-221
Displaying Cross System Coupling Facility (XCF) Information 4-226
DUMP Command 4-235
Wildcards 4-235
Syntax 4-235
Parameters 4-236
Specifying Dump Options 4-246
DUMPDS Command 4-254
Syntax 4-254
Adding System Dump Resources 4-255
Enabling and Disabling Automatic Dump Data Set Allocation 4-258
Making Dump Data Sets Ready To Receive Dumps 4-259
Deleting System Dump Resources 4-260
Setting the Name-Pattern for Dump Data Sets 4-262
FORCE Command 4-264
Considerations 4-264
Syntax 4-265
Parameters 4-265
HALT Command 4-269
Syntax 4-269
IOACTION Command 4-270
Syntax 4-270
Parameters 4-270
LIBRARY Command 4-272
LOG Command 4-273
Syntax 4-273
Parameters 4-273
LOGOFF Command 4-274
Syntax 4-274
LOGON Command 4-275
Scope in a Sysplex 4-275
Syntax 4-275
Parameters 4-275
MODE Command 4-277
Syntax 4-277

Controlling the Recording of Hard Machine Check Interruptions 4-279
Controlling the Recording of System Recovery and Degradation Machine Check Interruptions 4-281
Displaying Recording and Monitoring Status 4-282
MODIFY Command 4-283
Summary of MODIFY 4-283
Using Asterisks in MODIFY Commands 4-284
MODIFY Command Syntax 4-285
Passing Information to a Job or Started Task 4-285
Syntax 4-286
Parameters 4-286
Passing Information To An z/OS UNIX System Services Application 4-287
Starting TSO/TCAM Time Sharing. 4-288
Stopping TSO/TCAM Time Sharing 4-288
Modifying TSO/VTAM Time Sharing 4-289
Controlling z/OS UNIX System Services (z/OS UNIX) 4-290
Communicating with the Catalog Address Space 4-295
Changing the DLF Processing Mode. 4-301
Changing the DLF parameters 4-302
Displaying DLF Status 4-302
Building and Replacing Library Lookaside Directories 4-303
Operating with the Network File System Server. 4-304
Collecting Problem Information for the Network File System Server 4-306
Managing the Object Access Method (OAM) 4-307
Recycling z/OS UNIX System Services (z/OS UNIX) 4-307
Switching Workload Management Modes and Changing Resource States 4-309
Specifying Data Set Selection Criteria for an External Writer 4-310
Causing an External Writer to Pause 4-312
MONITOR Command 4-315
Scope in a Sysplex 4-315
Syntax 4-315
Parameters 4-315
MOUNT Command 4-317
Scope in a Sysplex 4-317
Syntax 4-317
Parameters 4-317
Tape Library Dataserver Considerations 4-318
MSGRT Command 4-319
Syntax 4-319
Parameters 4-319
Stopping Message Routing 4-321
PAGEADD Command 4-322
Syntax 4-323
Parameters 4-323
PAGEDEL Command 4-324
Syntax 4-325
Parameters 4-325
QUIESCE Command 4-327
Syntax 4-327
REPLY Command 4-328
Using System Symbols in REPLY Commands 4-328
Scope in a Sysplex 4-329
Syntax 4-329
Replying to System Information Requests 4-329
Replying to System Requests During Recovery Processing 4-330
Replying to System Security WTORs 4-330

Setting the Time-of-Day Clock 4-331
Specifying Component Trace Options 4-332
Specifying Dump Options 4-332
Specifying SMF Options 4-333
Specifying System Parameters 4-334
RESET Command 4-335
Scope in a Sysplex 4-335
Syntax 4-335
Forcing a Hung MCS or SMCS Console Offline 4-335
Changing Service Classes or Quiescing Work 4-337
ROUTE Command 4-340
Restrictions 4-340
How MVS Displays Aggregated Response from ROUTE 4-341
Using System Symbols in ROUTE Commands 4-342
Syntax 4-343
Parameters 4-343
SEND Command 4-348
Scope in a Sysplex 4-348
Syntax 4-348
Communicating with Other Operators 4-348
Communicating with Specified Users 4-349
Communicating with All Logged-On Terminal Users 4-350
Saving Messages in the Broadcast Data Set 4-352
Listing the Notices Section of the Broadcast Data Set 4-353
Deleting a Message from the Broadcast Data Set (Notices Section) 4-354
SET Command 4-355
Scope in a Sysplex 4-355
Syntax 4-356
Parameters 4-356
SETDMN Command 4-366
Syntax 4-366
Restrictions 4-366
Parameters 4-367
SETETR Command 4-368
Syntax 4-368
Parameters 4-368
SETGRS Command 4-369
Syntax 4-369
Parameters 4-369
SETIOS Command 4-371
Syntax 4-371
Parameters 4-371
SETLOAD Command 4-376
Syntax 4-376
Parameters 4-376
SETLOGRC Command 4-378
Syntax 4-378
Parameters 4-378
SETOMVS Command 4-380
Syntax 4-380
Parameters 4-381
SETPROG Command 4-390
Syntax 4-390
Updating the APF List 4-391
Updating Dynamic Exits 4-393
Updating LNKLST Concatenations 4-396

Managing Dynamic LPA Content 4-400
SETRRS CANCEL Command 4-403
Syntax 4-403
Parameters 4-403
SETSMF Command 4-404
Syntax 4-404
Parameters 4-404
SETSMS Command 4-405
Scope in a Sysplex 4-406
Syntax 4-406
Parameters 4-406
SETSSI Command 4-412
Syntax 4-412
Parameters 4-412
SETXCF Command 4-415
Scope in a Sysplex 4-415
Syntax 4-415
SETXCF COUPLE Command 4-416
SETXCF FORCE Command 4-419
SETXCF MODIFY Command 4-421
SETXCF PRSMPOLICY Command 4-423
SETXCF START Command 4-424
SETXCF STOP Command 4-429
SLIP Command 4-433
Syntax 4-433
Using SLIP Commands 4-433
Processing of SLIP Commands 4-434
Coding SLIP Command Parameters 4-435
Setting a SLIP Trap 4-439
Modifying an Existing SLIP Trap 4-498
Deleting an Existing SLIP Trap 4-499
START Command 4-501
Syntax 4-501
Starting a System Task from a Console 4-502
Starting the APPC/MVS Address Space 4-505
Starting the APPC/MVS Transaction Scheduler Address Space 4-506
Starting the DFSMS Offering 4-507
Starting the Generalized Trace Facility 4-507
Starting the Library Lookaside (LLA) Address Space 4-509
Starting the Object Access Method (OAM) 4-510
Starting Resource Recovery Services (RRS) 4-510
Starting the System Object Model (SOM) Subsystem 4-511
Starting TSO/VTAM Time-Sharing 4-512
Starting the Virtual Lookaside Facility or Data Lookaside Facility 4-514
Starting an External Writer 4-514
STOP Command 4-516
Syntax 4-516
Stopping an Address Space 4-516
Parameters 4-516
Stopping an ASCH Initiator 4-519
Stopping the Data Lookaside Facility (DLF) 4-519
Stopping the Library Lookaside (LLA) Address Space 4-519
Stopping the Object Access Method (OAM) Address Space 4-520
Stopping a System Object Model (SOM) 4-520
Stopping the Virtual Lookaside Facility (VLF) 4-521
STOPMN Command 4-522

Scope in a Sysplex 4-522
Syntax 4-522
Parameters 4-522
STOPTR Command 4-523
Scope in a Sysplex 4-523
Syntax 4-523
Parameters 4-523
SWAP Command 4-524
Syntax 4-524
Operator-Requested DDR 4-524
System-Initiated DDR 4-525
SWITCH Command 4-526
Scope in a Sysplex 4-526
Syntax 4-526
Parameters 4-526
TRACE Command 4-529
Syntax 4-529
Parameters 4-530
Specifying TRACE CT Options 4-533
Specifying TRACE TT Options 4-535
TRACK Command 4-539
Scope in a Sysplex 4-539
Syntax 4-539
Parameters 4-539
UNLOAD Command 4-541
Scope in a Sysplex 4-541
Syntax 4-541
Parameters 4-541
VARY Command 4-542
Scope in a Sysplex 4-543
Syntax 4-544
Controlling Problem Determination Mode for the System Console 4-544
Controlling MCS and SMCS Consoles 4-545
Changing the Master Console 4-554
Controlling Hardcopy Processing 4-556
Placing a Secondary Console Online or Offline 4-559
Defining a Tape Device as Automatically Switchable 4-560
Placing an I/O Device or a Range of I/O Devices Online or Offline 4-561
Controlling a Global Resource Serialization Complex 4-566
Placing an I/O Path or Paths Online or Offline 4-567
Changing the State of Coupling Facility Cache Structures and Volumes 4-569
Placing an Optical Drive or Library Online or Offline 4-571
Placing a System-Managed Tape Library Online or Offline 4-571
Analyzing the State of the PDSE Subsystem 4-571
Releasing PDSE Latches 4-572
Changing the SMS Status of a Storage Group or Volume 4-573
Placing a Switch Port Online or Offline 4-574
Controlling an Application Environment 4-575
Activating a Service Policy 4-578
Removing a System from the XCF Sysplex 4-579
WRITELOG Command 4-580
Syntax 4-580
Parameters 4-580

Appendix. Accessibility A-1
Using assistive technologies A-1

Keyboard navigation of the user interface A-1

Notices B-1
Trademarks B-3

Index X-1

Figures

1-1. Format of the LOAD Parameter 1-3
1-2. Example of a Successful Response to a DISPLAY AUTOSWITCH Command 1-30
1-3. Example of an Unsuccessful Response to a DISPLAY AUTOSWITCH Command 1-30
2-1. Comparison of the Display Screens of Full-Capability and Output-Only Display Consoles 2-5
2-2. Example of DISPLAY CONSOLES,BACKLOG Command Output 2-21
3-1. Example of DISPLAY CONSOLES,A Command Output 3-5
3-2. Format of a Console Screen in Message Stream Mode 3-11
4-1. One System Command Format 4-13
4-2. A Second System Command Format 4-14
4-3. Display Output Illustration (Column Descriptions). 4-157
4-4. Display Output from D A,WTOR (Membername) 4-157
4-5. Display Output for D A,WTOR (Membername and Identifier) 4-157
4-6. Display Output for D A,WTOR (Membername and JOBNAME). 4-158
4-7. Display Output from D A,SYM1 4-158
4-8. Display Output from D A,SYMTEST 4-158
4-9. Display Output from D A,SYMBOLS 4-158

© Copyright IBM Corp. 1988, 2002

Tables

1-1. Possible Values IMSI Characters . . . 1-4
1-2. MVS System Commands With Sysplex Scope . . . 1-11
2-1. IBM Defaults for PFKs . . . 2-9
2-2. Checking the Commands Defined for Each PFK . . . 2-10
3-1. Comparison of System Commands and CONSOLE Parameters in CONSOLxx . . . 3-1
3-2. Comparison of System Commands and INIT Statements in CONSOLxx . . . 3-3
3-3. Comparison of VARY HARDCPY Commands and HARDCOPY Statements in CONSOLxx 3-4
3-4. Command Groups Used to Determine Command Authority . . . 3-6
3-5. Message Routing Codes . . . 3-14
4-1. System Command Summary . . . 4-2
4-2. System Command Syntax Notation . . . 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 . . . 4-32
4-7. Summary of the CONFIG Command. . . . 4-45
4-8. Summary of the CONTROL Command . . . 4-57
4-9. Sysplex Scope for CONTROL Command . . . 4-58
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 . . . 4-150
4-13. Displaying System Activity: Information for the ALL Operand . . . 4-150
4-14. Displaying System Activity: Information for a Specific Name. . . . 4-152
4-15. Examples of START Commands to Start Jobs. . . . 4-154
4-16. Examples of DISPLAY Commands . . . 4-154
4-17. Summary of the DUMPDS Command . . . 4-254
4-18. FORCE Command Tasks . . . 4-264
4-19. Summary of the MODE Command . . . 4-277
4-20. MODE Parameters Allowed for Machine Check Interruptions . . . 4-278
4-21. Summary of the MODIFY Command . . . 4-283
4-22. Examples of START Commands to Start Jobs. . . . 4-284
4-23. Examples of MODIFY Commands . . . 4-285
4-24. Possible Volume and Device Combinations on MOUNT Command . . . 4-318
4-25. Summary of the REPLY Command . . . 4-328
4-26. Summary of the RESET Command. . . . 4-335
4-27. Summary of the SEND Command . . . 4-348
4-28. Sysplex Scope for SET Command . . . 4-356
4-29. Acceptable Parameter Statements and Their Applicability . . . 4-386
4-30. Comparison of SET SMS with SETSMS . . . 4-406
4-31. Incorrect Combinations of SETSMS Parameters . . . 4-406
4-32. Summary of the SETXCF Command . . . 4-415
4-33. Sysplex Scope for the SETXCF Command . . . 4-415
4-34. Summary of the SLIP Command. . . . 4-433
4-35. Summary of the SLIP SET parameters . . . 4-441
4-36. Summary of the START Command . . . 4-501
4-37. Summary of the STOP Command . . . 4-516
4-38. Summary of the SWAP Command . . . 4-524
4-39. Summary of the VARY Command . . . 4-542
4-40. Sysplex Scope for VARY Command . . . 4-543
4-41. Goal Mode and Compatibility Mode Actions for VARY WLM,APPLENV. . . . 4-576

© Copyright IBM Corp. 1988, 2002

About This Book

This book describes how to use MVS system operator commands for the z/OS operating system. 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 Book

This book is intended for anyone using a console and system commands to control the operating system. This book 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 an z/OS operating system.

How To Use This Book

To describe the basic tasks within these general tasks and to provide a convenient system commands reference, this book 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, “Console Characteristics and Operations” on page 2-1, describes the consoles that MVS supports as operators’ consoles. It describes the operations and characteristics that you *cannot* define, including the operations that are common on all operator’s consoles.
- Chapter 3, “Defining and Changing Console Characteristics” on page 3-1, continues the console descriptions of Chapter 2 by describing the console characteristics that you *can* define. It describes the commands that operators and system programmers can use to tailor the consoles and console operations to the installation’s requirements. Chapter 3 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 book references information in other books, using shortened versions of the book title. For complete titles and order numbers of the books for all products that are part of z/OS, see *z/OS Information Roadmap*.

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for most of the z/OS, z/VM, and VSE messages you encounter, as well as system abends and some 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 access LookAt from the Internet at:

© Copyright IBM Corp. 1988, 2002

http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/

or from anywhere in z/OS where you can access a TSO command line (for example, TSO prompt, ISPF, z/OS UNIX System Services running OMVS). You can also download code from the *z/OS Collection* (SK3T-4269) and the LookAt Web site so you can access LookAt from a PalmPilot (Palm VIIx suggested).

To use LookAt on the Internet to find a message explanation, go to the LookAt Web site and simply enter the message identifier (for example, `$HASP701` or `$HASP*`). You can select a specific release to narrow your search.

To use LookAt as a TSO command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO from a disk on your *z/OS Collection* (SK3T-4269) or from the LookAt Web site. To obtain the code from the LookAt Web site, do the following:

1. Go to http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/.
2. Click **News**.
3. Scroll to **Download LookAt Code for TSO and z/VM.**
4. Click the ftp link, which will take you to a list of operating systems. Click the appropriate operating system. Then click the appropriate release.
5. Open the **lookat.me** file and follow its detailed instructions.

After you have LookAt installed, you can access a message explanation from a TSO command line by entering: **lookat** *message-id*. LookAt will display the message explanation for the message requested.

Note: Some messages have information in more than one book. For example, IEC192I can be found in *z/OS MVS System Messages, Vol 7 (IEB-IEE)* and also in *z/OS MVS Routing and Descriptor Codes*. For such messages, LookAt displays a list of books in which the message appears. You can then select one of the books to view the message explanation.

Accessing licensed books on the Web

z/OS licensed documentation in PDF format is available on the Internet at the IBM Resource Link Web site at:

http://www.ibm.com/servers/resourcelink

Licensed books are available only to customers with a z/OS license. Access to these books requires an IBM Resource Link Web userid and password, and a key code. With your z/OS order you received a memo that includes this key code.

To obtain your IBM Resource Link Web userid and password log on to:

http://www.ibm.com/servers/resourcelink

To register for access to the z/OS licensed books:

1. Log on to Resource Link using your Resource Link userid and password.
2. Click on **User Profiles** located on the left-hand navigation bar.
3. Click on **Access Profile.**
4. Click on **Request Access to Licensed books.**
5. Supply your key code where requested and click on the **Submit** button.

If you supplied the correct key code you will receive confirmation that your request is being processed. After your request is processed you will receive an e-mail confirmation.

Note: You cannot access the z/OS licensed books unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

To access the licensed books:

1. Log on to Resource Link using your Resource Link userid and password.
2. Click on **Library**.
3. Click on **zSeries**.
4. Click on **Software**.
5. Click on **z/OS**.
6. Access the licensed book by selecting the appropriate element.

Summary of changes

Summary of changes
for SA22-7627-03
z/OS Version 1 Release 3

This publication contains information previously presented in *z/OS MVS System Commands*, SA22-7627-02, which supports z/OS Version 1 Release 3.

Changed information

Formatting changes are made to some commands to improve accuracy and usability.

You may notice changes in the style and structure of some content in this book—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 books.

Summary of changes
for SA22-7627-02
z/OS Version 1 Release 3

This publication contains information previously presented in *z/OS MVS System Commands*, SA22-7627-01, which supports z/OS Version 1 Release 2.

New information

- Additional information on the use of the SET GRSRNL command has been provided. See the description of the GRSRNL operand on page 4-359.
- A note has been added with additional clarification of how to use command grouping. See “System Commands Grouped According to System Command Authority” on page 3-5.
- A note has been added with additional information on storage usage by the PAGEDEL DELETE command.
- A note has been added with additional information on storage usage by the ADD parameter of the SETPROG LPA command.
- A note has been added about the use of the CSVDYLPA exit routine with the SETPROG LPA command.
- An appendix with z/OS product accessibility information has been added.
- Beginning with z/OS V1R3, WLM compatibility mode is no longer available. This affects the following system commands: DISPLAY DMN, MODIFY WLM, and RESET.
- JESLOG has been added to the list of supported optional keywords for the START command

Changed information

- The description of the E=id parameter of the CONFIG ESTOR command has been corrected.
- The description of the (dddddX-dddddX) parameter of the DISPLAY M STORAGE command has been corrected.

© Copyright IBM Corp. 1988, 2002

Vertical lines in the left margin indicate technical changes or additions to the text and illustrations. This book also includes terminology, maintenance, and editorial changes.

You may notice changes in the style and structure of some content in this book—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 books.

Summary of changes
for SA22-7627-01
z/OS Version 1 Release 2

This publication contains information previously presented in SA22-7627-00, which supports z/OS Version 1 Release 1.

New information

- Documentation of the new CMDS command. See “Command Flooding” on page 1-34 and “CMDS Command” on page 4-43.
- Descriptions of new parameters added to the SLIP command: ACTION=STOPGTF and MSGID=. See “SLIP Command” on page 4-433.
- Additional information about using IBM-supplied sample IEADMCxx parmlib members. See the PARMLIB= parameter in “DUMP Command” on page 4-235.

Changed information

- Clarification of how to use command grouping. See “System Commands Grouped According to System Command Authority” on page 3-5.
- Notification that symbol substitution is now permitted on all sub-operands of the DUMPDS command.

Deleted information

- Description of the SLIP SET,SB1 parameter. That parameter is supported only on processors that z/OS no longer supports.

Summary of changes
for SA22-7627-00
z/OS Version 1 Release 1

This book contains information also presented in *OS/390 MVS System Commands*.

Technical changes include updates to these commands to support SMCS consoles:

- “CONFIG Command” on page 4-45
- “CONTROL Command” on page 4-57
- “DISPLAY Command” on page 4-97
- “LOGOFF Command” on page 4-274
- “LOGON Command” on page 4-275
- “RESET Command” on page 4-335
- “SWITCH Command” on page 4-526
- “VARY Command” on page 4-542

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 1-21
- "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 1-30
- "Responding to Failing Devices" on page 1-37
- "Quiescing the System" on page 1-38
- "Stopping the System" on page 1-38

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.
- *z/OS MVS Recovery and Reconfiguration Guide*, which contains information about handling recovery and reconfiguration in a system or sysplex.

© Copyright IBM Corp. 1988, 2002

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 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".
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 1-5.

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

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

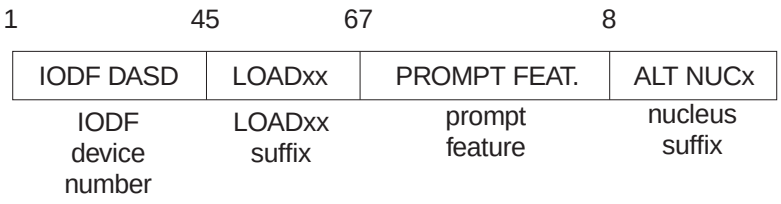


Figure 1-1. Format of the LOAD Parameter

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 zeroes 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 zeroes. 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 no volume is 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 still 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, however, 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

Table 1-1. Possible Values IMSI Characters (continued)

IMSI Character	Display Informational Messages	Prompt for Master Catalog Response	Prompt for System Parameters Response
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*.

Notes:

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 consoles on the system) or on the CONSOLE statement (for a single console). When LOGON is a system requirement, you can issue commands only through the master console until RACF is fully initialized and able to process logon requests. Until RACF is initialized, you cannot issue any commands from any secondary console, with one exception. You can issue the VARY MSTCONS command from a secondary console to establish an alternate master console if you need operator intervention to complete RACF initialization.

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 4-275 for more information.

IBM recommends 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 consoles on the system) 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 4-333 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 GMT   DATE=1991.301,CLOCK=22.31.53
*00 IEA888A LOCAL DATE=1991.301,CLOCK=17.31.53  REPLY U, OR GMT/LOCAL TIME
```

The values in this message indicate that the local time is 5:31:53 P.M. on October 28, 1991 and that Greenwich mean time (GMT) 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 GMT   DATE=1991.302,CLOCK=13.00.00
*00 IEA888A LOCAL DATE=1991.301,CLOCK=08.00.00  REPLY U, OR GMT/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 GMT and local time values are still not accurate enough, you can reply with new GMT 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 4-328.

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 4-109 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 4-501.

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.
- Switch the workload management mode in effect on a system.

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 4-193 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 4-315.

Using the TRACK command, you can get a periodic display of active batch jobs, started tasks, active time-sharing users, and active initiators. The information is

displayed at 30-second intervals unless you specify a different time interval with the CONTROL command. You can stop all or part of the TRACK display with the STOPTR command. See “TRACK Command” on page 4-539 for information about the TRACK command.

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 4-218 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 4-163 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 4-83 for information about the DEVSERV command.

Communicating Among Several Consoles

The MSGRT command directs the output of displays to one or more consoles in the system. You can route any display or message produced by the CONFIG, DISPLAY, MONITOR, or TRACK commands. You can also route the action of many CONTROL commands. See “MSGRT Command” on page 4-319.

Using the SEND command, you can communicate with operators of other consoles in your system, and you can send messages to remote terminals. See “SEND Command” on page 4-348.

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 4-57.

You can use the ROUTE command to send commands to be processed on other systems in the sysplex. See “ROUTE Command” on page 4-340.

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 4-545.

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=.

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

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 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	All
DISPLAY PFK	Has sysplex scope only when you specify CN=.
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=.

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

Command	Conditions
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.
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.
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 START|STOP	Have sysplex scope only when you specify POLICY or REBUILD.
STOPMN	Has sysplex scope only when you specify L=.
STOPTR	Has sysplex scope only when you specify L=.
SWITCH CN	All
TRACK	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: • You issue VARY CN(*conspec*),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 ...,MSTCONS	Has sysplex scope when you issue VARY *conname*,MSTCONS. Also has sysplex scope when you issue VARY *devnum*,MSTCONS, but only if you use a common IODF for the specified device across the sysplex.

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

Command	Conditions
VARY SMS, STORGRP|VOLUME	Has sysplex scope under these conditions only: • You specify (*storgrp|volume*,ALL) and all systems in the sysplex are in the same SMS complex. • You specify (*storgrp|volume system*) where *system* 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 4-213 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 WTORs 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=cca" 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 will occur on the systems where the commands are issued, not where the "L=" console is attached.
   - IBM recommends not using symbolic variables in the "L=" parameter on the ROUTE command. See the ROUTE command description in this manual.
   - You should 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 *OS/390 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 the description of the START command in *z/OS MVS System Commands*.

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 AZZZZC1 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, such as the installation performance specification (IPS). See "SET Command" on page 4-355.

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 1-38.)
- 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 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 the master console or its first designated alternate
- When the master console and its first designated alternate are not display consoles

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 the master console or its first designated alternate.

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 4-578.

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 MODIFY command to switch the workload management mode in effect on a system, see “Switching Workload Management Modes and Changing Resource States” on page 4-308.

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*.

Modifying a Job's Performance Group

Important

Beginning with z/OS V1R3, WLM compatibility mode is no longer available. Accordingly, the RESET PERFORM= command is no longer valid. The information has been left here for reference purposes, and for use on backlevel systems.

Using the PERFORM= keyword with the RESET command, you can change the performance group of a job that is executing. This command/keyword combination affects the current job step and all subsequent job steps.

Note: The PERFORM= keyword with the RESET command is valid only on systems operating in workload management compatibility mode; it is not valid in goal mode.

Modifying a Job's Service Class

Using the SRVCLASS= keyword with the RESET command, you can change the service class of a job that is executing. This command/keyword combination affects the current job step and all subsequent job steps.

Note: This command/keyword combination is not valid on systems operating in workload management compatibility mode (but is valid in goal mode).

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 in the installation performance specifications (IPS) 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 DFSMS 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
Force the started task	"FORCE Command" on page 4-264
Modify the started task	"MODIFY Command" on page 4-283
Name the started task	″START Command″ "Starting a System Task from a Console" on page 4-501 (JOBNAME= parameter)
Start the started task	″START Command″ "Starting a System Task from a Console" on page 4-501
Stop the started task	"STOP Command" on page 4-516

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 4-529.

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 4-501 and “STOP Command” on page 4-516. 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. The table contains the hardcopy message set. 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 1-29, turns the AUTOSWITCH attribute on and off.
2. The device is varied online through the VARY ONLINE command.
3. Prior to 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.

   Automatically switchable devices are managed using information stored in a list structure named IEFAUTOS, which resides on the coupling facility. Through this structure, the systems track the availability of automatically switchable tape devices, select appropriate tape devices to satisfy requests, and control their use. The structure is defined in the CFRM policy and the SETXCF START,POLICY command makes the policy active.

   For systems at z/OS Release 2, with APARs OW51103 and OW50900 installed, and higher, the ATS STAR function uses global resource serialization and XCF services to maintain serialization when allocating shared tape devices instead of the IEFAUTOS structure. The system maintains information about the use of an autoswitchable device in the Allocation address space (ALLOCAS).

The following section describes the coexistence behavior when some systems are at z/OS R2 (with the appropriate PTFs) or higher and are using ATS STAR to manage autoswitchable devices and other systems are at a level lower than z/OS R2 and are using the IEFAUTOS coupling facility structure for autoswitchable device management. Throughout this section, the term “participating systems” refers to systems connected to the same IEFAUTOS structure. The term “nonparticipating system” refers to a system that is not connected to (perhaps has lost its connection to) that same IEFAUTOS structure or which is using ATS STAR.

Participating and nonparticipating systems take turns using the automatically switchable devices:

- If a participating system requests an automatically switchable device that is allocated on another participating system, the system allows the job to wait and issues message IEF292I. When the device becomes available, the system issues message IEF294I and makes the device available.
- If a nonparticipating system varies online an automatically switchable device, once the device is allocated to the nonparticipating system, the device becomes “assigned to a foreign host” to the other system. until the device is varied offline. If participating systems try to allocate the device during that time, the system issues message IEF284I. When the device is varied offline by the nonparticipating system, the device becomes available to the participating systems and message IEF294I appears.

  Note that some automatic tape switching messages use the term **foreign host** to describe a nonparticipating system. The message appears when a system using the device is not connected to the same IEFAUTOS structure or is connected to the structure but is using the device as a dedicated device.

  Information in IEFAUTOS is not updated when a nonparticipating system allocates a device that is defined as automatically switchable. You cannot obtain this information through the DISPLAY AUTOSWITCH command. Participating systems are unaware that the device is not available to them. However, once an

attempt is made to allocate and assign the device within the IEFAUTOS configuration, the IEFAUTOS structure is updated to reflect the fact that the device is assigned to a foreign host. This information can be obtained through the DISPLAY U,,AS command.

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 4-560.

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 reIPL.

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 about which the coupling facility (that is, the IEFAUTOS structure) has information. 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
IEE343I 12.24.59 UNIT STATUS         FRAME LAST   F      *E   SYS=ALLOC5
UNIT  TYPE  STATUS     SYSTEM    JOBNAME   ASID  VOLSER   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 AUTOSWITCH command is in “Displaying Device Status and Allocation” on page 4-218.

If you issue DISPLAY U,,AUTOSWITCH and the system-to-IEFAUTOS connection is broken, you receive the following message:

```
 - d u,,as
IEE343I 15.09.07 UNIT STATUS 990
UNIT  TYPE  STATUS     SYSTEM    JOBNAME   ASID  VOLSER   VOLSTATE
AUTOSWITCH STATUS UNAVAILABLE - ALLOCATION COUPLING
FACILITY STRUCTURE NOT AVAILABLE
```

Figure 1-3. Example of an Unsuccessful Response to a DISPLAY AUTOSWITCH Command

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 *z/OS MVS Initialization and Tuning Guide* and *z/OS MVS Recovery and Reconfiguration Guide* for information on setting up hot I/O recovery defaults.) 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 xvii) or use the *MVS System Messages* books 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.

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.

Command Flooding

Starting with APAR OW45398, which is incorporated into z/OS V1R2, attached commands that run in the *MASTER* or CONSOLE address space are divided into four 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 cancelled, 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 CF
- DISPLAY CNGRP
- DISPLAY DLF
- DISPLAY DMN
- 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
- SEND
- SET UNI
- SETETR
- SETGRS
- SETIOS
- 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 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
- ROUTE
- 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
- SWITCH CN
- TRACK

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
- MSGRT
- RELEASE TP
- SETDMN
- STOP
- STOPMN
- STOPTR
- 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.

Chapter 2. Console Characteristics and 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

Chapter 2 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.

Chapter 3 continues the console descriptions of Chapter 2 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.

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 (a card reader, for example, cannot be used as an output device).
   - The device cannot be assigned as a console.

   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 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)
```

© Copyright IBM Corp. 1988, 2002

must appear in the CONSOLxx 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; one console is the **master console**. The other consoles can be active (being used) or inactive. They can also have different levels of authority, for more information see *z/OS MVS Planning: Operations*.

The master console is your principal means of communicating with the system. The master console, along with 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
- Switch to a new master console

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 3-5), 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 CONSOLxx member of parmlib to find which console is the master console, and what the operating modes of the consoles are. It also looks for other attributes, such as:

- 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
- Alternate console groups — the backup consoles assigned in case of a malfunction or a console switch

- Hardcopy medium — the system log or the console 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, canceling 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 2-1 on page 2-5 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 2-1 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.

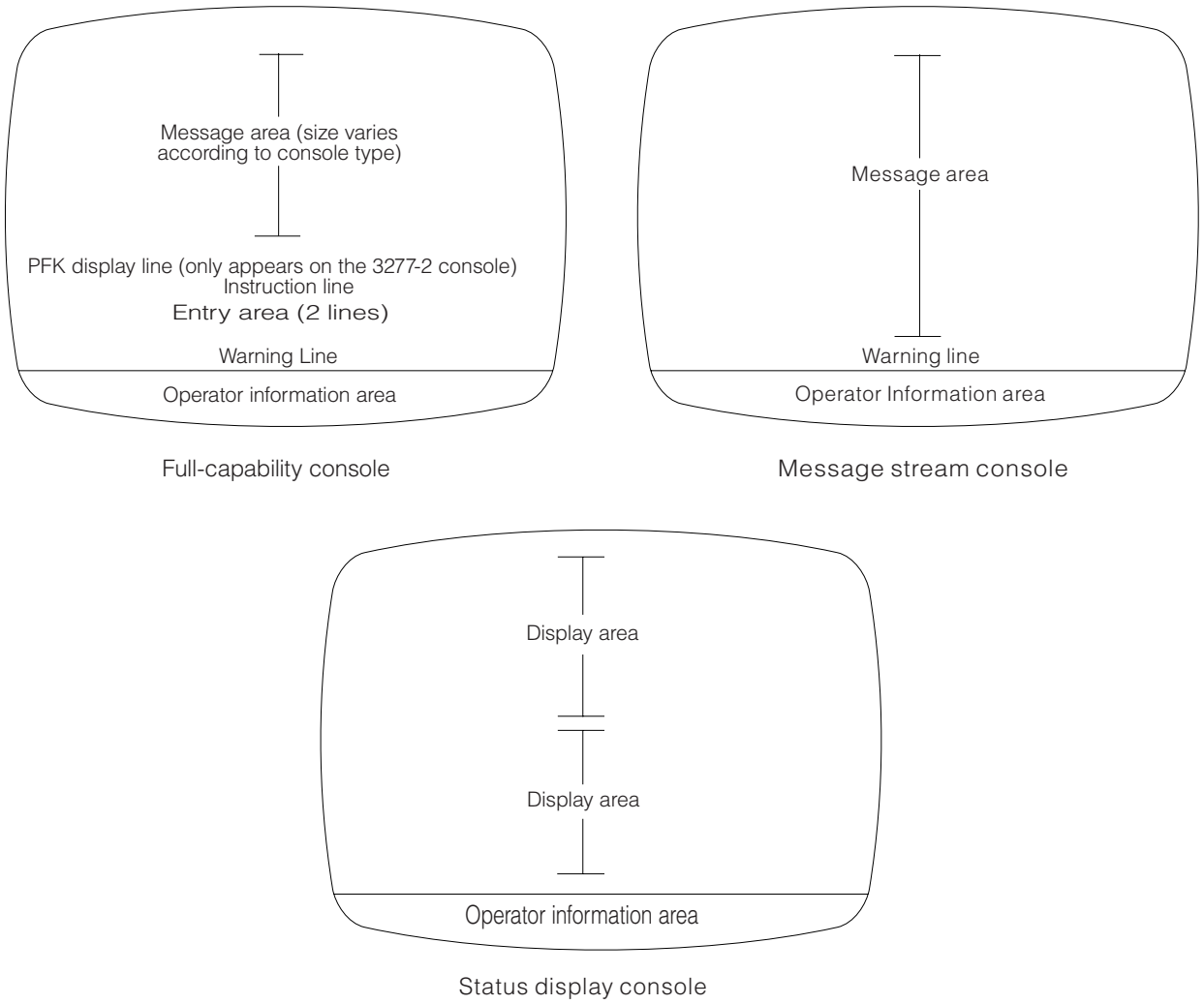


Figure 2-1. Comparison of the Display Screens of Full-Capability and Output-Only Display Consoles

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 or that you can delete 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 2-6.

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 3-16. For more information about console (IEE) messages, use LookAt (see “Using LookAt to look up message explanations” on page xvii) 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 or MSGRT 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 2-13.

If the system detects an error in a command other than a CONTROL or MSGRT 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 3-20 in Chapter 3.

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 the following definitions (in nonconversational mode):

Table 2-1. IBM Defaults for PFKs

PFK	Command Comment	Definition
1	CONTROL E,1	Erase one line from screen
2	CONTROL E	Erase one segment from screen
3	CONTROL E,D	Erase status display from screen
4	CONTROL D,F	Frame display forward in area
5	CONTROL S,DEL=N	Hold in-line output
6	CONTROL S,DEL=RD	Resume in-line output
7	DISPLAY A,L	List active jobs and TSO users
8	DISPLAY R,L	List all outstanding operator action requests
9	CONTROL D,U	Update dynamic display
10 and up		No definition provided

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 2-13.

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 2-2. 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
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=cc, where cc is the console’s id

“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 4 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 04
```

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 2-13.)
   - 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 cancelled, 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 Chapter 3 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 2-11.
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:

```
IEE722I PFK nn NOT DEFINED
```

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 2-13.
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,'DISLAY 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:

```
TRACK A,L=C_
```

Move the cursor back to the C, press the INS MODE key, type in 10, and press the RESET key. The command then reads:

```
TRACK A,L=10C
```

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 0 from the following command:

```
TRACK A,L=100A_
```

Position the cursor at either 0 and press the DEL key. The command then reads:

```
TRACK A,L=10A
```

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. See “Recovering Lost Messages” on page 2-19 later in this chapter.

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. In particular, the inactivity might be the

result of processing a multiple-line message. (For a more detailed discussion of unended multiple-line message processing, see ″Truncated Data in Multi-line Messages″ in the *MVS System Messages* books.

One function of the message handling portion of the control program is to check for the end line of a status display or other multiple-line messages. If, for some reason, the system fails to detect the end line, your console might be put into an inactive condition awaiting completion of the display or message. This situation occurs because the system presents all lines of a status display or multiple-line message, once it is begun, before presenting any other message on the console.

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.
- Issue the CANCEL command to cancel any jobs that have written multiple-line messages to 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. To avoid losing the console's messages, issue the CONTROL Q command to reroute the messages to another console or to the system log. Check for a console hardware error. Also, if possible, switch control to another console. 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 OR SWITCH CONSOLES
```

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 and automatically switches consoles. All messages (except status displays) are moved to the new console. (See “Console Groups” on page 3-8 for a description of console switching.)

Responding to a Blank Screen

If the console screen goes blank, the system is probably switching to the console's alternate. When the switch is done, the following message should appear on the alternate console:

```
IEE870I CONSOLE SWITCH. OLD=(dev,system) NEW=(dev,system) REASON=reason
```

Appropriate values and system symbols appear in the actual message. Use the alternate console to continue operating the system, and have the old console checked for the source of the error.

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.

If the system has switched to the console's alternate, operate the system from the alternate console, and have the old console examined for the source of the error.

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 2-15.

Responding to a Master Console Failure

Normally, MVS detects a failing master console and automatically switches to the alternate of the master console. A message is sent to the new master console to inform the operator that his console is now the master console. The message also informs the operator of the device number of the old master console so that it can be repaired.

Occasionally, the system does not detect a failure. In this case, you should press the external interrupt key to switch to the alternate.

When no alternate consoles are active, console switching cannot take place. If the master console fails, and the master console function cannot be switched to another console, all active MCS and SMCS consoles receive message IEE141A, which indicates that there is no master console. There are several ways to resolve a no-master-console condition:

- Issue VARY CN(),ONLINE to activate an MCS console and make it the master console.
- Issue CONTROL V,USE=FC to change an active MCS output-only console to a full-capability console, and make that console the master console.
- Issue the VARY MSTCONS command to make an active MCS or SMCS console the master console.
- Activate a master-authority SMCS console to make that console the master console.

Note: In a no-master-consoles condition, any MCS or SMCS console will accept the VARY MSTCONS command to make that console the master console, unless the installation has configured the security product to only allow certain operators to issue the command. See *z/OS MVS Planning: Operations* for more information on a no-master-consoles condition and protecting the VARY MSTCONS command.

If no secondary consoles are active when a master console failure occurs, a no-consoles condition occurs. See the next topic, Responding to a No-Consoles Condition.

Responding to a No-Consoles Condition

A no-consoles condition occurs when no full-capability consoles (consoles capable of input and output functions) are available to the system. When the system enters a no-consoles condition, WTO message buffers are not freed and, when the IPL-specified limit of buffers is reached, non-privileged tasks wait until WTO message buffers become available. See “Responding to Console Message Backups” on page 2-19.

Note: Output-only consoles can continue to function during a no-consoles condition.

The system tries to notify you of a no-consoles condition in one or more of the following ways:

- Issuing message IEA546E to any active output-only console.
- Sounding an alarm on a device. The system can sound a device alarm only if:
  - The device was specified as a console at system installation
  - The device has an alarm feature
  - The device is online or in console status
  - The device is unallocated
- Sounding the processor controller alarm.
- Stopping message traffic on full-capability consoles. A lack of message traffic could be the only indication of a no-consoles condition because the system might not find a console on which message IEA546E can appear and might not find any devices on which to sound an alarm.
- Rejecting any VARY CONSOLE commands

Before trying to recover from a no-consoles condition, understand that:

- You cannot attempt recovery until all alarms have sounded and message IEA546E has been issued.
- Recovering the master console should be your first priority.

To recover the master console, the operator can perform one of the following actions:

- Issue VARY CN(),ONLINE (from an extended MCS or subsystem console) to activate a full-capability console that is offline and make it the master.
- Issue CONTROL V,USE=FC (from an extended MCS or subystem console) to change an active message stream or status display MCS console to a full-capability MCS console and make it the master.
- Press the attention interrupt key on an MCS console device that is to become the master console, then press the external interrupt key on the system console to activate the device as the master console.
- In a sysplex, an operator can use the system console to IPL a system with a full-capability console (defined with AUTH=MASTER) into the sysplex.

- Activate an SMCS console to relieve the no-consoles condition. If the console was defined AUTH=MASTER, it will become the master console. Otherwise, it will not become the master console, and the system or sysplex will enter a no-master-console condition as described in ″No-Master-Console Condition″ in *z/OS MVS Planning: Operations*. In a no-master-console condition, the operator using the console can make the console the master console using the VARY MSTCONS command, unless the installation takes steps to prevent it as described in ″No-Master-Console Condition″ in *z/OS MVS Planning: Operations*.

If the recovery attempt is successful, the device that generated the attention interrupt becomes the master console and message IEE870I appears on it.

This message indicates that the failing master console's functions have been switched to the new master console, messages queued to appear on the failing master console now appear on the new master console, and the no-consoles condition is resolved.

Recovering Lost Messages

If a console fails, you might lose some messages. To recover lost messages, use the CONTROL Q command to reroute the failing console's message queue to another console or to the system log. If the failing console is a full-capability console, you must issue this command from the master console. Two types of messages are not requeued:

- Inline status displays, which are deleted
- Out-of-line status displays, which remain on the failing console

CONTROL Q requeues action messages. However, if they are being queued for display on only one console, the system log could be the only place they will subsequently appear.

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.

When messages fill up the buffers, most jobs that issue a WTO or WTOR go into a wait until buffers are available. 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
```

Most tasks issuing a WTO go into a wait until the buffer shortage is relieved. Some tasks might cause a system outage if they were prevented from processing by having to wait for WTO buffers. Examples are JES3, the communications task, authorized programs that issue a WTO as a command response, and authorized programs that issue additional lines of a multiple line WTO when the first line was issued before the buffer shortage. For these tasks, the system uses special WTO buffers in storage above 16 megabytes to hold the WTO requests and allow the task to continue processing. If the system exhausts these special WTO buffers, then WTOs may be lost and the system issues the following message:

```
IEA652A WTO STORAGE EXHAUSTED - WTOS WILL BE DISCARDED
```

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, perhaps because it is in a loop. The system issues message IEA653I when a job uses a high percentage of the WTO buffers. The message identifies the jobname and the address space.
- Another system in the sysplex is sending an excessive number of messages.
- Another system in the sysplex is experiencing a buffer shortage. Systems trying to send messages to the constrained system might experience a buffer shortage as result of holding messages for the constrained system. Message IEA654A identifies the constrained system.

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(s) specified in message IEA653I. If cancelling a job would cause a serious impact, look at the message(s) 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 2-2 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=1 LIM=1500` tells you the current use of WTO buffers and the specified limit. `RPLY: CURR=1356 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 2-2 on page 2-21 indicates, through NBUF, the number of buffers queued to each console. In this example, console 12, with 1217 message buffers filled, is the source of the problem. The buffer limit of 1500 seems adequate, so console 12 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/ALT        ID  --------------- SPECIFICATIONS ---------------
 12/CON2            12  COND=M      AUTH=MASTER       NBUF=1217 UD=Y
  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 BEING HELD FOR OTHER SYSTEMS - WTO BUFFER USAGE
 SYSTEM = 2     NBUF=   212
 MESSAGES COMING FROM OTHER SYSTEMS - WTO BUFFER USAGE
 SYSTEM = 2     NBUF=     4
```

Figure 2-2. 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.

However, the system puts all non-privileged tasks requesting a buffer into a wait until the buffer count goes below the limit.

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 a console) becomes active or NIP processing is complete. After NIP processing, consoles other than the master console 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 may have defined the master console, the system console, or any other MCS console as members of an alternate 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 an alternate console group or a console is not active, the system that issues the message tries to select a console in the following order:

- The master console, if it is active and physically attached to the system that issues the message.
- 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. However, any messages that would be routed exclusively to the bypassed console will be routed to the consoles that receive undelivered messages.

Before using the VARY command to bypass the master console, you must assign the master console's functions to another console.

Before you use the VARY command to bypass the hardcopy medium, you must change the hardcopy medium to another console device or to the system log.

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*.

Chapter 3. 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, MSGRT, 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, UD loss, or SYNCHDEST loss. If a loss occurs, MVS issues a DISPLAY CONSOLE,HCONLY 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
SWITCH CN

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 3-1. 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 (See Table 2-1 on page 2-9)	PFK table
CONTROL S,CON	CON(N)	Conversational or nonconversational mode
CONTROL S,DEL	DEL(RD)	Message deletion mode

© Copyright IBM Corp. 1988, 2002

Table 3-1. Comparison of System Commands and CONSOLE Parameters in CONSOLxx (continued)

MVS Commands	CONSOLE Parameters with DEFAULT	Characteristic that the Parameter Affects
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
CONTROL S,SEG	SEG	Number of lines in the message area that can be deleted by a CONTROL E,SEG command
CONTROL T,UTME	UTME(30)	Time interval for updating dynamic displays
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
MSGRT	MSGRT	Routing of system commands to a specified console and message area
VARY CN,ALTCONS	ALTERNATE	Alternate console
VARY CN,ALTGRP	ALTGRP	Alternate group for the console
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
VARY CN,UD	UD	Console is to receive undelivered messages

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 3-2 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 3-2. 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
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 3-3 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 3-3. Comparison of VARY HARDCPY Commands and HARDCOPY Statements in CONSOLxx

VARY HARDCPY Command Parameters	Parameters on HARDCOPY Statement	Description
devnum, SYSLOG, or OPERLOG	DEVNUM	Establishes whether the hardcopy medium is a device, 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
UD=N	UD(N)	Establishes whether the system console receives undelivered messages

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.
- Undelivered messages will be sent to the system console when they are not sent to any other console.

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 3-1 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 xvii) 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/ALT        ID  --------------- SPECIFICATIONS ---------------
 SYSLOG                 COND=H      AUTH=CMDS          NBUF=0    UD=N
                        ROUTCDE=ALL
 01/02              01  COND=M      AUTH=MASTER        NBUF=N/A  UD=Y
  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
 21/22              21  COND=A      AUTH=MASTER        NBUF=1    UD=N
  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 3-1. 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)

Note: Iif RACF is used to control who can issue commands, then the RACF OPERCMDS settings will 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) could issue a MODIFY command. Similarly, a job submitted in an AUTH(ALL) jobclass will be unable to issue a MODIFY command, it the user does not have access to the proper OPERCMDS profile.

The commands in each group are shown in Table 3-4. 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 the master console, you can enter all operator commands. Any console with AUTH(MASTER) in the CONSOLxx parmlib member has master console authority. In a sysplex, the first system that IPLs and has a MCS console statement with AUTH(MASTER) in the CONSOLxx parmlib member defines the master console for the sysplex.

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 3-4. Command Groups Used to Determine Command Authority

Command Group	Commands	
INFO	CMDS DISPLAY CMDS SHOW CONTROL (See Note 4 on page 3-7) DEVSERV DISPLAY (See Note 1 on page 3-7) LOG LOGOFF LOGON MONITOR MSGRT (See Note 1 on page 3-7)	REPLY (See Note 5 on page 3-7) ROUTE SEND STOPMN STOPTR (See Note 1 on page 3-7) TRACK (See Note 1 on page 3-7)
SYS (system control)	ACTIVATE CANCEL CHNGDUMP DUMPDS HALT (See Note 3 on page 3-7) HOLD (See Note 2 on page 3-7) LIBRARY MODE MODIFY PAGEADD PAGEDEL RELEASE (See Note 2 on page 3-7) RESET SET	SET GRSRNL SETDMN SETETR SETIOS SETLOAD SETOMVS SETPROG 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 3 on page 3-7) {OFFLINE} (See Note 6 on page 3-7) {ONLINE } (See Note 6 on page 3-7) {PATH } {name or [/]devnum}
CONS (console control)	CONTROL (See Note 4 on page 3-7)	VARY {CN{...}[,ALTGRP=...] } {CONSOLE[,ALTCONS=...]} {OFFLINE} (See Note 6 on page 3-7) {ONLINE } (See Note 6 on page 3-7) {PATH } {name or [/]devnum }

Table 3-4. Command Groups Used to Determine Command Authority (continued)

Command Group	Commands	
MASTER (master console control)	CMDS ABEND CMDS REMOVE CONFIG CONTROL (See Note 4) DUMP FORCE IOACTION QUIESCE RESET CN SETGRS SETLOGRC SETSSI SETXCF	SWITCH CN TRACE (with MT) VARY {CN(...)[,AUTH=...]} {CN(...)[,LOGON=...]} {CN(...)[,LU=...]} {CONSOLE[,AUTH=...]} {GRS } {HARDCPY } {MSTCONS } {OFFLINE,FORCE } {XCF }

Notes:

1. CONS command group when message routing is specified.
2. HOLD and RELEASE are related to the Telecommunications Access Method (TCAM).
3. HALT NET and VARY NET are related to the Virtual Telecommunications Access Method (VTAM)
4. CONTROL is in the INFO command group except when
   - Rerouting 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.
5. An operator can reply to any message that the console is eligible to receive. Any console with master authority can reply to any message.
6. 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 console authority. If you try to enter this command from a console without master console 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.

Changing the Master Console in a System or Sysplex

By assigning the system command groups for a console, you establish the master console and the command level for other consoles. In a sysplex, the first system

that IPLs and has a MCS console statement with AUTH(MASTER) in the CONSOLxx parmlib member defines the master console for the sysplex.

You assign the master console by:

- **Using the VARY Command**:

  The VARY device,MSTCONS command defines the master console.

  Example

  To reassign the console named CON31E as a console with master console authority, enter:

  ```
  VARY CON31E,MSTCONS
  ```

  Enter this command through any console that has master console authority. If you try to enter this command from a console without master console 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.

See “Selecting a Master Console Using Alternate Console Groups” on page 3-9 to define a master console when no full-capability consoles are available.

Console Groups

Each MCS, SMCS or extended MCS console can be assigned a console group to be used as a backup if a console fails. An extended MCS console (1) increases the number of consoles by using TSO/E terminals on MVS systems or (2) allows applications and programs to access MVS messages and send commands. See *z/OS MVS Planning: Operations* for information about extended MCS consoles.

The console group is a list of other functioning consoles in the system configuration, with the same or greater capability (that is, a full-capability console can be the alternate for a status display console, but an output-only console cannot back up a full-capability console). When the system detects that the console is failing, it automatically switches to the first available console in the failing console’s alternate console group. If the system cannot find an active alternate, the system does not switch the failing console. See *z/OS MVS Planning: Operations* for more information about the arrangement of console groups.

When a console fails, and no console groups have been established, the system tries to switch to the failing console’s alternate if one has been specified. The alternate console is assigned the functions of the failing console. MVS transfers to the alternate console all unanswered and unissued messages from the failing console. If there is no active alternate console, the console function provided by the failed console is no longer available.

Notes:

1. Multiple-line messages being written when the failure occurs are not transferred. The alternate console operator receives a message stating the functions the alternate console has assumed from the failing console.
2. MONITOR displays **are not** transferred when an MCS or SMCS console is varied offline or switched. The displays are ended. The operator must re-issue the MONITOR command when the MCS or SMCS console is restarted or on the new MCS or SMCS console.

To determine the alternate console group or alternate console for your consoles, use the DISPLAY CONSOLES command. Figure 3-1 on page 3-5 shows the display that appears in response to this command.

You assign an alternate console by:

- **Issuing the VARY Command**:

  Use the ALTCONS operand on the VARY command to assign an alternate console to your console. You can only specify one alternate console on the ALTCONS keyword.

  Example

  To assign the device with device number 2E1 as an alternate console for the console with device number 1E1, enter:

  ```
  VARY (1E1),CONSOLE,ALTCONS=2E1
  ```

  Use the ALTGRP operand on the VARY command to assign an alternate console group. Console groups and their members are defined in CNGRPxx.

  Example

  To assign the console group 'group9' as the alternate console group for the console 'TAPECN', enter:

  ```
  VARY CN(TAPECN),ALTGRP=GROUP9
  ```

  The effect of this command lasts only for the duration of the IPL. The system programmer can specify ALTGRP or ALTCONS on the CONSOLE statement in CONSOLxx to make a permanent change.

Selecting a Master Console Using Alternate Console Groups

When no full-capability consoles are available in a system or a sysplex, you can select a console from an alternate console group specified on the INIT NOCCGRP statement and activate it as the master console. When the operator presses the attention interrupt key on any console device that is a member of the alternate console group and then presses the external interrupt key of the system console, MVS can activate the console as the master console.

In a system or a sysplex, the master console becomes the alternate for any console without a defined alternate or alternate console group.

Switching Console Attributes from One Console to Another

Operators can use the SWITCH command to switch console attributes between consoles. Using SWITCH might help to

- Handle message traffic during operator shifts
- Redistribute operator workload by rerouting messages.

Consoles can be MCS, SMCS or extended MCS consoles. Operators can use SWITCH to do the following:

- 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 in the alternate console group or the console specified on ALTERNATE.
- Restore the console attributes of an inactive console that has switched to an active alternate.

When an operator uses the SWITCH command to switch from a console to another console or its alternate, MVS first appends the console attributes with the other console and then deactivates the switched console.

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 the master 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, or define routing defaults with the MSGRT command.

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 3-5.

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 away from the master console. 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. They cannot, however, receive output from the TRACK 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 2-13.

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. One full-capability console is the master console; 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 3-2 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.

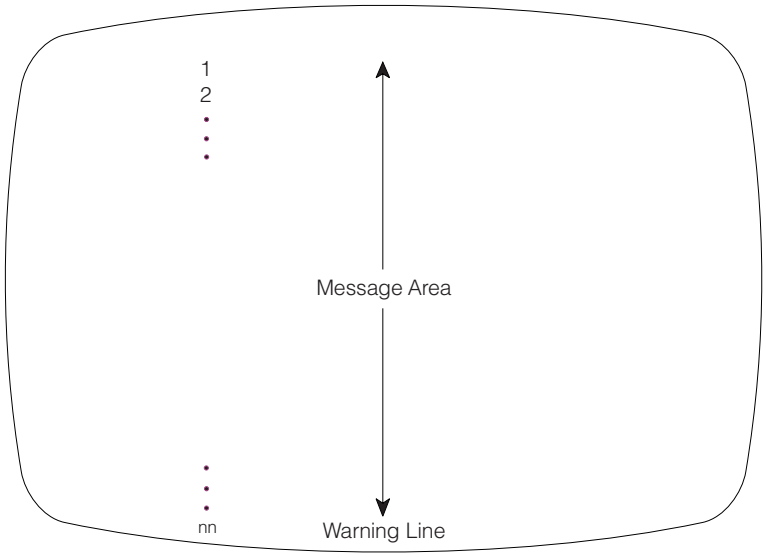


Figure 3-2. Format of a Console Screen in Message Stream Mode

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 2-13.

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. You cannot change a console to message stream or status display if it is the master console; you must first designate another console to take over the master console function.

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 3-2 on page 3-11.

Note: When you specify SD or MS for status display or message stream mode, the system considers the console to be an output-only console. If you then want to change any console characteristics using the VARY command, you must use the O-unit parameter of the VARY command (see “Controlling MCS and SMCS Consoles” on page 4-545). Otherwise, the VARY command is rejected.

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 console 8 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 the master console screen fills often 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 3-5 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 3-1 on page 3-5 shows the display that appears in response to this command.

Table 3-5. Message Routing Codes

Code	Definition
1	Master console action
2	Master console 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
29-40	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. You might want to direct only messages with routing codes 1, 2, 9, and 10 to the master console. The master console does not have to receive tape, DASD, or teleprocessing messages.

To learn what the routing codes for a console are, enter the DISPLAY CONSOLES command. Figure 3-1 on page 3-5 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 3-1 on page 3-5 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,HCONLY command.

To specify that the system console will not receive undelivered messages, enter:

```
VARY ,HARDCOPY,UD=Y
```

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 id 06, enter:

  ```
  CONTROL V,LEVEL(R,I),L=06
  ```

  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 then appear only on the system log, the hardcopy console, and any extended MCS consoles that are receiving the hardcopy message set. The system displays this message to warn you of the message loss:

  ```
  IEE828E SOME MESSAGES NOW SENT TO HARDCOPY ONLY
  ```

  Example 2

To assign to the console with console id 12 (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=12
```

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 2-6 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 2 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=2
  ```

  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 system log, the hardcopy console, 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 3-16.

You can activate the installation exit IEAVMXIT, if it is installed, by:

- **Using the CONTROL Command**:

  From the master console, 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 MFP
- The MPFLSTxx member that identifies the message ID, color attribute, or command installation exit definition

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=9A'),CON=N
```

then pressing PFK 5 has the same effect as typing DISPLAY U,L=09A and pressing the ENTER key.

On the other hand, if you specify conversational mode by entering:

```
CONTROL N,PFK=(5,CMD='D U,L=9A'),CON=Y
```

then pressing PFK 5 causes the command D U,L=09A 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 IPS=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 IPS=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 9 have the defaults that IBM supplies. These defaults are listed in Table 2-1 on page 2-9.

To define PFKs for your consoles, see "Defining PFKs Using PFK Tables" on page 3-19.

Processing Hardcopy

Logging provides a permanent record of system activity. Your installation can record system messages and, optionally, commands and command responses, by using either the system log (SYSLOG) or an MCS printer console. Your installation can also allow an extended MCS console to receive the same set of messages, by using the system log (SYSLOG), operations log (OPERLOG), or an MCS printer console. The system log or device that receives messages is called the hardcopy medium. The group of messages that is recorded is called the hardcopy message set.

In a JES3 system, the hardcopy message set is always sent to the system log. In a JES2 system, the hardcopy message set is usually sent to the system log but may be sent to a printer console, if the installation chooses. 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 can be printed at once by the hardcopy output device or directed to the system log. The system log is printed periodically.

Note: The term “hardcopy log” can refer to:

- The system log (SYSLOG)
- The operations log (OPERLOG)
- The device used to print hardcopy messages
- The data set containing hardcopy messages
- The actual printed copy of the hardcopy messages

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,HCONLY 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, excluding dynamic status displays, in the hardcopy message set.

CMDS The system includes all operator commands and their responses, including all status displays (both dynamic and static), 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 3-1 on page 3-5 shows the display that appears in response to this command.

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 a device, such as a printer console, or the system log (SYSLOG). 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 CONSOLxx, 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 hardcopy medium and SYSLOG fails, the system attempts to switch hardcopy processing to an appropriate printer console. If a suitable console is not active at the time of failure, hardcopy processing is suspended and you are notified through the master console.

Assigning the Hardcopy Medium

You assign the hardcopy medium by using the VARY command.

Use the HARDCPY operand on the VARY command to assign a device as the hardcopy medium. Only enter this command from a console with master authority. You can select any of the parameters shown in the syntax diagram for the VARY device,HARDCOPY command in chapter 4.

Example:

To specify that the hardcopy medium is an output-only device with device number 0E3, issue:

```
VARY 0E3,HARDCPY
```

The effect of this command lasts only for the duration of the IPL.

If the hardcopy medium is not active and you issue a VARY ddd,CONSOLE command to add a second active console to the system, MVS assigns a hardcopy medium. See the *MVS System Messages* books for the format of the output from the hardcopy medium.

To display information about the hardcopy medium, enter:

```
DISPLAY CONSOLES,HARDCOPY
```

The resulting display tells you whether the hardcopy medium is SYSLOG or a device, the criteria that have been defined by the installation for selecting messages for the hardcopy message set, and 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 perator 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 Table 4-2 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

© Copyright IBM Corp. 1988, 2002

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 [5]	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 an z/OS UNIX System Services process	MCS, SMCS or extended MCS consoles or job stream [5]	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 [5]	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 [5]	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 [5]	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	MCS and SMCS consoles	INFO
	Delete certain messages		INFO
	Halt printing of a status display		INFO
	Control area displays		INFO
	Remove information from the screens		INFO
	Activate, deactivate, or display the status of the action message retention facility		MASTER
	Change or display the number of allowed message and reply buffers		MASTER
	Change or display message deletion or format specifications		INFO
	Change or display the status of WTO user exit IEAVMXIT		MASTER
	Define commands for PFKs		INFO
	Reroute message queue		INFO or MASTER
	Change or display time interval for updating a display		INFO
	Change operating mode of console		INFO
	Select the message levels for a console		INFO
	Increase the RMAX value		MASTER
	In a sysplex, change the maximum time MVS waits before aggregating messages from routed commands		MASTER
DEVSERV (DS)	Display current status of devices and corresponding channel paths	MCS, SMCS or extended MCS consoles or job stream [5]	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 domain descriptor table 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 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 [5]	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 active workload management service policy and mode status for systems, or application environments Display information about registered products and the product enablement policy.	MCS, SMCS or extended MCS consoles or job stream [5]	INFO
DUMP	Request a dump of virtual storage to be stored in a SYS1.DUMP data set	MCS, SMCS or extended MCS console [5]	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 [5]	SYS
FORCE	Force termination of: • 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 [5]	MASTER
HALT (Z)	Record statistics before stopping the system (Must first stop subsystem processing with a subsystem command)	MCS, SMCS or extended MCS console [5]	SYS
IOACTION (IO)	Stop or resume I/O activity to DASD	MCS, SMCS or extended MCS console [5]	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 [5]	INFO
LOGOFF	To log off MCS and SMCS consoles	MCS and SMCS consoles	SYS
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 [5]	INFO
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 Start TSO/TCAM time-sharing once TCAM is active Stop TSO/TCAM time-sharing 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 Switch the workload management mode in effect for a system	MCS, SMCS or extended MCS consoles or job stream [5]	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 [5]	INFO

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
MOUNT (M)	Mount volumes	MCS, SMCS or extended MCS consoles or job stream [5]	I/O
MSGRT (MR)	Establish message routing instructions for certain options of DISPLAY, TRACK, STOPTR, MONITOR, STOPMN, CONFIG, and CONTROL commands Stop message routing Establish message routing for z/OS UNIX System Services information	MCS and SMCS Consoles	INFO
PAGEADD (PA)	Add local page data sets Specify data sets as non-VIO page data sets	MCS, SMCS or extended MCS console [5]	SYS
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 [5]	SYS
QUIESCE	Put system in MANUAL state without affecting step timing	MCS, SMCS or extended MCS console [5]	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 [5]	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 [5]	SYS
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 [5]	INFO

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
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 [5]	INFO
SET (T)	Add modules to, or delete modules from, the LPA dynamically. Change: • 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 Restart SMF or change SMF parameters by changing the active SMFPRMxx parmlib member Start or stop the common storage and tracking functions Start, refresh, or stop MMS. Update: • the APF list and dynamic exits • the format or contents of the APF list • the LNLKST set for LNKLST concatenation	MCS, SMCS or extended MCS consoles or job stream [5]	SYS

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
SETDMN (SD)	Respecify domain parameters	MCS, SMCS or extended MCS consoles or job stream [5]	SYS
SETETR	Enable external time reference (ETR) ports that have been disabled	MCS, SMCS or extended MCS console [5]	SYS
SETGRS	Migrate a currently active global resource serialization ring complex to a global resource serialization star complex	MCS, SMCS or extended MCS console [5]	SYS
SETIOS	Respecify, add, or delete MIH time intervals without changing the active IECIOSxx parmlib member	MCS, SMCS or extended MCS console [5]	SYS
SETLOAD	Switch dynamically from one parmlib concatenation to another without having to initiate an IPL	MCS, SMCS or extended MCS console [5]	SYS
SETLOGRC	Change the logrec recording medium	MCS, SMCS or extended MCS console [5]	SYS
SETOMVS	Change the options that OS/390 UNIX System Services uses.	MCS, SMCS or extended MCS console [5]	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 [5]	SYS
SETRRS CANCEL	End RRS processing	MCS, SMCS or extended MCS console [5]	SYS
SETSMF (SS)	Change SMF parameters without changing the active SMFPRMxx parmlib member	MCS, SMCS or extended MCS consoles or job stream [5]	SYS
SETSMS	Change SMS parameters without changing the active IGDSMSxx parmlib member	MCS, SMCS or extended MCS console [5]	SYS
SETSSI	Dynamically add, activate or deactivate a subsystem.	MCS, SMCS or extended MCS console [5]	SYS
SETXCF	Control the cross-system coupling facility (XCF)	MCS, SMCS or extended MCS console [5]	MASTER
SLIP (SL)	Set SLIP traps Modify SLIP traps Delete SLIP traps	MCS, SMCS or extended MCS consoles or job stream [5]	SYS

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group
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 [5]	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 [5]	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 [5]	INFO
STOPTR (PT)	Halt or reduce information displayed with the TRACK command	MCS and SMCS Consoles	INFO
SWAP (G)	Move a volume from one device to another	MCS, SMCS or extended MCS consoles [5]	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 Switch a consoles attributes to another console	MCS, SMCS or extended MCS console [5]	SYS MASTER
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 [5]	SYS MASTER MASTER SYS
TRACK (TR)	Periodically display job information on the display consoles Periodically display unit of work information executing for APPC/MVS transaction requestor Periodically display the number of z/OS UNIX address spaces	MCS and SMCS Consoles	INFO
UNLOAD (U)	Remove a volume from system use	MCS, SMCS or extended MCS consoles or job stream [5]	I/O
VARY (V)	Change the master console Control the hardcopy message set and the hardcopy medium. Change the status of a secondary console Change the console's alternate console group 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 (Note 4)
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 [5]	SYS

Table 4-1. System Command Summary (continued)

Command (Abbr)	Function	Acceptable From	Command Group

Notes:

1. CONS command group when message routing is specified.
2. For information about VTAM commands, see *VTAM Operation*.
3. For information about TCAM commands, see *ACF/TCAM Operation*.
4. This command is in a different command authority group depending on the parameters specified on the command. See Table 3-4 on page 3-6 for more information.
5. 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 Table 4-2 to help you with the syntax.

Table 4-2. System Command Syntax Notation

Notation	Meaning	Example	
		Book Syntax	Sample Entry
Apostrophes	Must be entered as shown.	`SEND 'message',NOW`	`SEND 'listings ready',NOW`
Comma	Must be entered as shown.	`DISPLAY C,K`	`DISPLAY C,K`
Ellipsis ...	The parameter can be repeated. Do not enter the ellipsis.	`VARY (devspec[,devspec]...),ONLINE`	`VARY (282,283,287),ONLINE`

Lower Case Parameter	A parameter must be substituted. You can enter the command and the parameter in either upper or lower case.	`MOUNT devnum`	`MOUNT A30`
			or
			`mount a30`

Or-bar (|)	You must enter one of the items. You cannot enter more than one.	`ACTIVATE|RECOVER=SOURCE`	`RECOVER=SOURCE`
Parentheses and special characters	Must be entered as shown.	`DUMP COMM=(text)`	`DUMP COMM=(PAYROLL)`
Single parameter in brackets	The parameter is optional. Multiple optional parameters may be illustrated adjacent or stacked vertically for space reasons.	`DISPLAY DMN[=domainum]`	`DISPLAY DMN=5`

Stacked items in braces	You must enter one of the items. You cannot enter more than one.	`MN {DSNAME}`
		`{SPACE }`
		`{STATUS}`
			`MN SPACE`

Stacked items with or-bars (|) and brackets	Optional, mutually exclusive parameters. Enter one or none.	`CD RESET [ ,SDUMP ]`
		`|,SYSABEND`
		`|,SYSUDUMP`
		`|,SYSMDUMP`
		`|,ALL`
			`CD RESET,SYSUDUMP`

Table 4-2. System Command Syntax Notation (continued)

Notation	Meaning	Example	
		Book Syntax	Sample Entry
Underline	If you do not enter one of the parameters, the system supplies the underlined parameter, which is the default	`K T [,REF ]` `[,UTME=nnn]`	`K T`

Upper Case Parameter	The parameter must be spelled as shown. You can enter the command and the parameter in either upper or lower case.	`DISPLAY SMF`	`DISPLAY SMF`
			or
			`display smf`

System Command Formats

Two system command formats are defined.

Typical Format

Most system commands can use the format shown in Figure 4-1.

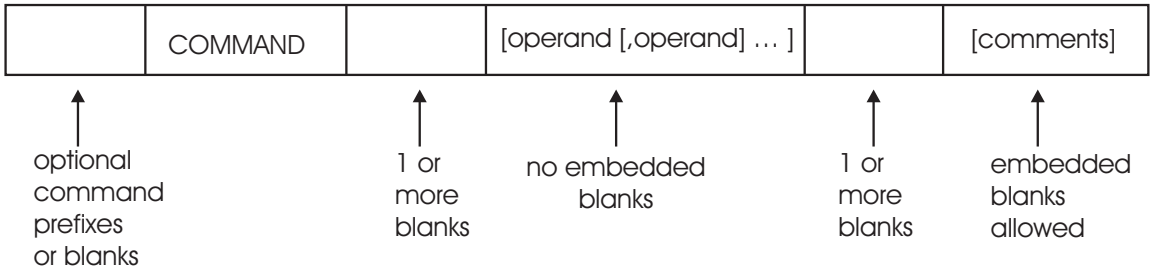


Figure 4-1. One System Command Format

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.

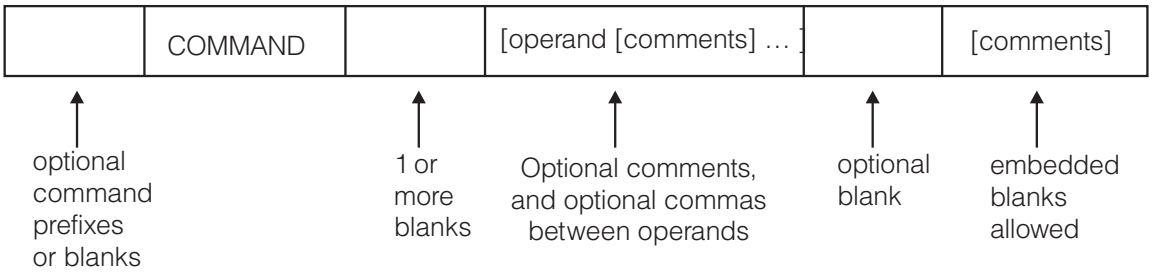


Figure 4-2. A Second System Command Format

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*.

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 (IODFxx) 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
  - 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.

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: • 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: • 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)
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**
• A job in execution • A running Advanced Program-to-Program Communication/MVS (APPC/MVS) transaction program • A started task	CANCEL jobname
• A time-sharing user	CANCEL U=userid
• A started task • A MOUNT command • An external writer allocation • The output processing for a job • An 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.

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 name of a started task is determined 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 provided with the JCL JOB statement.
- A ***procedure***, the system will use the member name as the job 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 SYSMDUMP) 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 an z/OS UNIX process.

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.

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.tl*`).

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.

```
DISPLAY OMVS,U=CHAD

BPXO001I 17.12.23 DISPLAY OMVS 700                                    C
OMVS     ACTIVE                                          BPXPRMHF
USER    JOBNAME  ASID        PID       PPID  STATE   START     CT_SECS
CHAD    CHAD     001D     262147          1  RI    17.00.10      1.203
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 SYSMDUMP. 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 SYSMDUMP 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.

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 SYSMDUMP from parmlib members IEAABD00, IEADMP00, and IEADMR00 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 SYSMDUMP” 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 SYSMDUMP” 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

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 IEADMR00 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 theIEAABD00, IEADMP00, and IEADMR00 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.

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.

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 *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU*.

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.

XESDATA
: XES-Related information in the sysplex.

SDATA options for SYSABEND dump type:

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.
SUM	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.)

SDATA options for SYSUDUMP dump type:

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.
SUM	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:

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:

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'.)

SYSMDUMP options:

ALL	Sets all of the other SYSMDUMP 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.
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 *z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU*.
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 IEADMR00 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 IEADMR00 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

Table 4-6. Example of How CHNGDUMP Commands Affect Dump Modes and Options (continued)

FUNCTION (* indicates operator commands/actions)	OPTIONS	MODE
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
* 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

Table 4-6. Example of How CHNGDUMP Commands Affect Dump Modes and Options (continued)

FUNCTION (* indicates operator commands/actions)	OPTIONS	MODE
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
```

```
Where s-option represents:

STRNAME=strname
  [,CONNAME=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

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=*nnnn***K** *or nnn***M**
Reserves storage for exclusive use by SVC dump. SVC dump can use this storage while it captures the contents of common area storage. The amount of storage that is set aside is expressed in kilobytes (0K to 9999K) or in megabytes (0M to 999M). The default is 0.

You can enter the BUFFERS option at any time to change the amount of storage reserved for use by SVC dump only. You should decide how much storage the installation needs to set aside for use by SVC dump, without seriously impacting system performance. Specifying a large BUFFERS value may degrade performance because the system reserves a large amount of storage for use by SVC dump.

SDUMP,MAXSPACE=*xxxxxxxx***M**
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.

CONNAME= *or* **CONNM=***conname*
When specified for a coupling facility cache structure, requests the user registry information for this user be included in the dump. *conname* is

the name of a connected user. If the connected user represented by the conname does not exist, the dump will not contain user registry information.

ACCESSTIME= *or* **ACC={ENFORCE** *or* **ENF** *or* **NOLIMIT** *or* **NOLIM}**
Indicates whether the dump time limit specified on the ACCESSTIME parameter of the IXLCONN macro is in effect. When ACCESSTIME=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 ACCESSTIME=0 is specified on the IXLCONN macro and ACCESSTIME=ENFORCE is specified on the dump request, the structure will not be included in the dump.

When ACCESSTIME=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.

When ENTRYDATA=SERIALIZE is specified, the entry data is dumped while serialization is held. If ACCESSTIME=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 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 ACCESSTIME are specified more than one time for a structure, the first CONNAME and the last ACCESSTIME 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:

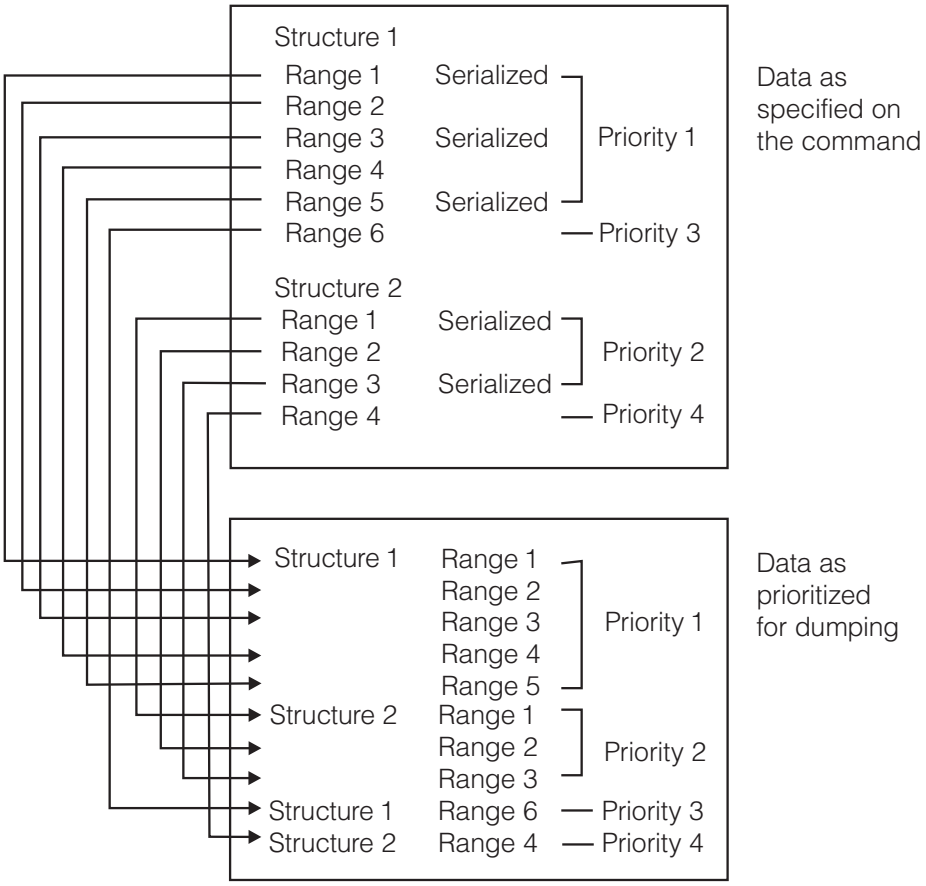
a. The system will attempt to dump the first requested structure first.
   1) Within that structure, the system processes the LOCKENTRIES, USERCNTLS, COCLASS, STGCLASS, and LISTNUM parameters in the order that they are specified. COCLASS, STGCLASS, and LISTNUM may be specified more than once for a single 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.

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.

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.
- **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 C1 commands may be essential to clearing a backlog of Class C2 commands.
4. Class C2 commands are ordinary attached commands that run in the CONSOLE address space.

For a list of the commands by class, see “Command Flooding” on page 1-34.

- **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.

CONFIG Command

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, Vector Facilities attached to online processors, storage amounts, storage ranges, central and expanded 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 ESTOR CONFIG STORAGE CONFIG VF	"Reconfiguring the System Directly"
CONFIG MEMBER	"Reconfiguring the System with a CONFIGxx Parmlib Member" on page 4-54
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, Vector Facilities attached to online processors, central storage ranges, amounts, and elements, and expanded storage 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, Vector Facilities, and Integrated Cryptographic Facilities (ICRFs), attached to online processors, storage amounts, storage ranges, central and expanded storage elements, logical partitions, or channel paths.

See *z/OS MVS Recovery and Reconfiguration Guide* 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}[,VFON  ]|,{OFFLINE|OFF}] }
                                        |,VFOFF

   {VF(x[,x]...)[,{ONLINE|ON}|,{OFFLINE|OFF}]                   }

   {{STORAGE|STOR}{(ddddM)      }[,{ONLINE|ON}|,{OFFLINE|OFF}]  }
                  {(ddddM-ddddM)}
                  {(E=id)       }

   {ESTOR(E=id)[,{ONLINE|ON}|,{OFFLINE|OFF}]                    }

   {CHP{(xx)    }[,{ONLINE|ON}[NOVARY]|,{OFFLINE|OFF}[,UNCOND]] }
       {(aa-bb) }                                      |,FORCE
       {(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 Vector Facilities and ICRFs attached to specified online processors.

(x[,x]...)
One or more processors identified by *x* (0 through F).

ONLINE *or* **ON**
The system is to bring the specified processor(s) online. If necessary, the system synchronizes the processor's TOD clocks. Each processor's Vector Facility remains in the state it was in when the system last varied the processor offline. If Integrated Cryptographic Service Facility/MVS (ICSF/MVS) is active, the system brings online any ICRF attached to each processor.

ONLINE,VFON
The system is to bring online the specified processor(s), and bring online the Vector Facility attached to each specified processor. If ICSF/MVS is active, the system brings online any ICRF attached to each processor.

ONLINE,VFOFF
The system is to bring online the specified processor(s), but keep offline the Vector Facility attached to each processor. If 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).

VF
The system is to reconfigure, both logically and physically, one or more Vector Facilities attached to online processors.

(x[,x]...)
One or more online processors identified by *x* (0 through F) that have Vector Facilities attached.

ONLINE *or* **ON**
The system is to bring online each specified Vector Facility.

OFFLINE *or* **OFF**
The system is to take offline each specified Vector Facility.

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.

dddddX
The amount of central storage to be reconfigured. Specify up to five 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 *ddddd* must be a multiple of the storage increment size (usually 2, 4, or 8), and cannot exceed 16383P.

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.

dddddX-*dddddX*
The starting and ending addresses of the central storage section to be reconfigured. Specify up to five decimal digits followed by a multiplier (M-megabytes, G-gigabytes, T-terabytes, P-petabytes) for each address. The value for each *ddddd* 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.

ESTOR

The system is to reconfigure expanded storage elements.

(E=id)

The expanded storage element to be reconfigured, identified by its storage element id.

The expanded storage element to be reconfigured, identified by its storage element id. If you specify E=0 in LPAR mode, the system is to reconfigure the initial allocation of the expanded storage element (as defined in the activation profile), unless the initial allocation is zero, in which case E=0 will reconfigure the reserved allocation. If you specify E=1, the system is to reconfigure the reserved allocation of the expanded storage element (as defined in the activation profile), unless the initial allocation is zero, in which case no storage will be reconfigured.

ONLINE *or* **ON**

The system is to bring the specified expanded storage element online. The system issues message IEE524I to display the ranges of expanded storage that come online. The system rejects the command if you specify an id for an expanded storage element that is not available to the system.

If the system cannot bring some of the requested storage online, message IEE578I displays the total amount of requested storage that is left offline.

OFFLINE *or* **OFF**

The system is to take the specified expanded storage element offline. The system issues message IEE510I to display the ranges of expanded storage that went offline with the expanded storage element. If the system cannot reconfigure the expanded storage element physically offline, the system rejects the command and issues message IEE148I.

Notes:

1. There can be a delay between the time you enter CONFIG ESTOR... OFFLINE and the time the system issues a message indicating the expanded 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 ESTOR...OFFLINE to take an expanded storage element offline, the system issues a display that tells what address ranges have been removed.
3. ESTOR processing is not applicable to the z/Architecture environment.

CHP

The system is to reconfigure one or more channel paths.

(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.

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

The command processor determines if taking the CHPID offline will cause a loss of consoles that should not be lost, for example, the MCS master console. If the MCS master console could be lost, the CONFIG command processor rejects the command. With SMCS consoles, there is no way to associate the CHPID to an LU so the command cannot detect if an SMCS console will be lost.

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.

Specifically, when the master console goes offline because of a CONFIG CHP OFFLINE command, a CONFIG CHP ONLINE command for the same channel path brings the device back online. The master console, however, remains the alternate console assigned when the master console went offline. The original master console remains an alternate console unless you again make it the master console by issuing the VARY CONSOLE and VARY MSTCONS commands.

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:

```
cf cpu(2),offline
```

Example 2

To bring online a storage range from real addressses 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

To bring processor 1 online with its Vector Facility, enter:

```
cf cpu(1),online,vfon
```

If the Vector Facility was online when the system last varied processor 1 offline, the following command would produce the same result:

```
cf cpu(1)
```

Example 8

To bring processor 2 online but keep offline the attached Vector Facility, issue:

```
cf cpu(2),online,vfoff
```

Example 9

To bring the Vector Facility attached to processor 3 online, issue:

```
cf vf(3),online or cf vf(3)
```

Example 10

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 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 and expanded 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, Vector Facilities and 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, Vector Facilities and ICRFs attached to the processors, total amount of central storage, central storage elements, expanded 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 and expanded 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|cc|cca|name|name-a}]
   {OFFLINE|OFF}
```

ONLINE *or* **ON**
The system is to display the system configuration so that you can decide which processors, Vector Facilities and ICRFs attached to the processors, central and expanded storage elements, and channel paths you want to bring online. The system brings online the processors, Vector Facilities and 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, Vector Facilities and ICRFs attached to the processors, central and expanded storage elements, and channel paths you want to take offline. The system takes offline the processors, Vector Facilities and 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, cc, cca, name or name-a*
The display area where the system is to display the system configuration. You can specify the display area by area (*a*), console id (*cc*), or both (*cca*); or, you can specify the display area by 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 (unless you specify routing instructions for messages IEE521I and IEE522D with the MSGRT command, as described later in this chapter).

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 and expanded 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 4-58
CONTROL C	"Deleting Retained Action Messages" on page 4-59
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 4-61
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 4-64
CONTROL M,APPLID	"Setting the APPLID of the System" on page 4-69
CONTROL M,GENERIC	"Setting or Turning Off the VTAM Generic Name for SMCS" on page 4-70
CONTROL M,LOGLIM	"Changing or Displaying the Number of Allowed WTL SYSLOG Buffers" on page 4-65
CONTROL M,MLIM	"Changing or Displaying the Number of Allowed WTO and WTOR Message Buffers" on page 4-65
CONTROL M,ROUTTIME	"Changing the Time the System Waits for ROUTE Command Responses" on page 4-66
CONTROL M,RMAX	"Increasing the Maximum Number of Reply IDs" on page 4-67
CONTROL M,UEXIT	"Changing or Displaying the Status of WTO Installation Exit IEAVMXIT" on page 4-68
CONTROL N,PFK	"Changing a PFK Definition" on page 4-71
CONTROL Q	"Rerouting Message Queues" on page 4-73
CONTROL S	"Changing or Displaying Message Deletion and Format Specifications" on page 4-74
CONTROL T	"Changing or Displaying Time Intervals for Dynamic Displays" on page 4-78
CONTROL V,USE	"Changing the Operating Mode of a Console" on page 4-79
CONTROL V,LEVEL	"Selecting the Message Levels for a Console" on page 4-80

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

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 Chapter 3, “Defining and Changing Console Characteristics” on page 3-1 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.

```
K A[,nn[,nn]...][,L={cc  }]
    |,NONE          {name}
    |,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 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 10.

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=*cc or 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, 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|cc|cca|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, cc, cca, name, or name-a*
The identification number or 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 number 20, enter:

```
k c,d,121,L=20
```

Example 2

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

Use the CONTROL D command to control displays, message numbering, and the PFK display line.

```
K D[,N[,HOLD]                      ]
    |{,F|,H|,U}[,L={a|cca|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.

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.

L=*a, cca, or name-a*
Specifies the display area (*a*), the console id and display area (*cca*), or the console name and display area (*name-a*) that is affected by this command.

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.

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|cc|cca|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, cc, cca, 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 id (*cc*), or both (*cca*); or, 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 the master console, or from a console with master authority.

The AMRF parameter on the INIT statement in the CONSOLxx 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 value of the CONTROL M,AMRF operand.

Example

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}]
    |,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 20 to 999999. In a sysplex, the WTL SYSLOG buffers hold messages that the system has not yet presented to the hard-copy device 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
The current values for CONTROL M operands are to be displayed in the entry area of a console with master authority. 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 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 the CONTROL M,MLIM and RLIM operands.

Example

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

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 value of the CONTROL M,ROUTTIME operand.

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 value of the CONTROL M,RMAX operand.

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 value of the CONTROL M,UEXIT operand.

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.

Displaying the SMCS APPLID of the current system and VTAM generic name for SMCS

Use the CONTROL M,REF command to display the SMCS APPLID of the current system and VTAM generic 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 SMCS APPLID of the current system and VTAM generic name for SMCS.

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 Name for SMCS

Use the CONTROL M,GENERIC command to set or turn off the VTAM generic 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 recyce 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 name for SMCS in the sysplex.

NONE
Turns off the VTAM generic 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
```

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 3-20.

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={cc|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=*cc or 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,PFK=(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,PFK=(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,PFK=(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,PFK=(5,CMD='')
```

Example 5

To assign the commands in the table PFK22 to cons8, enter:

```
K N,PFK=PFK22,L=CONS8
```

Note: The CONTROL N,PFK command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

Rerouting Message Queues

Use this command, which is intended for use in error situations, to route a console's messages to another console or to the system log. It affects only messages currently on the console's queue. Subsequent messages are queued as usual.

The command gives you more control over MCS and SMCS console message queues. If there is a WTO buffer shortage, you can re-queue 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 rerouted are:

- Action messages. If they are being queued for display on only one console, they might subsequently appear only on the system log.
- All in-line commands queued for a particular console.
- WTOR messages and unconditional messages.

The kinds of messages that are not rerouted are:

- Out-of-line messages. You can delete these with K E,D.
- Messages queued to SYSLOG.
- Messages on the hard-copy device. First, you must use the VARY command to direct hardcopy to another device. Then, you can re-queue the messages being routed to the hardcopy device.

In addition to being routed to another console with the UD attribute, WTORs and messages queued unconditionally are also routed to the master console (unless the specified device is the master console, in which case, they are already there). The successful completion message IEE188I is explicitly sent to both the console from which the messages came and the console to which the messages are rerouted.

Any in-line multiple line message that is being presented on only one console is terminated to allow a console in a "hung endline" situation to continue processing. To end a "hung endline" situation, issue the CONTROL Q,R command for the affected console. This allows backed-up messages to appear.

The syntax is:

```
K Q[,R={dd  }][,L={cc|name}]
       {name}
       {HC  }
```

Q The CONTROL command is to affect a console's message queue.

R The message queue is to be rerouted.

dd or name
The id or name of the console that is to receive the rerouted messages.

HC
The hardcopy device is to receive the rerouted messages.

L=*cc or name*
The id or name of the console whose message queue is to be rerouted. You cannot specify the hardcopy device. If this operand is omitted, the message queue of the console from which the K Q command is entered is rerouted.

Example 1

To reroute any messages on the issuing full-capability console's queue to console BACKUP, enter:

```
CONTROL Q,R=BACKUP
```

Example 2

To reroute any messages on console CON1's queue to console CON2, enter:

```
CONTROL Q,R=CON2,L=CON1
```

Example 3

To reroute any messages on the issuing full-capability console's queue to console 03 and to the system log, enter:

```
CONTROL Q,R=03
```

Notes:

1. Using K Q can cause duplicate lines to appear on hardcopy, the receiving console, and/or the master trace table, if master trace is active. A multiple line message might be re-displayed with some of the lines missing.
2. The CONTROL Q command has no effect on extended MCS consoles or on system consoles and is not valid for managing these consoles.

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 inl termwidth=″medium″>

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={cc|name}]
```

S The current console specifications are to be temporarily altered or referenced.

CON=
Conversational message deletion is requested or cancelled.

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 cancelled. Messages must be removed manually.

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. WTORs and action messages are also overlaid. Extended MCS consoles do not support wrap mode.

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=*cc or 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.
- You can specify, for *cc* or *name*, the identifier or name of a full-capability console only from a console with master authority or a pseudo-master console.
- You can't specify, for *cc* or *name*, the identifier or name of a status display console.
- If you specify, for *cc* or *name*, the identifier or name of a message stream console, you can't specify DEL=Y or DEL=N, CON=N, or, if the identifier or name is for a non-display console, any operand other than MFORM.

Note: Do not specify the system console as the target console (on the L= parameter) when issuing the K S,MFORM=option(s),L= command. The system rejects this command and displays message IEE926I.

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:

```
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 master console:

```
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

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:

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=*cc or 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={cc  }]
   |                                            {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.

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=*cc or name*
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={cc  }]
   |                             {name}
   |,LEVEL=(type[,type]...)
```

The parameters are:

V,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 messages routed to the console including broadcast messages. When a console is initialized, the ALL option is in effect.
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 an informational message is not directed to any console, it is routed to the system log.
2. If you don't specify NB, your console receives broadcast messages.
3. If a WTOR or action message is not directed to any console, it is routed to the consoles with the UD attribute.
4. If a message is directed to a specific console, it appears there regardless of the message level of the console.
5. If you specify a second K V,LEVEL command, the K V,LEVEL command in effect is cancelled.
6. 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=*cc or name*
The console where the specified action is to take place. The issuing console is the default. An operator at one full-capability console cannot change the message level of another full-capability console.

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)
```

Example 4

To suppress all broadcast and informational messages destined for the issuing console, enter:

```
K V,LEVEL=(NB,UNCOND)
```

This example is of special interest to those installations that use consoles to print tape labels. Here, broadcast messages and informational messages with certain routing codes will be sent directly to the hardcopy medium without appearing at the issuing console. If your consoles are set up so that informational messages with

certain routing codes or broadcast messages are not sent to a console, the system executes the command and sends you a warning message. It then issues message IEE250I to inform you of the message loss at your consoles.

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 DTYPE 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

Additionally, you may 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

Alternatively, you may 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

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.

Syntax

The syntax for the DEVSERV command is:

```
DS {PATHS|P},[/]devnum[,nn][,ONLINE|,ON   ][,NOSYM|,NOS][,DUMP]
   {SMS|S  }               |,OFFLINE|,OFF

         [,L={a|cc|cca}]


   {QDASD | QD}[,? |

     [ [ [,ccuu  [,1] | ,VOL=volser ] [ [,UCB] [,DCE][,SSSCB] [,DPCT]
        [ [,NOIO] | [,RDC] [,RCD] [,SNSS] ] ] ] | ,ccuu,nnn |
        [,ccuu,nnn] ,VOL=volser |
                    ,MACH=[mmpp-sssss | XXXX-sssss] |
                    ,SSID=[ssid | ALL] |
                    ,TYPE=[type | ALL] ]
     [,ONLINE] [,OFFLINE] [,DEFINED] [,CHKFAIL] [,VALIDATE] [,TOTALCYL]


   {QTAPE | QT}[,? |

     [ [ [,ccuu  [,1] ] [ [,UCB] [,DCE][ [,NOIO] | [,RDC]
        [,RCD] ] ] ] | ,ccuu,nnn |
        [,ccuu,nnn] ,LIB=libid | ALL |
                    ,MACH=[mmpp-sssss | XXXX-sssss] |
                    ,TYPE=[type | ALL] ]
     [,ONLINE] [,OFFLINE] [,DEFINED]


   {QPAVS | QP},{dddd}
                {dddd,nn}
                {SSID=ssid}
                {dddd,{VOLUME | UCB | UNBOX}}
```

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, cc, or cca*
The display area (*a*), console (*cc*), or both (*cca*) indicate where to present the display. 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.

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.

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.

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-ssssss
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-ssssss**. 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 **ssssss**, 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.

UNBOX
Causes QPAVS to unbox the unbound alias device if it is in a BOX state.

QPAVS Output Formats

When UNBOX *is not* specified (see Example 13) the format is:

```
IEE459I           DEVSERV QPAVS
     HOST                              SUBSYSTEM
 CONFIGURATION                        CONFIGURATION
 -------------                      ----------------------
 UNIT                                     UNIT    UA
 NUM. UA  TYPE         STATUS      SSID  ADDR.   TYPE
 ---- --  ----         ------      ----  ----    --------
 dddd aa  BASE         INV-ALIAS   ssss   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
   dddd(rc) dddd(rc) dddd(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
  - ssss = the SSID of the subsystem where device dddd belongs
  - uu = the unit address from the subsystem configuration
  - BASE = the unit is a BASE
  - 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

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  ,O,XP0101,ENABLED    SXP01     ,ENABLED
    431 ,3380  ,A,XP0201,ENABLED    SXP02     ,ENABLED
    432 ,3380  ,A,XP0202,ENABLED    SXP02     ,ENABLED
    433 ,3380  ,O,XP0301,ENABLED    SXP03     ,ENABLED
    434 ,3380  ,O,XP0302,ENABLED    SXP03     ,ENABLED
    435 ,3380  ,O,XP0303,ENABLED    SXP03     ,ENABLED
    436 ,3380  ,O,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

    16.24.41          IEE459I 16.24.41 DEVSERV PATHS 572
    UNIT DTYPE    M   CNT  VOLSER     CHPID= PATH STATUS
    430, 3380D,   O,  000, XP0101,    25=+ 2E=+

    ********************** SYMBOL DEFINITIONS **********************
    O = ONLINE                              + = PATH AVAILABLE
```

Issuing **DEVSERV S,430** produces this display:

```
16.24.26           devserv s,430

16.24.26           IGD001I 16:24:26 DEVSERV SMS 569
UNIT,      TYPE     ,M,  VOLUME,   VOLSTAT         STORGRP,  SGSTAT
430, 3380       ,O,  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

  16.24.41         IEE459I 16.24.41 DEVSERV PATHS 572
  UNIT DTYPE   M   CNT  VOLSER    CHPID= PATH STATUS
                   RC   TC   DFW  PIN  DC-STATE   CCA   DDC  ALT  CU-TYPE
  430, 3380D,  O,  000, XP0101,   25=+ 2E=+
                   IE   YY   NN   N    SIMPLEX    C0    01        3990-3
  431, 3380E,  O,  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 ,O,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   PSECONDRY 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.

a. 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
 CONFIGURATION                        CONFIGURATION
 -------------                     ----------------------
 UNIT                                    UNIT    UA
 NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
 ---- --  ----        ------     ----  ----    --------
 D222 22  BASE                   0102   22     BASE
 D2FE FE  ALIAS-D222             0102   FE     ALIAS-22
 D2FF FF  ALIAS-D222             0102   FF     ALIAS-22
****      3 DEVICE(S) MET THE SELECTION CRITERIA
```

DS QP,E27B,UCB

```
IEE459I 08.03.55 DEVSERV QPAVS 920
     HOST                              SUBSYSTEM
 CONFIGURATION                        CONFIGURATION
 -------------                     ----------------------
 UNIT                                    UNIT    UA
 NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
 ---- --  ----        ------     ----  ----    --------
 E27B 7B  ALIAS-E200             3205   7B     ALIAS-00
  UCB AT V02336568
0088FF04E27B0000 0000000008E4C3C2 3010200F00336541 0000000000000000
0000000000100000 00F451F0023362A8 0080000000000000
  UCB PREFIX AT V023BF6E0
0004004000000000 0000000000013920 289C0DB2B00080F0 0A0B1213FFFFFFFF
0108000000000001
  UCB COMMON EXTENSION AT V02336540
00000900202A0000 023BF6E000000000 0000000000FCD31C 00F4539000000000
****      1 DEVICE(S) MET THE SELECTION CRITERIA
```

DS QP,E279,2

```
IEE459I 08.23.08 DEVSERV QPAVS 952
     HOST                              SUBSYSTEM
 CONFIGURATION                        CONFIGURATION
 -------------                     ----------------------
 UNIT                                    UNIT    UA
 NUM. UA  TYPE        STATUS     SSID  ADDR.   TYPE
 ---- --  ----        ------     ----  ----    --------
 E279 79  ALIAS-E200             3205   79     ALIAS-00
**** 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.
```

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 and expanded storage, workload management service policy and mode 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 4-147	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
“Displaying CONTROL Command Functions” on page 4-109	DISPLAY C,K
“Displaying Attached Coupling Facility Information” on page 4-110	DISPLAY CF
“Displaying Console Group Definitions” on page 4-110	DISPLAY CNGRP
“Displaying Console Status Information” on page 4-111	DISPLAY CONSOLES
“Displaying Data Lookaside Facility Information” on page 4-116	DISPLAY DLF
“Displaying the Domain Description Table” on page 4-118	DISPLAY DMN
“Displaying Dump Options or Dump Data Set Status” on page 4-119	DISPLAY DUMP
“Displaying the Timer Synchronization Mode and ETR Ports” on page 4-128	DISPLAY ETR
“Displaying Extended MCS Information” on page 4-123	DISPLAY EMCS
“Displaying Global Resource Serialization Information” on page 4-129	DISPLAY GRS
“Displaying TSO/E Parmlib Information” on page 4-139	DISPLAY IKJTSO
“Displaying I/O Configuration Information” on page 4-141	DISPLAY IOS,CONFIG
“Displaying Dynamic Channel Path Management Information” on page 4-141	DISPLAY IOS,DCM
“Displaying IOS Group Information” on page 4-141	DISPLAY IOS,GROUP
“Displaying MIH and I/O Timing Limits” on page 4-142	DISPLAY IOS,MIH
“Displaying the Devices Stopped by the IOACTION Command” on page 4-145	DISPLAY IOS,STOP
“Displaying IPL Information” on page 4-146	DISPLAY IPLINFO
“Displaying System Activity” on page 4-147	DISPLAY JOBS *or* DISPLAY J *or* DISPLAY A *or* DISPLAY TS
“Displaying Library Lookaside Information” on page 4-158	DISPLAY LLA
“Displaying the System Logger and its Log Streams” on page 4-160	DISPLAY LOGGER
“Displaying the Logrec Recording Medium” on page 4-162	DISPLAY LOGREC
“Displaying System Configuration Information” on page 4-163	DISPLAY M

Table 4-10. Summary of the DISPLAY Command (continued)

Topic	Command
“Displaying MVS Message Service Status and Languages” on page 4-169	DISPLAY MMS
“Displaying Message Suppression, Retention, Color, Intensity, and Highlighting Options” on page 4-170	DISPLAY MPF
“Displaying z/OS UNIX System Services Status” on page 4-171	DISPLAY OMVS
“Displaying Sysplex-Wide Operator Information” on page 4-181	DISPLAY OPDATA
“Displaying PARMLIB Information” on page 4-182	DISPLAY PARMLIB
“Displaying Commands Defined for PFKs” on page 4-185	DISPLAY PFK
“Displaying Registered Products” on page 4-186	DISPLAY PROD
“Displaying Entries in the List of APF-Authorized Libraries” on page 4-187	DISPLAY PROG,APF
“Displaying Dynamic Exits” on page 4-188	DISPLAY PROG,EXIT
“Displaying LNKLST Information” on page 4-191	DISPLAY PROG,LNKLST
“Displaying LPA Information” on page 4-192	DISPLAY PROG,LPA
“Displaying System Requests” on page 4-193	DISPLAY R
“Displaying RTLS Information” on page 4-199	DISPLAY RTLS
“Displaying SLIP Trap Information” on page 4-202	DISPLAY SLIP
“Displaying SMF Data” on page 4-203	DISPLAY SMF
“Displaying Storage Management Subsystem Information” on page 4-203	DISPLAY SMS
“Displaying Information about All Subsystems” on page 4-211	DISPLAY SSI
“Displaying Static System Symbols” on page 4-213	DISPLAY SYMBOLS
“Displaying the Local and Greenwich Mean Time and Date” on page 4-214	DISPLAY T
“Displaying Component or Transaction Trace Status” on page 4-214	DISPLAY TRACE
“Displaying System Activity” on page 4-147	DISPLAY TS
“Displaying Device Status and Allocation” on page 4-218	DISPLAY U
“Displaying Workload Manager Information” on page 4-221	DISPLAY WLM
“Displaying Cross System Coupling Facility (XCF) Information” on page 4-225	DISPLAY XCF

Some uses of the DISPLAY command are described in other books. They are:

- TCAM teleprocessing functions and activity. See *ACF/TCAM Operation*.
- TCPIP activity and functions. See *z/OS Communications Server: IP Configuration Reference*.
- 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 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	All
DISPLAY PFK	Has sysplex scope only when you specify CN=.
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

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|cc|cca|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.

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 xvii) 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 xvii) 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 xvii) 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=*lluname*
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
```

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=*.

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, cc, cca, name, or name-a*

Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see “MSGRT Command” on page 4-319 later in this chapter).

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|cc|cca|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 xvii) 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.

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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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|cc|cca|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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will appear. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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
```

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|cc|cca|name|name-a}]
```

C,K
A summary of CONTROL command operands is to be displayed.

L=*a, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will appear. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

Example 1

To display the CONTROL command operands and their functions in display area A of console 5, enter:

```
D C,K,L=5A
```

Example 2

DISPLAY C,K Command

To display a summary of CONTROL command operands in display area A of console 10, enter:

```
D C,K,L=10A
```

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]...)]
```

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.

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.

```
D CNGRP[,{GROUP|G}[=(name[,name]...)]][,L={a|cc|cca|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, cc, cca, name, or name-a*

Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will appear. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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 CONSOLES EMCS command to display information for extended MCS consoles.

See Figure 3-1 on page 3-5 and Figure 2-2 on page 2-21 for examples of the resulting display of the DISPLAY CONSOLES and DISPLAY CONSOLES,BACKLOG commands. The syntax of the command is:

DISPLAY CONSOLES Command

```
D {CONSOLES},{{ACTIVE|A }[,CA={name            }][,ROUT={NONE|ALL|rr }] }
  {C        } {{SS       }     {(name[,name]...)}        {(rr-ss)     }  }
              {{NACTIVE|N}[MSTR]                         {(rr[,ss]...)}  }
              {                                  [,SYS=system name]     }

              {KEY[=key]                                                }

              {{BACKLOG|B}                                              }

              {{MASTER|M}[,SYS=system name]                             }

              {{MCONLY}                                                 }

              {*                                                        }

              {{LIST|L}                                                 }

              {{HARDCOPY|HC}[,SYS=system name]                          }

              {CN={xx          }[,ROUT={NONE|ALL|rr }]                  }
              {   {(xx[,yy]...)}       {(rr-ss)     }                  }
              {   {(xxx-yyy)   }       {(rr[,ss]...)}                  }
              {                 [,SYS=system name]                      }

              {HCONLY                                                   }

              {U={([/]devnum1[,[/]devnum2]...)}                         }
              {  {([/]lowdevnum-[/]highdevnum)}                         }
              {  {[/]devnum                   }                         }
              {MSTR[,SYS=system name]                                   }
              {SMCS                                                     }

     [,L={a|cc|cca|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 xvii) 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.

NACTIVE *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.

MCONLY
The system is to display the status of the current master console, if any. If the system is a member of a sysplex, display the master console for the sysplex.

CA
The system is to display, for a sysplex, the console/system association list to match the specified keyword, ACTIVE or NACTIVE.

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=, HCONLY, and MCONLY.

MSTR
The system displays the status of the master console and 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=
The system is to display the status of a console or consoles identified by console ID or console name.

xx The system is to display the status of the console with an id of *xx.*

(xx[,yy]...)
The system is to display the status of all consoles with IDs listed as *xx, yy,* and so forth.

(xxx-yyy)
The system is to display the status of all consoles with IDs in the range of *xxx* through *yyy*. For MCS consoles xxx and yyy are values in the range 1 through 99. For EMCS consoles that have migration IDs the values of xxx and yyy are in the range 100 through 250 except number 128, which is reserved.

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 the master console and 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.

HCONLY
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, OPERLOG, or a device
- 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, indicating whether the SYSLOG and the OPERLOG have the UD attribute.

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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where to present the display. For *cc*, you must specify a decimal number from 1 to 99.

If you omit this operand, the system presents the display in the first available display area or the message area of the console through which you enter the command (unless routing instructions are in effect; see the MSGRT command later in this chapter).

Example 1

To display information about all consoles with IDs in the range of 5 to 20, enter:

```
D C,CN=(5-20)
```

Example 2

To display information about consoles named CON24 and TAPE, enter:

```
D C,CN=(CON24,TAPE)
```

Example 3

To display information about all active consoles that receive messages with routing code 3, enter:

```
D C,A,ROUT=3
```

Example 4

To display information about all subsystem consoles that receive routing code 15, enter:

```
D C,SS,ROUT=15
```

Example 5

To display information about the console device number 81B, enter:

```
D C,U=81B
```

Example 6

To display information about the console device number 3480, enter:

```
D C,U=/3480
```

Example 7

DISPLAY CONSOLES Command

To display the list of available class names (keys), enter:

```
D C,KEY
```

Example 8

To display information about all inactive consoles, and have the output go to area A on the console named MASTER, enter:

```
D C,N,L=MASTER-A
```

Example 9

To display information about hardcopy processing on console CON13, area B, enter:

```
D C,HC,L=CON13-B
```

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|cc|cca|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=(SYSZSDO,*)
   ```
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 requestors. 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:
- 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 requestors, 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), (#, @, 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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

Example 1

To display resource information about all resources that have requestors, 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
EEECEDC
2823337
```

Displaying the Domain Description Table

Important
Beginning with z/OS V1R3, WLM compatibility mode is no longer available. As the DISPLAY DMN 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 DISPLAY command to obtain both the domain information and the name of the current installation performance specification (IPS). A domain is a group of users whose characteristics are defined by the installation performance specification (IPS). There may be from 2 to 129 domains in the system.

Note: The DISPLAY DMN command is not valid on systems operating in workload management goal mode. The command is supported on systems operating in workload management compatibility mode.

```
D DMN[=domainnum][,L={a|cc|cca|name|name-a}]
```

DMN
The domain description table is to be displayed (message IEE796I).

domainnum
A specific domain table entry (0-128) is to be displayed.

L=*a, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

Example 1

To display the name of the current IPS and information from domain 5, enter:

```
DISPLAY DMN=5
```

Example 2

To display the name of the current IPS and information from all entries in the domain description table, enter:

```
D DMN
```

The following information is displayed:
- Domain number
- Minimum and maximum multiprogramming levels
- Multiprogramming level swap-in and swap-out targets
- Current multiprogramming level
- Average number of ready users
- Number of swappable users currently in storage
- Number of non-swappable users
- Number of ready users currently swapped out of storage
- Time weighted service rate
- Contention index

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|cc|cca|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:

- 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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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)
```

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.

DISPLAY EMCS Command

The syntax for the DISPLAY EMCS command is:

```
D EMCS,{SUMMARY|S                }[,L={a|cc|cca|name|name-a}]
       {INFO|I                   }
       {FULL|F                   }

       {STATUS=A|N|L|B[{nn}]|ERR }
       {ST                       }

       {CN=consname|*            }

       {SYS=sysname|*            }

       {KEY=keyname|*            }

       {AUTH={ANY}               }
             {MASTER}
             {SYS}
             {IO}
             {CONS}
             {ALL}
             {INFO}
             {SYSONLY}rbrc;
             {IOONLY}
             {CONSONLY}
             {ALLONLY}
             {INFOONLY}

       {ATTR={ANY}               }
             {YES}
             {ROUT}
             {UD(YES|NO)}
             {HC}
             {AUTO(YES|NO)}
             {MN}
             {NONE}

       {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.

The following keyword parameters define the criteria used to limit the number of consoles displayed.

STATUS|ST=A|N|L|B[(*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, UD messages, hardcopy messages, AUTO(YES) messages, or MONITOR messages.)

ROUT
Consoles that receive any routing codes.

UD[(*YES***|***NO***)]**
Consoles that are or are not receiving UD messages. The default is YES.

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.

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|cc|cca|name|name-a}]
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the system is to present the display. For *cc*, you must specify a decimal number from 1 to 99.

If you omit this operand, the system presents the display in the first available display area or the message area of the console through which you enter the command (unless routing instructions are in effect; see the MSGRT command later in this chapter).

Example

Assume a single system where the system console is named SYS01, and there are two 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 looks like:

```
IEE129I 12.07.47 DISPLAY EMCS        FRAME LAST   F      E   SYS=SYS01
DISPLAY EMCS,SUMMARY
NUMBER OF CONSOLES MATCHING CRITERIA: 3
SYS01    EMCS1    EMCS2
```

```
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:

```
IEE130I 12.08.21 DISPLAY EMCS        FRAME LAST   F      E   SYS=SYS01
DISPLAY EMCS,INFO
NUMBER OF CONSOLES MATCHING CRITERIA: 3
CN=SYS01    STATUS=A    CNID=01000001 MIGID=100 KEY=SYSCONS
  SYS=SYS01    ASID=000A JOBNAME=--------- JOBID=---------
  UD=N HC=N AUTO=N DOM=NORMAL
  MONITOR=---------
  CMDSYS=SYS01
  ALTGRP=--------- LEVEL=ALL,NB       AUTH=MASTER
  ROUTCDE=NONE
CN=EMCS1    STATUS=A    CNID=02000001 MIGID=101 KEY=EXAMPLE
  SYS=SYS01    ASID=0019 JOBNAME=EXTMCS   JOBID=STC00008
  UD=N HC=N AUTO=N DOM=NORMAL
  MONITOR=---------
  CMDSYS=SYS01
  ALTGRP=--------- LEVEL=ALL          AUTH=MASTER
  ROUTCDE=ALL
```

DISPLAY EMCS Command

```
CN=EMCS2    STATUS=A    CNID=03000001 MIGID=102 KEY=EXAMPLE
  SYS=SYS01    ASID=001C JOBNAME=EXTMCS   JOBID=STC00009
  UD=N HC=N AUTO=N DOM=NORMAL
  MONITOR=---------
  CMDSYS=SYS01
  ALTGRP=--------- LEVEL=ALL          AUTH=MASTER
  ROUTCDE=ALL
```

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 complete syntax for the DISPLAY ETR command is:

```
D ETR[,DATA][,L={a|cc|cca|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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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.

DISPLAY GRS Command

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][,LATCH[,{JOBNAME|JOB}=jobname]][,HEX]}

    {,LATCH,{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|cc|cca|name|name-a}]
```

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 will show whether or not there is a functioning CTC link between this system and the system whose information is being displayed.
  - The display for a star complex will show the coupling facility structure name for the global resource serialization lock structure, ISGLOCK.
  - When the global resource serialization ring complex is using XCF signalling, the display shows that XCF paths are used.

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:

- 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

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 requestors. 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)

‘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 requestors, 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 #.

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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will appear. For *cc*, you must specify a decimal number from 1 to 99.

If you omit this operand, the display appears in the first available display area or the message area of the console where you entered the command (unless routing instructions are in effect; see the MSGRT command later in this chapter).

Example 1

To display resource information about all resources that have requestors, 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          FRAME  1     F      E   SYS=S3
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          FRAME LAST   F      E   SYS=S3
S=SYSTEMS GRJGQE07 TESTCASE#SERIALIZATION#ENQ
          CDDCDCFF ECEECCEC7ECDCCDCECECDD7CDD
          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
    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    FRAME LAST  F   E SYS=S1
   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.BRODCAST
   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]
```

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

TRANSREC
TRANSMIT/RECEIVE command options and defaults

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 TSO/E PARMLIB SETTINGS :
SYS1.PARMLIB(IKJTSOA1) 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.BRODCAST)
          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|cc|cca|name|name-a}]
```

IOS,CONFIG
The system displays information (message IOS506I) about the I/O configuration.

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.

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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will appear. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

Displaying Dynamic Channel Path Management Information

Use the DISPLAY IOS,DCM command to display the current status of dynamic channel path management.

```
D IOS,DCM
```

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
```

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.

Note: The display, which is delivered by the WTO facility, is limited to 256 lines of output. If the limit is reached, the display ends with a period and there is no indication that the information has been truncated. Use the TIME or 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}                                                    ]

     [,L={a|cc|cca|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.

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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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.
```

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 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|cc|cca|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, cc, cca, name, or name-a*

Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter.)

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 IEASYS*xx* and IEASYM*xx*
- LOAD*xx* 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|cc|cca|name|name-a}]
```

L=*a, cc, cca, name, or name-a*

Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display is presented. For *cc*, you must specify a decimal number from 1 to 99.

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 entered the command (unless routing instructions are in effect; see the MSGRT command).

Example 1

Assume a system has the following characteristics:

- IPL occurred on September 26, 2000 at 8:30 a.m.
- The release level is OS/390 2.6.0
- Parmlib member LOADJH, which resides on device 980, was used to IPL
- The IODF device resides on 224
- The IPL device is 980 and its volume serial is DR260B
- IEASYMKP is the symbol table
- The system treats MTL tape devices as standalone drives
- The operator entered IEASYSAB and IEASYSAC as the system parameters.

Issue the following command:

```
D IPLINFO
```

The system returns the following display, slightly modified to improve readability:

```
D IPLINFO
IEE254I 16.41.05 IPLINFO DISPLAY 048
SYSTEM IPLED AT 08.30.10 ON 09/26/2000
RELEASE OS/390 02.06.00
USED LOADJH IN PARMLIB ON 0980
MTLSHARE = Y
IEASYM LIST = KP
IEASYS LIST = AB, AC (OP)
IODF DEVICE 0224
IPL DEVICE 0980 VOLUME DR260B
```

Example 2

Assume a system has the following characteristics:

- IPL occurred on September 26, 2000 at 8:30 a.m.
- The release level is OS/390 2.6.0
- Parmlib member LOADJH, which resides on device 980, was used to IPL

- The IODF device resides on 224
- The IPL device is 980 and its volume serial is DR260B
- There was no IEASYM*xx* member specified
- The system treats MTL tape devices as MTL resident drives
- IEASYSKP was specified in LOADJH.

Issue the following command:

```
D IPLINFO
```

The system returns the following display, slightly modified to improve readability:

```
D IPLINFO
IEE254I 16.41.05 IPLINFO DISPLAY 048
SYSTEM IPLED AT 08.30.10 ON 09/26/2000
RELEASE OS/390 02.06.00
USED LOADJH IN PARMLIB ON 0980
MTLSHARE = N
IEASYM LIST = KP
IEASYS LIST = AB,AC(OP)
IODF DEVICE 0224
IPL DEVICE 0980 VOLUME DR260B
```

Example 3

Assume a system has the following characteristics:

- IPL occurred on September 1, 2000 at 9:00 a.m.
- The release level is OS/390 2.10.0
- LOADN7 in SYS0.IPLPARM, which resides on device BDA8, was used to IPL
- The system is running in z/Architecture mode
- The IODF device resides on BDA8
- The IPL device is BA9E and its volume serial is D83RLA
- The system treats MTL tape devices as MTL resident drives

Issue the following command:

```
D IPLINFO
```

The system returns the following display, slightly modified to improve readability:

```
D IPLINFO
IEE254I  18.19.23 IPLINFO DISPLAY 411
SYSTEM IPLED AT 09.00.48 ON 09/01/2000
RELEASE OS/390 02.10.00
USED LOADN7 IN SYS0.IPLPARM ON BDA8
ARCHLVL = 2  MTLSHARE = N
IEASYM LIST = (64,ME,N7)
IEASYS LIST = (64) (OP)
IODF DEVICE BDA8
IPL DEVICE BA9E VOLUME D83RLA
```

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 xvii) or use the *MVS System Messages* books to see the display. The syntax of this command is:

DISPLAY JOBS or J or A or TS Command

```
D {JOBS|J|A|TS}[,{LIST|L},[USERID=userid]]
               |,{ALL|A}
               |,{jobname[.identifier]|(jobname)}

   [,L={a|cc|cca|name|name-a}]
```

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 or 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 4-153 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 4-156 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 4-153 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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the description of the MSGRT command in this chapter).

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: • 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: • Userid • Address space status
A	List of all active units of work, including: • 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

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: • 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)
	Note: Beginning with z/OS V1R3, WLM compatibility mode is no longer available. The following information has been left here for reference purposes, and for use on backlevel systems.The following are displayed only on a system running in workload management **compatibility** mode: • Performance group number • Domain number In goal mode these fields are initialized to "N/A."

Table 4-13. Displaying System Activity: Information for the ALL Operand (continued)

Primary Operand	Information for the ALL Operand
	The following are displayed only on a system running in workload management **goal** mode: • 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: • 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
	Note: Beginning with z/OS V1R3, WLM compatibility mode is no longer available. The following information has been left here for reference purposes, and for use on backlevel systems.The following are displayed only on a system running in workload management **compatibility** mode: • Performance group number • Domain number In goal mode these fields are initialized to “N/A.”
	The following are displayed only on a system running in workload management **goal** mode: • 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, including: • For each active job, started task, APPC/MVS transaction program, and APPC/MVS scheduler initiator: – All of the information listed for JOBS or J • For each time-sharing user: – All of the information listed for TS • For each active system address space: – 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: • 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
	Note: Beginning with z/OS V1R3, WLM compatibility mode is no longer available. The following information has been left here for reference purposes, and for use on backlevel systems.The following are displayed only on a system running in workload management **compatibility** mode: • Performance group number • Domain number In goal mode these fields are initialized to "N/A."
	The following are displayed only on a system running in workload management **goal** mode: • 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 with the specific name: • 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

Table 4-14. Displaying System Activity: Information for a Specific Name (continued)

Primary Operand	Information for the NAME Operand
	Note: Beginning with z/OS V1R3, WLM compatibility mode is no longer available. The following information has been left here for reference purposes, and for use on backlevel systems.The following are displayed only on a system running in workload management **compatibility** mode: • Performance group number • Domain number In goal mode these fields are initialized to “N/A.”
	The following are displayed only on a system running in workload management **goal** mode: • 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: • For an active job, active APPC/MVS transaction program name, initiator address space and started task: – All of the information listed for JOBS or J • For a time-sharing user: – All of the information listed for TS • For an active system address space: – 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:

- 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
D A,*	Not valid (can be done using D A,ALL)

Table 4-16. Examples of DISPLAY Commands (continued)

DISPLAY Command	Displayed Jobs
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 4-501 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.

- **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,JOBNAME=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,JOBNAME=SYMBOLS (*membername* and *JOBNAME* were specified), only the job name specified in the command appears in the output.

```
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.

Syntax

The DISPLAY LLA command has no parameters. The complete syntax is:

```
D LLA
```

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.SEZALINK
     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 xvii) 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

Starting with OS/390 R6, 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. To enable earlier releases of OS/390, including MVS/ESA SP 5.2, to recognize this command, install the PTFs for APAR OW27153 on the earlier releases. The following DISPLAY LOGGER functions are in OS/390 R3 and higher:
- Display of resource manager information
- Display of import connect log streams.

Support for DASDONLY display output for system connection status is only available on OS/390 R4 and higher.

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[,LSName=logstreamname[[,Jobname=mvsjobname][,SUMM  ]] ]    ]
                                                                  [,Detail]
                     |,Jobname=mvsjobname[[,LSNAME|,LSN=logstreamname][,SUMM  ]  ]
                                                                      [,Detail]
                     |,SYSPLEX[,LSName=logstreamname]
                     |,DASDONLY
        [,Logstream[,LSName=logstreamname][,STRNAME|,STRN=structurename]        ]
                                          |,DASDONLY
        [,STRUCTURE|,STR[,STRNAME|STRN=structurename]                          ]
```

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
Due to XCF local mode, the system logger will not be available for the life of this IPL.

INITIALIZING
The system logger is in the process of initializing.

ACTIVE
The system logger is up and running.

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.

There are the following four filters you can specify 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.

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*
```

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*
```

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

(see “Using LookAt to look up message explanations” on page xvii) 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|cc|cca|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,cc,cca,name, or name-a
Indicates the display area (a), console (cc), both the console and the display area (cca), console name (name), or both the console name and the display area (name-a) where the display will be presented. For cc, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect). For more information, see “MSGRT Command” on page 4-319.

Example

To display information for all the logrec medium settings:

```
DISPLAY LOGREC,ALL
```

Displaying System Configuration Information

Use the DISPLAY M command to display the status of sides, processors, vector facilities, ICRFs, channel paths, devices, central storage, and expanded storage or to compare the current hardware configuration to the configuration in a CONFIGxx parmlib member.

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)]
     |={CPUID|CPU}[(x)|(list)]
     |={DEVICE|DEV}[([/]devnum)|([/]lowdevnum-[/]highdevnum)|(list)]
     |={DEVICE|DEV}([/]devnum,(chp))
     |={DEVICE|DEV}(([/]devnum),chp)
     |=ESTOR[(ddddM-ddddM)|(list)|(E[=id])]
     |=HIGH
     |=HSA
     |=SIDE[(id)]
     |={STORAGE|STOR}[(ddddM-ddddM)|(list)|(E[=id])]
     |=SWITCH(sssss [,pp[-pp] [,pp[-pp]]...])
     |=(parm[,parm]...)

   [,L={a|cc|cca|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, vector facilities, ICRFs, central storage, expanded 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 xvii) 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.

(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 xvii) 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 Vector Facilities or ICRFs attached to those processors. For a description of the display format, use LookAt (see "Using LookAt to look up message explanations" on page xvii) 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 Vector Facilities or 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, vectors, and ICRFs defined to the partition.

(x) A single processor identified by processor identifier. The processor identifier can have a value of 0 to F.

(list)
One or more processor identifiers, each separated by a comma.

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 xvii) or use the *MVS System Messages* books.

([/]*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 or 4 hexadecimal digits, optionally preceded by a slash (/). A channel path identifier can have a value from 0 to FF.

ESTOR
The system is to display the number of megabytes of expanded storage assigned and available to the system. The display includes:

- The amount of online expanded storage
- The amount of expanded storage waiting to go offline
- The amount of bad expanded storage
- The amount of expanded storage in offline expanded storage elements
- The amount of expanded storage that belongs to another configuration

If you do not specify any qualifiers, such as (*dddd*M-*dddd*M), (*list*), or (E[=*id*]), the system displays information for all installed expanded storage. Use ESTOR only when your system has expanded storage installed. ESTOR is not applicable to the z/Architecture environment. For a description of the display format, see message IEE174I. Use LookAt (see "Using LookAt to look up message explanations" on page xvii) or use the *MVS System Messages* books.

(*dddd***M-***dddd***M)**
A range of addresses (in decimal) of expanded storage. *dddd*M must be a multiple of 64 megabytes, and cannot exceed 4095. The starting and ending addresses must not be the same.

(list)
One or more address ranges (in decimal) of expanded storage, each separated by a comma.

(E[=*id***])**
The system is to display the amount of storage in the expanded storage element and identify whether it is online or offline. If you omit the *id*, the system displays this information for all installed expanded storage elements. If the processor complex is partitioned and the specified expanded storage element is part of another configuration, no information is provided.

HIGH
The system is to display the highest possible central storage and expanded storage addresses in decimal M bytes (megabytes). 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 xvii) 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 xvii) 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 xvii) 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 (*dddddX-dddddX*), (*list*), or (E[=*id*]), the system displays the status of all central storage. For a description of the display format, see message IEE174I. Use LookAt (see "Using LookAt to look up message explanations" on page xvii) or use the *MVS System Messages* books.

(*dddddX***-***dddddX***)**

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 (*dddddX*) 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 xvii) 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 *parm*s you specify within the parentheses may contain any combination of CHP, CPU, DEV, HIGH, HSA, ESTOR, 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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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 and expanded storage, enter:

```
D M=CPU
D M=CHP(01,03-05)
D M=HIGH
       or
D M=(CPU,CHP(01,03-05),HIGH)
```

Example 6

To display the number of megabytes assigned and available to the system in the expanded storage ranges 64M through 128M, enter:

```
D M=ESTOR(64M-128M)
```

Example 7

To display the number of megabytes assigned and available to the system in the expanded storage element 0 and the status of the element, enter:

```
D M=ESTOR(E=0)
```

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|cc|cca|name|name-a}]
```

MMS
Displays the status of the MVS message service and the list of the languages that are currently available.

L=*a, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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:

DISPLAY MMS Command

```
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
```

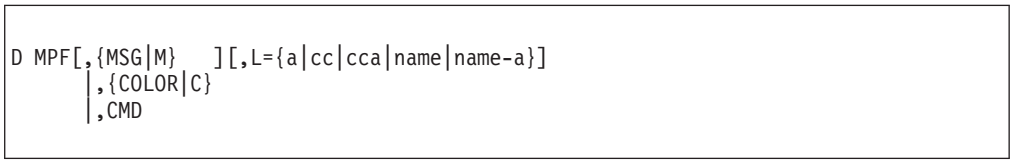
When the MVS message service is not active, the system issues message IEE294I.

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:



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

Use LookAt (see “Using LookAt to look up message explanations” on page xvii) 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 attribute, or command installation exit definition

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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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

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)
- Hierarchical file system (HFS) 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 z/OS UNIX System Services parmlib statements
- 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 Pre-Router routing tables.
- Information about z/OS UNIX System Services parmlib limits, including current system-wide and process limits, their high-water marks, and current usage.
- 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. 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 an 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
      |,U=userid
      |,{PID}=processid[,BRL]
      |,{FILE|F[,CAPS|C]}
      |,{VSERVER|V}
      |,{PFS|P}
      |,{CINET|CI}=All|TPname
      |,{OPTIONS|O}
      |,{LIMITS|L[,PID=ProcessId][,RESET]}

   [,L={a|cc|cca|name|name-a}]
```

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.

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, the D OMVS,PID= command must always be issued from the system on which the specified process is running.

FILE *or* **F**
Displays a list of HFS file systems that z/OS UNIX System Services is currently using and the status of each HFS.

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.

CAPS *or* **C**
Displays variable data containing lowercase letters in uppercase.

CINET = *or* **CI = ALL|tpname**
Displays the Common Inet routing information for all of the active transport providers in use by the Common Inet Pre-Router. 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.

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.

RESET
Resets the high-water mark for a system limit to 0.

BRL
Displays thread-level information for any thread that is in a byte-range lock wait. This operand can only be specified with PID=.

L=*a, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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.

```
BPXO040I 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.

```
BPXO040I 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.

```
BPXO040I 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 (OMVS ACTIVE) appears before the file system information.

```
00  BPXO045I 12.28.28 DISPLAY OMVS 011
    OMVS     000E ACTIVE          OMVS=(66)
    TYPENAME   DEVICE ----------STATUS----------- MODE
    HFS             4 ACTIVE                     READ
      NAME=POSIX.USR.LPP
```

```
      PATH=/usr/lpp
      MOUNT PARM=SYNC(60)
      OWNER=SYSTEM2   AUTOMOVE=Y   CLIENT
      QSYSTEM=system1 QJOBNAME=FRED   QPID=34567
    HFS             3 ACTIVE                      READ
      NAME=POSIX.HFS.NLS
      PATH=/usr/ib/nls
      OWNER=SYSTEM2   AUTOMOVE=Y   CLIENT=Y
    HFS             2 ACTIVE                      READ
      NAME=POSIX.HFS.MAN
      PATH=/usr/man
      OWNER=SYSTEM3   AUTOMOVE=Y   CLIENT=Y
    HFS             1 ACTIVE                      RDWR
      NAME=POSIX.HFS.FS
      PATH=/
      OWNER=          AUTOMOVE=N   CLIENT=N
```

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.

```
BPXO040I 14.38.46 DISPLAY OMVS 030
OMVS     000E ACTIVE          OMVS=(93)
USER     JOBNAME  ASID        PID       PPID STATE   START    CT_SECSS
IBMUSER  BPXOINIT 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 initilization by the parmlib member BPXPRMxx or with the SET command, enter:

```
DISPLAY OMVS,O
```

```
d omvs,o
```

```
BPXO043I 10.26.49 DISPLAY OMVS 007
OMVS     000E ACTIVE          OMVS=(69)
z/OS UNIX CURRENT CONFIGURATION SETTINGS:
MAXPROCSYS      =        256    MAXPROCUSER     =         16
MAXFILEPROC     =        256    MAXFILESIZE     = NOLIMIT
MAXCPUTIME      =       1000    MAXUIDS         =        200
MAXPTYS         =        256
MAXMMAPAREA     =        256    MAXASSIZE       =   41943040
MAXTHREADS      =        200    MAXTHREADTASKS  =         50
MAXCORESIZE     =    4194304    MAXSHAREPAGES   =       4096
IPCMSGQBYTES    =     262144    IPCMSGQMNUM     =      10000
IPCMSGNIDS      =        500    IPCSEMNIDS      =        500
IPCSEMNOPS      =         25    IPCSEMNSEMS     =         25
IPCSHMMPAGES    =        256    IPCSHMNIDS      =        500
IPCSHMNSEGS     =         10    IPCSHMSPAGES    =     262144
SUPERUSER       = BPXROOT       FORKCOPY        = COW
STEPLIBLIST     =
USERIDALIASTABLE=
PRIORITYPG VALUES: NONE
PRIORITYGOAL VALUES: NONE
MAXQUEUEDSIGS   =       1000    SHRLIBRGNSIZE   =   67108864
```

```
SHRLIBMAXPAGES  =       4096    VERSION         = /
SSYSCALL COUNTS = NO            TTYGROUP        = TTY
SYSPLEX         = NO            BRLM SERVER     = N/A
LIMMSG          = NONE
```

Note: The SYSPLEX (YES) option indicates the system is in a sysplex and is using the shared HFS capability. You cannot dynamically change the SYSPLEX parameter through SETOMVS or SET OMVS. For more information, see the chapter on Shared HFS in *z/OS UNIX System Services Planning*.

To display the current setting of the options that were set during initilization 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,O
```

```
BPXO043I 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
IPCMSGQBYTES    =     262144    IPCMSGQMNUM     =      10000
IPCMSGNIDS      =        500    IPCSEMNIDS      =        500
IPCSEMNOPS      =         25    IPCSEMNSEMS     =         25
IPCSHMMPAGES    =        256    IPCSHMNIDS      =        500
IPCSHMNSEGS     =         10    IPCSHMSPAGES    =     262144
SUPERUSER       = BPXROOT       FORKCOPY        = COW
STEPLIBLIST     =
USERIDALIASTABLE=
PRIORITYPG VALUES: NONE
PRIORITYGOAL VALUES: NONE
MAXQUEUEDSIGS   =       1000
SYSCALL COUNTS  = NO            TTYGROUP        = TTY
```

Example 7

To display the thread information for process id 1, enter:

```
DISPLAY OMVS,PID=1
```

```
BPXO040I 11.13.40 DISPLAY OMVS 971
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
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
```

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
```

```
BPXO046I 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
  BPXFCSIN    CHAR SPECIAL          BPXFCSIN
  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 is 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 SUBFILESYSTPE statements is displayed. For the HFS, 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:

```
DISPLAY OMVS,CINET=ALLBPXO0nnI 17:12:37 DISPLAY OMVS nn
OMVS     000E ACTIVE          OMVS=(ZD)
HOME INTERFACE INFORMATION
 TP NAME    HOME ADDRESS      FLAGS
  TCP41      127.116.117.233   DRS
  TCP42      127.116.118.234
  TCP43      127.116.119.235

HOST ROUTE INFORMATION
 TP NAME    HOST DESTINATION
  TCP41      127.117.193.234
  TCP41      127.117.194.234
  TCP42      127.117.195.234

NETWORK ROUTE INFORMATION
 TP NAME    NET DESTINATION   NET MASK          METRIC
  TCP41      127.111.000.000   255.255.000.000    10
  TCP42      127.113.000.000   255.255.000.000     0
  TCP41      197.119.119.000   255.255.255.000     F
  TCP43      009.000.000.000   255.000.000.000     F
```

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.

HOST DESTINATION
When a transport provider is connected to a host, the host IP address is displayed.

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.

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 Pre-Router selects the route with the best metric. The higher the number, the better the metric. The metric 255 = 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
```

```
 BPXO051I 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
 IPCMSGNIDS               0          0        500
 IPCSEMNIDS               0          0        500
 IPCSHMNIDS               0          0        500
 IPCSHMSPAGES             0          0     262144 *
 IPCMSGQBYTES           ---          0     262144
 IPCMSGQMNUM            ---          0      10000
 IPCSHMMPAGES           ---          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, IPCSMSGQMNUM, and IPCSHMMPAGES 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
 BPXO051I 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
MAXCORESIZE            ---        ---    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.

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 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  HIGHWATER    PROCESS
                     USAGE      USAGE      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  HIGHWATER    PROCESS
                    USAGE      USAGE      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
```

```
BPXO040I 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
 250640E000000002 009C8550 OMVS                    .124 RED  C
 BRLWAIT DEV=00000001 INO=0000002E FILE=/u/john/filenam+ PID=12345678
```

The information displayed is:

FILE
Up to 16 characters of the filename of the file that is being locked. It 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.

INO
The inode number of the file, as shown by **ls -i**.

DEV
The device number of the file's mounted file system.

Displaying Sysplex-Wide Operator Information

Use the DISPLAY OPDATA command to display sysplex-wide operator information (OPDATA). The command prefixes defined for subsystems in the sysplex are displayed. See *z/OS MVS Planning: Operations* for more information.

The syntax of the DISPLAY OPDATA command is:

```
D {OPDATA|O}[,PREFIX][,L={a|cc|cca|name|name-a}]
```

OPDATA *or* **O**
The system is to display (message IEE603I) sysplex-wide information about message processing and presentation.

PREFIX
The command prefixes defined for subsystems in the sysplex are displayed.

L=*a, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

Example

To display information about the command prefixes current and active on the system, enter:

```
D O,PREFIX
```

Displaying PARMLIB Information

Use the DISPLAY PARMLIB command to display:

- The parmlib data sets and volume serial numbers that are defined in LOAD*xx*.
- The parmlib data sets and volume serial numbers that are defined in the MASTER JCL (when there are no LOAD*xx* 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|cc|cca|name|name-a}]
```

ERRORS or E
Parmlib data sets and volume serial numbers that were defined in LOAD*xx* PARMLIB statements but were not found.

L=*a, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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 LOAD*xx*.

```
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 LOAD*xx*.

```
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 LOAD*xx* but they were not found. Issue the D PARMLIB,ERRORS command.
  - No parmlib data sets were specified in either the LOAD*xx* 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:

```
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=cc                ][,L={a|cc|cca|name|name-a}]
     |,{TABLE|T}[=nnnnnnnn]
```

PFK

The system displays information about the PFKs (message IEE235I). Unless you specify otherwise on the CN=nn 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=*cc*

Requests the PFK definitions for the console with ID *cc*.

L=*a, cc, cca, name, or name-a*

Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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

DISPLAY PFK Command

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 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]
```

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.

REGISTERED|REG
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.

Example 1

If you enter the following command:

```
D PROD,REG
```

the response has the following format:

```
IFA111I 11.11.51 PROD DISPLAY 097
S   OWNER          NAME             FEATURE          VERSION  ID
E   IBM CORP       OS/390           OS/390           01.01.00 5645-001
```

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|cc|cca|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.

L=*a, cc, cca, name, or name-a*
Specifies where the system is to present the display:
- The console identifier (*cc*)
- The display area (*a*)
- Both the console identifier and the display area (*cca*)
- The console name (*name*)
- Both the console name and the display area (*name-a*).

For *cc*, you must specify a decimal number from 1 to 99.

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|cc|cca|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 4-190.

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, cc, cca, name, or name-a*
Specifies where the system is to present the display:

- The display area (*a*).

- The console identifier (*cc*). For *cc*, you must specify a decimal number from 1 to 99.
- Both the console identifier and the display area (*cca*).
- The console name (*name*).
- Both the console name and the display area (*name-a*).

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:

```
CSV460I 17.01.16 PROG,EXIT DISPLAY 710
EXIT             DEF EXIT             DEF EXIT             DEF
IEF_ALLC_OFFLN    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                           ]
```

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 LNKLIST 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 ?).

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 xvii) or use the *MVS System Messages* books to see message CSV470I.

```
CSV470I  01.00.00
LNKLST DISPLAY
    LNKLST SET LNKLST1 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 LNKLST set associated with the job that matches the jobname MYJOB, enter:

```
D PROG,LNKLST,JOBNAME=MYJOB
```

The output appears in the following format:

```
CSV473I  02.15.00 LNKLST 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|cc|cca|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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both display area and console (*cca*), console name (*name*), or both console name and display area (*name-a*) where the system is to present the display. For *cc*, you must specify a decimal number from 1 to 9.

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:

- 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 xvii) 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={xx|name|(ALL)}][,ROUT={ALL|(rrr[,sss]...)    }
   |,CE                                        {(rrr-sss[,rrr-sss]...)}
   |,R
   |,M
   |,{LIST|L}
   |,{ALL|A}
   |,
                                          (See Note)
     [,L={a|cc|cca|name|name-a}]
```

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 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 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.

xx Requests those outstanding action messages that the system directed to the console with the two-digit ID *xx*.

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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the system is to present the display. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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 console 2, enter:

```
D R,L,L=2A   or   D R,A,L=2A
```

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 the master console:

```
DISPLAY R,A,S
```

Example 8

To display all outstanding messages issued on system SY2:

```
DISPLAY R,M,SYS=SY2
```

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 master console (with an ID of 02), excluding messages that were directed to the master console by default, enter:

```
DISPLAY R,L,CN=02
```

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 04 by any routing code, enter:

```
DISPLAY R,ROUT=ALL    issued from console 04
or
DISPLAY R,ROUT=ALL,CN=04   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 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 JOBNAME 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|cc|cca|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=lname
Displays the common storage use and definitions, as well as the names of the physical and logical libraries that match the specified name. For *lname*, 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.

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, cc, cca, name, or name-a
Specifies where the system is to present the display:

- The display area (*a*).
- The console identifier (*cc*).
- Both the console identifier and the display area (*cca*).
- The console name (*name*).
- Both the console name and the display name (*name-a*).

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

DISPLAY RTLS Command

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
```

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|cc|cca|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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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|,O][,L={a|cc|cca|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, cc, cca, name,* **or** *name-a*

Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

Displaying Storage Management Subsystem Information

Use the DISPLAY SMS command to display information about the:

- 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

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'
- *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, 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*.

```
D SMS[,{ACTIVE|A}                                    ]

     [,CACHE                                         ]

     [,CFCACHE(structurename|*)                      ]

     [,CFLS                                          ]

     [,CFVOL(volid)                                  ]

     [,{DRIVE|DRI}(name|ALL)[,STATUS ]               ]
     [                        |,DETAIL               ]

     [,{LIBRARY|LIB}(name|ALL)[,STATUS[,LISTDRI]] ]
     [                          |,LISTDRI          ]
     [                          |,DETAIL           ]
     [,MONDS(specmask|*)                             ]

     [,OAM                                           ]

     [,OPTIONS                                       ]

     [,OSMC[,TASK(name)]                             ]

     [,SHCDS                                         ]

     [,SMSVSAM[,ALL]                                 ]

     [,SMSVSAM,QUIESCE                               ]

     [,{STORGRP|SG}{(storgrp|ALL)}[,LISTVOL ]     ]
     [                             |,DETAIL       ]
     [,{TRACE|T}                                     ]

     [,{VOLUME|VOL}(volume)                          ]

   [,L={a|cc|cca|name|name-a}]
```

SMS

Displays information about the Storage Management Subsystem (SMS). 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) that SMS waits between reading device statistics for the 3990-3 control unit. DINTERVAL is shown in seconds. Also displayed are the values that were specified for the REVERIFY and ACSDEFAULTS parameters in the IGDSMSxx parmlib member.

This 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. This interval (not to be confused with DINTERVAL) is the number of seconds delay that an SMS subsystem is to allow before synchronizing with the other SMS subsystems in the complex. A longer interval setting allows 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 DFSMSdfp 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.

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 a drive named ALL, place the name in double parentheses as DRIVE((ALL)).

Note: To obtain the online or offline status of devices within a tape library, use the DISPLAY UNIT, DEVSERV, or LIBRARY DISPDRV command.

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.

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; specify LIBRARY((ALL)).

LISTDRI
Displays the offline/online status for all the optical drives associated with the specified libraries.

Note: When you specify the LISTDRI keyword, you cannot specify the DETAIL keyword.

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 '*' 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 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 intermal for recording data, and all other parameter specifics.

For a description of the output, use LookAt (see “Using LookAt to look up message explanations” on page xvii) or use the *MVS System Messages* books to see message IGD002I.

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.

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)).

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.)

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:
- 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 there are no storage groups defined to the active configuration, this message is displayed:

```
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 trace table size. 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 xvii) or use the *MVS System Messages* books to see message IGD002I.

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 indications 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.

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)).

L={a *or* **cc** *or* **cca** *or* **name** *or* **name-a}**

Specifies the display area (L=a), console (L=cc), both (L=cca), console name (L=name), or both (L=name-a) where the display will be presented. For cc, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

Example 1

To display the active SMS configuration, you would enter:

```
d sms or 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 10:45:30 DISPLAY SMS 444
SCDS = TEST120.TDCS11J
ACDS = WORK.ACDS
COMMDS = WORK.COMMDS
DINTERVAL = 150
REVERIFY = NO
ACSDEFAULTS = YES
    SYSTEM          CONFIGURATION LEVEL       INTERVAL SECONDS
    SYSTEM1         1995/08/23    10:45:20   15
IEF196O  IEF237I 0801 ALLOCATED TO SYS00052
```

Example 2

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 3

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 STORGRPNAME
***************************** 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 4

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
```

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.

```
15.55.32 SYSTEM1           IGD002I 15:55:32 DISPLAY SMS 452
ACDS     = SYS1.STAGE2.ACDS
COMMDS   = SYS1.STAGE2.COMMDS
INTERVAL = 15
   ⋮
COMPRESS = GENERIC
TRACE    = ON     SIZE = 128K     TYPE = ERROR
  JOBNAME = *    ASID = *
  TRACING EVENTS:
    MODULE = OFF  SMSSJF = OFF  SMSSSI = OFF  ACSINT = ON
    OPCMD  = OFF  CONFC  = ON   CDSC   = OFF  CONFS  = OFF
    MSG    = ON   ERR    = OFF  CONFR  = OFF  CONFA  = OFF
    ACSPRO = OFF  IDAX   = OFF  DISP   = OFF  CATG   = OFF
    VOLREF = OFF  SCHEDP = OFF  SCHEDS = OFF  VTOCL  = OFF
    VTOCD  = OFF  VTOCR  = OFF  VTOCC  = OFF  VTOCA  = OFF
    RCD    = OFF  DCF    = OFF  DPN    = OFF  TVR    = OFF
    DSTACK = OFF
```

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|cc|cca|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 xvii) 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.

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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display is presented. For *cc* the operator must specify a decimal number from 1 to 99.

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 entered the command (unless routing instructions are in effect; see the MSGRT 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 1000 lines, the system may issue a X'D23' abend with error code 33020000. This abend indicates a WQE buffer shortage. 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|cc|cca|name|name-a}]
```

SYMBOLS
The system is to display, in message IEA007I, the static system symbols defined to this system.

L={a *or* **cc** *or* **cca** *or* **name** *or* **name-a}**
Specifies the display area (L=a), console (L=cc), both (L=cca), console name

(L=name), or both the console name and display area (L=name=a) where the display is to be presented. For cc, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

Example

To display the static system symbols defined to this system, enter:

```
D SYMBOLS
```

Displaying the Local and Greenwich Mean Time and Date

Use the DISPLAY T command to display the local time of day and date and the Greenwich mean time (GMT) of day and date.

```
D T
```

T The local time of day and date and the Greenwich mean time (GMT) 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|cc|cca|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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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*
     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
- 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|cc|cca|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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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.
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 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 workload management mode in effect for a specific system, or for all systems in a sysplex.
- 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.

```
D WLM[,SYSTEM=sysname|,SYSTEMS]
     [,APPLENV=applenvname|*]
     [,SCHENV=schenvname[,SYSTEM=sysname|,SYSTEMS]]
     [,RESOURCE=resourcename[,SYSTEM=sysname|,SYSTEMS]]
     [,L={a|cc|cca|name|name-a}]
```

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 and the workload management mode 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*), or if you specify APPLENV=*, the command displays status information for all application environments.

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=*resourcename*
Displays status information for the specified resource (*resourcename*). 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.

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: COEFFS
  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 SYSZWLM_WORKUNIT STATUS: CONNECTED
  STRUCTURE SYSZWLM_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: COEFFS
  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 SYSZWLM_WORKUNIT STATUS: CONNECTED
  STRUCTURE SYSZWLM_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: COEFFS
  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 SYSZWLM_WORKUNIT STATUS: CONNECTED
  STRUCTURE SYSZWLM_53052064 STATUS: CONNECTED

  *SYSNAME*  *MODE*  *POLICY*  *WORKLOAD MANAGEMENT STATUS*
  WLTEAM9    GOAL    WEEKDAY   ACTIVE
  SYS2       COMPAT  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

  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
```

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:

```
D XCF[,{PATHIN|PI}                                                          ]
          [,{DEVICE|DEV}={([/]indevnum[,[/]indevnum...])|ALL}]
             [,{STRNAME|STRNM}={(strname[,strname]...)|ALL}]
          [,{SYSNAME|SYSNM}=(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|SYSNM}=(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}]
             [,{CONNAME|CONNM}={(conname[,conname]...)|ALL}]
             [,{STATUS|STAT}=([ALLOCATED][,NOTALLOCATED]
                        [,POLICYCHANGE][,DEALLOCPENDING]
                        [,LARGERCFRMDS][,REBUILD][,STRDUMP]
                        [,ALTER][,FPCONN][,NOCONN])]

     [,{CF}[,{CFNAME|CFNM}={(cfname[,cfname]...)|ALL}]                  ]

     [,{POLICY|POL}[,TYPE={(name[,name]...)|ALL}]                       ]

     [,{PRSMPOLICY|PRSMPOL}                                             ]

     [,{ARMSTATUS|ARMS}                                                 ]
          [,{RESTARTGRP|RG}=rgname]
          [,{ELEMENT|EL}=elname|{JOBNAME|JOB}=jobname]
          [,INITSYS=initsys]
          [,CURRSYS=currsys]
          [,STATE=([{STARTING|START}][,{AVAILABLE|AVAIL}][,FAILED]
                   [,{RESTARTING|RESTART}][,{RECOVERING|RECOVER}])]
          [,DETAIL]

     [,L={a|cc|cca|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 IXL355I. If there are no inbound paths to this system, the system displays message IXC338I.

STRNAME= *or* **STRNM=(***strname[,strname...]***)** *or ALL*
Requests that the system display (message IXC360I) 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.

LINKING *or* **LINK**
Establishing communication.

WORKING *or* **WORK**
Capable of being used.

QUIESCING
Quiescing the use of

QUIESED
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 IXL356I. If there are no outbound paths to this system, the system displays message IXC338I.

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 4-227.

SYSNAME= *or* **SYSNM=***sysname***(***s***)**
"See the SYSNAME attribute of the PATHIN keyword" on page 4-227.

STATUS= *or* **STAT=***state***(***s***)**
"See the STATUS attribute of the PATHIN keyword" on page 4-227.

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.

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.

CONNAME= *or* **CONNM=***conname***(***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

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
- LOGR for system logger

PRSMPOLICY *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.

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, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 (unless routing instructions are in effect; see the MSGRT command later in this chapter).

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"}
```

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'*. For example:

```
DUMP PARMLIB=xx,SYMDEF=(&PAGING1.='AQFT',&CICS.='CICS1')
```

Note: 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.

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]...) ]
```

```
Where s-option represents:

STRNAME=strname
  [,CONNAME=conname                                                    ]

  [,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. You can specify a maximum of 15 address space identifiers.

If your REPLY specifies more than 15 address spaces total, the system issues message ASA104I. The system processes the dump request using the first 15 address spaces it determined. The address spaces are specified as follows:

- The address spaces directly requested in the ASID parameter
- The address spaces associated with the jobs named in the JOBNAME parameter
- The address spaces associated with the user IDs in the TSONAME parameter
- The address spaces for the data spaces in the DSPNAME parameter

Wildcards used in the JOBNAME, TSONAME, and DSPNAME parameters can result in multiple address spaces.

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 limit on address spaces that can be specified.

You can specify each name explicitly or with wildcards. See “Wildcards” on page 4-235.

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 limit on address spaces that can be specified.

You can specify the name explicitly or with wildcards. See “Wildcards” on page 4-235.

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 TCAM/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 data space you want to dump.

'*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 4-235.

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 4-235.

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.

Note: When a data space is owned by an address space not included in the dump, that address space's ASID is added to the dump. Because SDUMP supports a maximum of 15 address spaces, the address space limitation may prevent specifying as many as 256 data spaces. See the ASID parameter for the limit on the number of address spaces you can specify.

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 4-235.

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.

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 4-235.

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=(*dspname-entry***[,***dspname-entry***]...)**
Specifies the data spaces to be dumped on the other systems. See the "DSPNAME parameter" on page 4-238 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" on page 4-242 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" 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.
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, SQ, SUM, and IO.

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.

CONNAME= *or* **CONNM=***conname*
When specified for a coupling facility cache structure, requests the user registry information for this user be included in the dump. *conname* is the name of a connected user. If the connected user represented by the conname does not exist, the dump will not contain user registry information.

ACCESSTIME= *or* **ACC={ENFORCE** *or* **ENF** *or* **NOLIMIT** *or* **NOLIM}**
Indicates whether the dump time limit specified on the ACCESSTIME parameter of the IXLCONN macro is in effect. When ACCESSTIME=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 ACCESSTIME=0 is specified on the IXLCONN macro and ACCESSTIME=ENFORCE is specified on the dump request, the structure will not be included in the dump.

When ACCESSTIME=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.

EVENTQS *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.

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 ACCESSTIME=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 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 ACCESSTIME are specified more than one time for a structure, the first CONNAME and the last ACCESSTIME 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 or 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
⋮	⋮

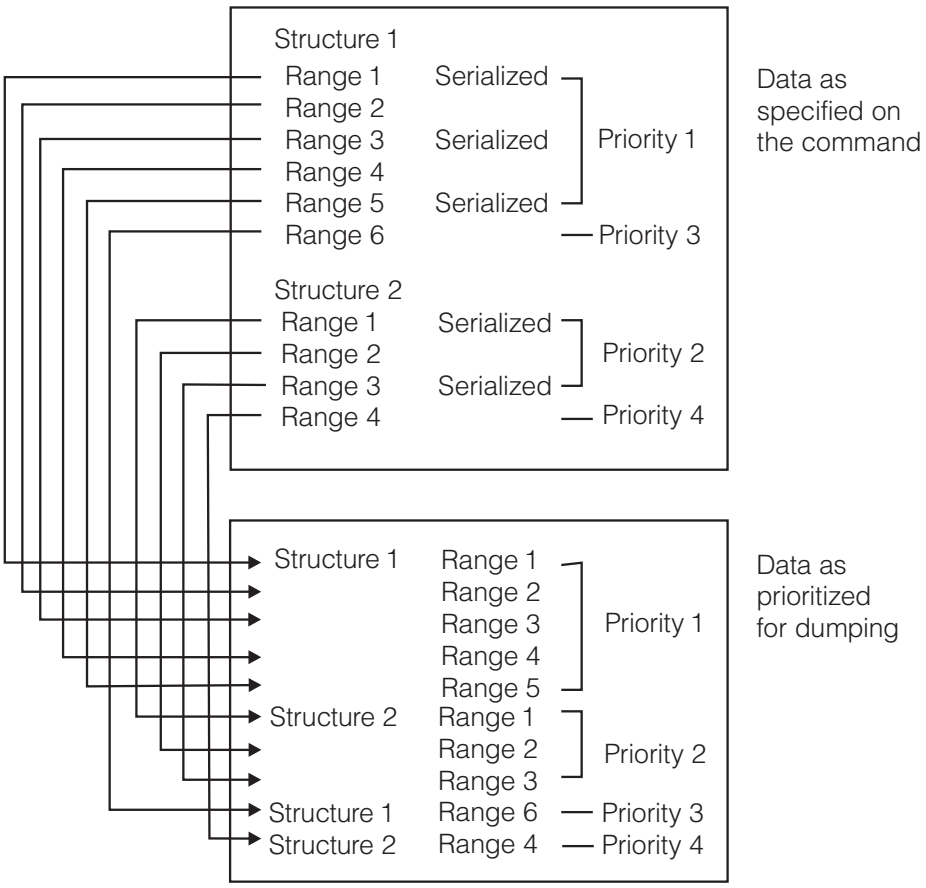
If the system cannot dump all the data you requested, it prioritizes the data according to your specifications on the command in the following manner:

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 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=(S1)),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))
```

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,CONNAME=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. JOBNAME=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,JOBNAME=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,JOBNAME=IRLM*,REMOTE=(SYSLIST=*('IRLM*'))
```

Example 12

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,JOBNAME=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 DUMPS)
* 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 4-17. Summary of the DUMPDS Command

Command:	Topic:
DUMPDS ADD	"Adding System Dump Resources" on page 4-255
DUMPDS ALLOC	"Enabling and Disabling Automatic Dump Data Set Allocation" on page 4-258
DUMPDS CLEAR	"Making Dump Data Sets Ready To Receive Dumps" on page 4-259
DUMPDS DEL	"Deleting System Dump Resources" on page 4-259
DUMPDS NAME	"Setting the Name-Pattern for Dump Data Sets" on page 4-261

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 4-261 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.

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

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 inactivate 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.

```
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 inactivate 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.

```
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&HHMMDD..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
```

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 4-18. FORCE Command Tasks

Task - Immediately Terminate:	Syntax:
• A job in execution • A running Advanced Program-to-Program Communication/MVS (APPC/MVS) transaction program • A started task	FORCE jobname
• A time-sharing user (U=*userid*)	FORCE U=userid
• A started task • A MOUNT command • An external writer allocation • The output processing for a job • An z/OS UNIX process	FORCE identifier

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.
- 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              }[,ARM][,A=asid][,ARMRESTART]
      {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:

1. 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.
2. 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.
3. 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.
- *[/]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

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. See *z/OS MVS Recovery and Reconfiguration Guide* for more information.

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

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.

Note: If you need more information about RACF groups, see *z/OS Security Server RACF General User's Guide*.

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.

If you need more information about security labels, see *z/OS Security Server RACF General User's Guide*.

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

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 4-19 summarizes the information that the MODE command provides.

Table 4-19. 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 MODE VS	"Controlling the Recording of Hard Machine Check Interruptions" on page 4-278
MODE DG MODE SR	"Controlling the Recording of System Recovery and Degradation Machine Check Interruptions" on page 4-281
MODE STATUS	"Displaying Recording and Monitoring Status" on page 4-282

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

Table 4-20 shows the machine check interruption types you can specify and the parameters allowed for each type:

Table 4-20. 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)
VS — Vector Facility source	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:

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}           {300  }          |=ALL       {ALL}
     {IV}                            |=25
     {TC}                            |=16
     {PT}                            |=5
     {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.

VS
Machine checks indicating Vector Facility source 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 or Vector Facility, 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 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 or vector facility 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
```

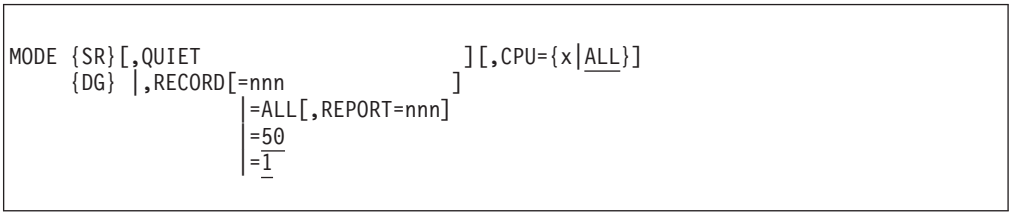
Example 4

Monitor the frequency of Vector Facility machine-check interruptions on processor 1. If one of these interruptions occurs in 300 seconds, disconnect the Vector Facility and issue message IGF970E.

```
mode vs,record=1,cpu=1
```

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.



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 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 does not recognize the input specified on 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 4-21 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 4-21. Summary of the MODIFY Command

Topic:	Command:
“Passing Information to a Job or Started Task” on page 4-285	F jobname.identifier,parameters
“Passing Information To An z/OS UNIX System Services Application” on page 4-287	F jobname.identifier,APPL=text
“Starting TSO/TCAM Time Sharing” on page 4-288	F jobname.identifier,TS=START,member
“Stopping TSO/TCAM Time Sharing” on page 4-288	F jobname.identifier,TS=STOP
“Modifying TSO/VTAM Time Sharing” on page 4-289	F jobname.identifier,USERMAX=,USER=
“Controlling z/OS UNIX System Services (z/OS UNIX)” on page 4-290	F BPXOINIT,parameters
“Communicating with the Catalog Address Space” on page 4-295	F CATALOG,parameters
“Changing the DLF Processing Mode” on page 4-301	F DLF,MODE=
“Changing the DLF parameters” on page 4-302	F DLF,NN=
“Displaying DLF Status” on page 4-302	F DLF,STATUS
“Building and Replacing Library Lookaside Directories” on page 4-303	F LLA,parameters
“Operating with the Network File System Server” on page 4-303	F MVSNFS,parameters
“Collecting Problem Information for the Network File System Server” on page 4-305	F MVSNFS,LOG=
“Managing the Object Access Method (OAM)” on page 4-307	F OAM,parameters
“Recycling z/OS UNIX System Services (z/OS UNIX)” on page 4-307	F OMVS,parameters
“Switching Workload Management Modes and Changing Resource States” on page 4-308	F WLM,MODE
“Specifying Data Set Selection Criteria for an External Writer” on page 4-310	F XWTR.identifier,devnum,parameters

Table 4-21. Summary of the MODIFY Command (continued)

Topic:	Command:
"Causing an External Writer to Pause" on page 4-312	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 4-22 shows examples of START commands that are used to start jobs. Columns three and four show the associated jobnames and identifiers.

Table 4-22. 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

Table 4-22. Examples of START Commands to Start Jobs (continued)

Job Number	START Command	Jobname	Identifier
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 4-23 shows examples of MODIFY commands. The numbers in the second column indicate to which jobs in Table 4-22 on page 4-284 each MODIFY command applies.

Table 4-23. Examples of MODIFY Commands

MODIFY Command	Affected Jobs
F YZ,*parameters*	1, 2, 7
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

Passing Information to a Job or Started Task

Use the MODIFY command to pass information to one or more jobs or started tasks. You can communicate with a currently running program only if it is designed to recognize input from the MODIFY command. If a program does not recognize the input specified on the MODIFY command, the system issues message IEE342I.

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*.

Syntax

```
F [jobname.]identifier,parameters
```

Parameters

The parameters for this command are:

jobname
One of the following:

- The name of a job
- The job name that the system assigns to a started task, which 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 4-284 for more information.

identifier
The identifier assigned to the job or started task. (Refer to “Displaying Started Task Status” on page 4-156 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 4-284 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 An 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 4-284 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 4-156 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 4-284 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

To modify the jobname DATASRVR, enter:

```
F DATASRVR,APPL=threadlimit=5
```

Jobnames can be a maximum of eight characters. No spaces are allowed.

Starting TSO/TCAM Time Sharing

You can use the MODIFY command to start TSO/TCAM time sharing once TCAM is active.

```
F [jobname.]identifier,TS=START[,member]
```

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 4-284 for more information.

identifier
The identifier used on the START command to identify TCAM.

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 4-156 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 4-284 for more information.

TS=START
Initiate TSO/TCAM time sharing.

member
The parmlib member containing TSO/TCAM time-sharing system parameters.

Example

To change the TCAM procedure TCAM2 (used to start TSO/TCAM time-sharing) using the time-sharing system parameters in parmlib member IKJPRM02, enter:

```
S tcam2
F tcam2,TS=start,ikjprm02
```

Stopping TSO/TCAM Time Sharing

You can use the MODIFY command to stop TSO/TCAM time-sharing.

```
F [jobname.]identifier,TS=STOP
```

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 4-284 for more information.

identifier
The identifier used on the START command to identify TCAM.

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 4-156 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 4-284 for more information.

TS=STOP
Halt TSO/TCAM time-sharing. If you enter TS=STOP a second time before time-sharing activity has completely ended, you are asked to respond to the following message:

```
* id IKJ024D TIME SHARING STOP IN PROGRESS,
     REPLY 'U' OR 'FSTOP'
```

Reply ‘U’ to allow the system to wait for time-sharing to stop normally and for users to receive all messages queued for them. If the system is unable to stop normally, reply ‘FSTOP’ to force time-sharing to stop immediately.

Example

To modify the TCAM procedure TCAM2 in order to stop TSO/TCAM time-sharing, enter:

```
F tcam2,TS=STOP
```

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 4-284 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 4-156 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 4-284 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}
           {SHUTDOWN={FILESYS | FORKINIT | FORKS}}
           {TERM=pid[.tid]}
```

The parameters are:

BPXOINIT
The name of the job.

APPL=*appl_data*
Allows information to pass straight through to the application. appl_data is a string that is passed back to the invoker in whatever format the application expects it.

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-HFS 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-HFS 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-HFS software service level. To determine which system is executing with the highest Shared-HFS 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:

DISPLAY *or* **D**

Display the type BPXMCDS couple data set information relating to the Shared-HFS file system. **D** is an alias of **DISPLAY**.

Specify one of the following display options:

ALL

Displays all file systems in the Shared-HFS 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 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.

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-HFS 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.

Always issue the FIX function before performing the UNMOUNTALL function.

OVERRIDE
Normally only one MODIFY command for a FILESYS= function canbe 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.

SHUTDOWN=FILESYS
Unmounts the UNIX System Services filesystems.

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.

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
```

```
BPXO040I 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     WELLIE1        .002 SLP JSN
 049625B000000002 009D8798 OMVS     WELLIE1        .003 SLP JSN
 04962E5800000003 009D5090 OMVS     WELLIE1        .012 SLP JSN
 0496370000000004 009D5228 OMVS     WELLIE1        .011 SLP JSN
 04963FA800000005 009D5A88 OMVS     WELLIE1        .010 SLP JSN
 0496485000000006 009D8048 OMVS     WELLIE1        .011 SLP JSN
 049650F800000007 009D81E0 OMVS     WELLIE1        .011 SLP JSN
 049659A000000008 009D8378 OMVS     WELLIE1        .011 SLP JSN
 0496624800000009 009D8510 OMVS     WELLIE1        .011 SLP JSN
 04966AF00000000A 009D8930 OMVS     WELLIE1        .030 SLP JSN

 f bpxoinit,force=117440514.04962E5800000003
 BPXM027I COMMAND ACCEPTED.

f bpxoinit,term=117440514.0496624800000009
BPXM027I COMMAND ACCEPTED.
```

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  BPXF041I 2000/05/12 11.19.18 MODIFY BPXOINIT,FILESYS=DISPLAY,GLOBAL
SYSTEM   LFS VERSION ---STATUS---------------------- RECOMMENDED ACTION
SY1        1.  1.  0 VERIFIED                       NONE
SY2        1.  1.  0 VERIFIED                       NONE
SY3        1.  1.  0 VERIFIED                       NONE
CDS VERSION=  1        MIN LFS VERSION=  1.  1.  0
BRLM SERVER=SY3        DEVICE NUMBER OF LAST MOUNT=      11
SY1  BPXF040I MODIFY BPXOINIT,FILESYS PROCESSING IS COMPLETE.
```

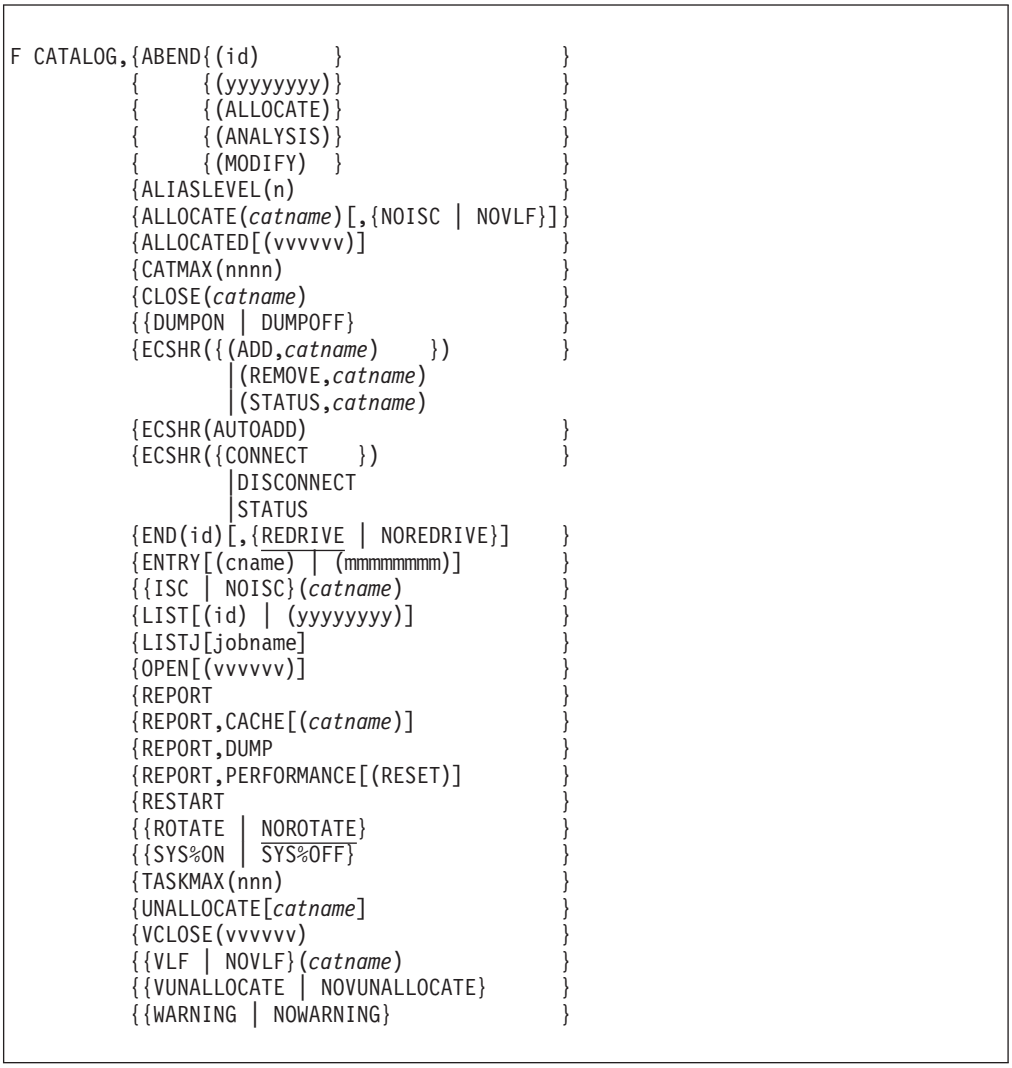
- F BPXOINIT,FILESYS=DISPLAY,FILESYSTEM=POSIX.SY4.HFS

```
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.
```

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. There is one exception to this: for a MODIFY CATALOG,RESTART command, both the master console and the console issuing the command receive messages.



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***)**
The task ID of the catalog address space service task.

(*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[(*vvvvvv***)]**
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.

vvvvvv specifies the volser that can be used to limit the list to allocated catalogs on a specific volume. Use *vvvvvv* to specify the volume serial number.

Note: This form is synonymous with the OPEN(vvvvvv) request, but should be used instead of that form.

CATMAX(*nnnn*)
Closes all open integrated catalog facility user catalogs, and sets a maximum number of such catalogs that can be open concurrently in the catalog address space. This limit is checked as integrated catalog facility catalogs are accessed.

nnn is the limit specified as a decimal value from 0000 to 9999. A value of 0000 indicates no limit, and the user catalogs are allocated in the user's address space.

When the limit is reached, the least recently accessed integrated catalog facility catalog is closed to free its storage and maintain the specified limit. This action has no effect on catalog allocations; the catalogs remain allocated but in restart status with no storage in the catalog address space.

CLOSE(*catname*)
Closes an integrated catalog facility catalog dynamically.

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.

DUMPON
DUMPOFF
Activates or deactivates dynamic dumping by the catalog address space. The default, DUMPOFF, is set at IPL.

ECSHR
Activates or deactivates enhanced catalog sharing or displays status for the entire system or for an individual catalog.

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.

STATUS
Displays (in message IEC380I) the status of the CF connection and the status of each catalog referenced since the last IPL.

ADD,*catname*
Adds the named catalog to the ECS structure in the CF if the catalog has the ECSHARING attribute, shareoptions (3 4), and is on a shared volume, and if the system is connected to the ECS structure. The sharing method for this catalog switches immediately to ECS. If any of the above conditions is not true, the system rejects the request. The issuer of this command should ensure that all current or potential sharers of the catalog are able to use ECS; otherwise, manual intervention may be required to return to the non-ECS sharing method.

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,*catname*
Displays (in message IEC380I) the status of the coupling facility connection and the ECS status of the named catalog.

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.

END(*id***)**
[,REDRIVE]
[,NOREDRIVE]
This is the option IBM recommends to terminate a CAS service task. The task is abended with ABEND91A (REDRIVE option) or ABEND71A (NOREDRIVE option).

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.

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.

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.

NOWARNING
Tells the system not to issue message IDC1997I or IDC1998I before performing an IDCAMS DELETE usercatalog FORCE, or DELETE VVDS RECOVERY. The messages require an operator response to allow the commands to complete.

If a MODIFY CATALOG,NOWARNING is performed, these IDCAMS commands will be executed without intervention. For information about the forms of IDCAMS DELETE, see *z/OS DFSMS Access Method Services for Catalogs*.

OPEN[(*vvvvvv***)]**
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.

vvvvvv is the volume serial number that you specify to limit the list to allocated catalogs on a specific volume.

This option is obsolete and should no longer be used. Use the ALLOCATED option instead.

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***)]**
Causes general information on catalog cache status to be listed. The report generated shows information useful in evaluating the catalog cache performance for the specified catalogs. If you do not specify a catalog name (*catname*), the report will show performance information for all catalogs currently active in the catalog address space.

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.

RESTART
Terminates the catalog address space with an abend code of 81A and requests an SVC dump. 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.

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 SYSCATnn member of SYS1.NUCLEUS.

UNALLOCATE(*catname*)
Causes the integrated catalog facility catalog named in *catname* to be unallocated from the catalog address space. (Once an integrated catalog facility catalog is allocated to the catalog address space, it remains allocated for the life of the IPL unless you use this command to unallocate it.) UNALLOCATE frees all the catalog address space private storage used by the catalog, and unallocates the device from the catalog address space. CSA storage related to the catalog remains allocated. (To cause the CSA storage to be freed, you issue DELETE CATALOG.)

VCLOSE(*vvvvvv*)
Closes the VSAM volume data set (VVDS) whose volume serial is *vvvvvv* 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[(*catname***)]**
Provides the ability to activate catalog data space cache for a currently allocated and integrated catalog facility catalog.

catname specifies the integrated catalog facility catalog name.

WARNING
Tells the system to issue message IDC1997I or IDC1998I before performing an IDCAMS DELETE usercatalog FORCE, or DELETE VVDS RECOVERY. The messages require an operator response to allow the commands to complete.

The MODIFY CATALOG,REPORT command displays the current setting of this option under the heading ″DELETE UCAT/VVDS WARNING.″ For information about the forms of IDCAMS DELETE, see *z/OS DFSMS Access Method Services for Catalogs*.

At IPL time the setting of this option is WARNING; you may use a MODIFY CATALOG,NOWARNING command to turn off the request for operator intervention. See the NOWARNING form of the MODIFY CATALOG command for additional information.

For examples of the Modify Catalog command see *z/OS DFSMS: Managing Catalogs*.

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 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}]
      |,SM|
      |,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  }
      {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 Customization and Operation*. For information about network file system server messages (GFSAxxxi), use LookAt (see "Using LookAt to look up message explanations" on page xvii) or use the *MVS System Messages* books.

You can use the MODIFY command to:

- Display information about mount points or status of the network file system server.

- Request that an operation be performed on the mount points
- Set the correct level of message logging in the network file system server

Use this command only when the network file system server is operational.

```
F [MVSNFS.|jobname.]identifier,

    {FREEZE={ON|OFF}                }
    {LIST={MOUNTS|DSNAMES}          }
    {RELEASE=datasetname[(member)] }
    {STATUS                         }
    {STOP                           }
    {UNMOUNT=name                   }
```

The parameters are:

MVSNFS
The generic (IBM-supplied cataloged procedure) name assigned to the network file system server.

jobname
The name assigned to the network file server.

The name of a started task is determined 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 provided with the JCL JOB statement.
- A ***procedure***, the system will use the member name as the job name.

identifier
The identifier, from the START command, of the network file server to be modified.

If no identifier was specified, the identifier defaults to the job name. (Refer to “Displaying Started Task Status” on page 4-156 for information about determining the jobname and identifier of currently active started tasks.)

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, followed by an asterisk
- The entire identifier

When you use the asterisk format, the command takes effect on *all* started tasks that begin with the specified characters.

If you specify both the job name and the entire identifier, the command takes effect on *all* work units running with that combination of job name and identifier.

FREEZE=ON
Suspends processing of user mount requests. MVS rejects future mount requests by client users, and message “Permission Denied” displays on their monitors.

FREEZE=OFF
Resumes processing of a suspended user mount request. Client users can again mount MVS directories as normal.

LIST=MOUNTS
Returns a list of all mount points that are currently active in the network file system server, and their associated current use counts. A “mount point” is a place established in a client workstation that is used during the transparent accessing of a remote MVS file. Two entries must be created, one for a directory statement specified in the EXPORTS file for existing MVS data sets or MVS high-level qualifier, and one for an empty local directory created in the client workstation.

A “current use count” indicates how many mount requests have been made without an unmount request for the same mount point, regardless of which local directory the mount is attached to

LIST=DSNAMES
Returns a list of all data sets and PDS members that are either (1) in use by network file system server users or (2) opened for accessing but have not been yet closed by timeout (even though the file is not in use).

RELEASE=*datasetname***[(***member***)]**
Forces the network file system server to release a data set or PDS member that is currently being used. If the data set is active, the network file system server closes and deallocates it.

STATUS
Displays the status of the network file system server’s active subtasks.

STOP
Shuts down the network file system server. All current I/O operations are completed, and all open data sets are closed.

UNMOUNT=*name*
Unmounts a currently active mount point by removing the entry for the mount point from the file-handle database data set. This data set is used to record the active mounts during network file system server operation and allows clients to stay mounted when the network file system server is shut down and restarted.

Once the mount point is removed from the file-handle database data set, attempts to access this mount point from the client machine will fail, and the “Stale NFS File Handle” messages will display at the client machine. Enter the MODIFY UNMOUNT command from the client machine to end the stale file handle problem.

Example

To remove the mount point JONES from the file handle database, issue the following command:

```
F UNMOUNT=JONES
```

Collecting Problem Information for the Network File System Server

Use the MODIFY MVSNFS,LOG= command to collect diagnostic messages issued by the network file system server. These messages are sent to the console or the data set pointed to by the SYSERR DD statement in the startup procedure for the network file system server.

```
F [MVSNFS.|jobname.]identifier,LOG={ERROR|WARN|INFO|MEMSTATS}
```

The parameters are:

MVSNFS
The generic (IBM-supplied cataloged procedure) name assigned to the network file server.

jobname
The name assigned to the network file server.

The name of a started task is determined 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 provided with the JCL JOB statement.
- A ***procedure***, the system will use the member name as the job name.

identifier
The identifier, from the START command, of the network file server to be modified.

If no identifier was specified, the identifier defaults to the job name. (Refer to “Displaying Started Task Status” on page 4-156 for information about determining the jobname and identifier of currently active started tasks.)

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, followed by an asterisk
- The entire identifier

When you use the asterisk format, the command takes effect on *all* started tasks that begin with the specified characters.

If you specify both the job name and the entire identifier, the command takes effect on *all* work units running with that combination of job name and identifier.

LOG=ERROR *or* **WARN** *or* **INFO**
Collects messages issued by the network file system server, which stores messages in the data set specified in the SYSERR DD statement of the network file system server’s startup procedure.

The diagnostic message level can also be set in the EXEC statement of the network file system server startup procedure. LOG=ERROR collects error messages only, LOG=WARN collects error and warning messages only, and LOG=INFO collects error, warning, and informational messages.

LOG=MEMSTATS
Collects a “snapshot” of the memory use statistics for performance tuning or debugging. The network file system server logs and stores the statistics in the

data set specified in the SYSERR DD statement of the network file system server's startup procedure. This operand does not reset the level of diagnostic messages being collected.

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 the system in order to reinitialize the z/OS UNIX System Services environment. This command should be used only on a **limited** 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}
```

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.

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
```

Switching Workload Management Modes and Changing Resource States

Important
Beginning with z/OS V1R3, WLM compatibility mode is no longer available. Accordingly, you can no longer use the MODIFY WLM, MODE= command. The information has been left here for reference purposes, and for use on backlevel systems.

You can still use the MODIFY WLM, RESOURCE= command on all systems to change resource states.

Use the MODIFY WLM command to switch the workload management mode in effect on a system or to change the state of a resource. This command switches modes or 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.

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.

Before switching to goal mode, you should complete your service definition, install it in the WLM couple data set, and activate a service policy.

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,[MODE={GOAL|COMPAT}]
     ,[RESOURCE=resourcename,{ON|OFF|RESET}]
```

WLM
The name of the job.

MODE=
Changes the processing mode for workload management.

GOAL
Requests that the system switch from workload management compatibility mode to workload management goal mode.

Following a switch to goal mode, the system starts processing towards the goals specified in the active service policy.

COMPAT
Requests that the system switch from workload management goal mode to workload management compatibility mode.

Following a switch to compatibility mode, the system uses the IEAICSxx and IEAIPSxx parmlib members that were in effect when the system was last in compatibility mode. If the system was IPLed in goal mode, it issues messages prompting the operator for the IEAICSxx and IEAIPSxx parmlib members.

After switching to compatibility mode, a user may change the IEAICSxx and IEAIPSxx parmlib members that are in effect by entering the following command:

```
SET ICS=xx,IPS=xx
```

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 1

To change from compatibility mode to goal mode, enter:

```
F WLM,MODE=GOAL
```

If a service policy was active, workload management starts processing towards the goals defined within the service policy.

If a service policy was not active, workload management starts processing towards the goals defined in the default service policy.

Example 2

To change from goal mode to compatibility mode, enter:

```
F WLM,MODE=COMPAT
```

Even though the service policy is still active, the system does not use the goals to control resource access. Instead, it uses the performance criteria in the IEAICSxx and IEAIPSxx parmlib members that were in effect when the system was last in compatibility mode. If the system was initially loaded in goal mode, it issues messages prompting the operator for the IEAICSxx and IEAIPSxx parmlib members. You can specify them using the SET ICS=xx and SET IPS=xx commands.

Example 3

To change the setting of the DB2A resource to ON, enter:

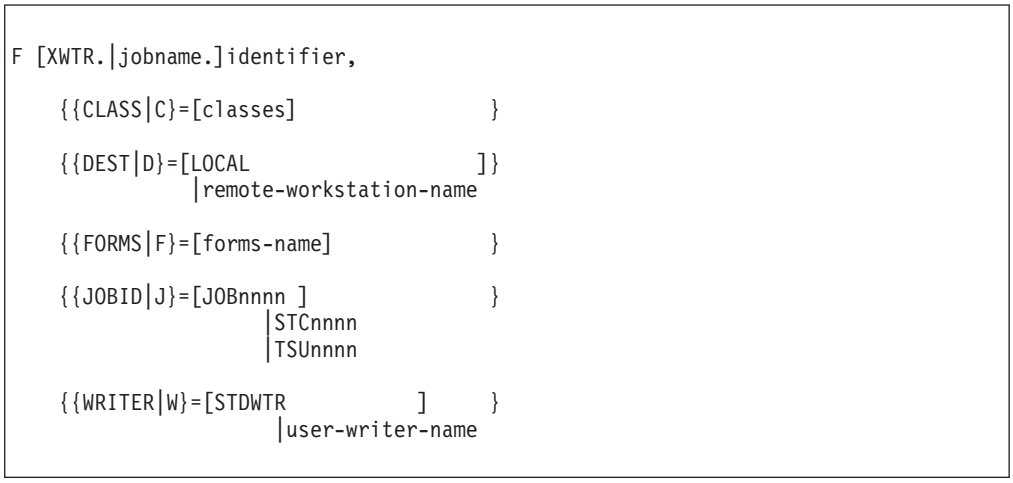
```
F WLM,RESOURCE=DB2A,ON
```

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.



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 4-284 for more information.

identifier
The identifier, from the START command, of the writer to be modified. (Refer to "Displaying Started Task Status" on page 4-156 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.
- *[/]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 4-284 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***nnnn***, STC***nnnn***,** *or* **TSU***nnnn*
Select only data sets from the job with this subsystem-assigned JOBID, where *nnnn* is the JOB id number or the STC id number or the TSU id number. If JOBID is omitted, the external writer does not select data sets by job.

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 4-284 for more information.

identifier
The identifier of the writer to be modified. (Refer to “Displaying Started Task Status” on page 4-156 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 4-284 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

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.
5. The DSNAME and SPACE operands are not routable.

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|cc|cca|name|name-a}] }
   {DSNAME                                  }
   {SPACE                                   }
   {STATUS[,L={a|cc|cca|name|name-a}]       }
   {SESS[,T][,L={a|cc|cca|name|name-a}]     }
```

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 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, cc, cca, name, or***,** *name-a*
The console id or name 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 on console 3 (and on all other consoles where the **MONITOR jobnames** command has been issued) the job name information and the time when each job starts and terminates, enter:

```
MN jobnames,T,L=3
```

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.

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=(sl,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=(sl,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 4-24 shows the possible volume and device combinations you can specify on the MOUNT command, and how the system will handle those inputs.

Table 4-24. 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

Use the MSGRT command to establish or change message routing instructions for displays from the DISPLAY, TRACK, or CONFIG commands to a specified message area, console, or both. MSGRT also controls the action of the CONTROL, MONITOR, STOPTR, 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                      {cc    }
     |K                         {cca   }
     |CF                        {name  }
     |MN                        {name-a}

   {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:

DMN, JOBS, J, TS, T

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.

K Subsequent CONTROL command action is to affect the specified MCS or

SMCS console until the option is discontinued or you IPL the system. Only those CONTROL functions that you can explicitly route with the L=cca or name-a operand can be routed by default using MSGRT.

CF
The system is to route the reply to the CONFIG command to the specified MCS or SMCS console.

L=*a, cc, cca, name, or name-a*
Specifies the display area (*a*), console (*cc*), both (*cca*), console name (*name*), or both (*name-a*) where the display will be presented. For *cc*, you must specify a decimal number from 1 to 99.

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 CONTn appears in the display response, all current routing instructions have not been displayed. Issue MR CONTn 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), TRACK (TR), 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=1),(K,L=2).
2. The MSGRT routing instructions for a console can be temporarily overridden by:
   - A command the system generates internally
   - A command on which you specify the location (L=cca) 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 12, and all TRACK A status displays in area A of the console through which you enter the MSGRT command, enter:

```
MR (D=U,L=12B),(TR=A,L=A)
```

Example 3

To present all DISPLAY SMS status displays in area B of console 14, enter:

```
MR D=SMS,L=14B
```

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 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).

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 then 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 the master console or its alternate:

```
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 4-25 summarizes the system requests for information for which you would use the REPLY command.

Table 4-25. Summary of the REPLY Command

Topic:
“Replying to System Information Requests” on page 4-329
“Replying to System Requests During Recovery Processing” on page 4-330
“Replying to System Security WTORs” on page 4-330
“Setting the Time-of-Day Clock” on page 4-330
“Specifying Component Trace Options” on page 4-331
“Specifying Dump Options” on page 4-332
“Specifying SMF Options” on page 4-333
“Specifying System Parameters” on page 4-333

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.

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]
```

Replying 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 zeroes 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
```

Replying to System Requests During Recovery Processing

During system recovery processing, normal console operations are suspended. The system uses the first available console specified in SYNCHDEST 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 SYNCHDEST. If there is no reply on the next console, then the system displays the WTOR messages to the master console. If there is no reply on the master console, then 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]
```

Replying 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][,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 GMT in your reply, the time and date are Greenwich mean time. Without the GMT parameter the system assumes the values are the local time and date, converts them to 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 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 GMT DATE=yyyy.ddd,CLOCK=hh.mm.ss
      IEA888A LOCAL DATE=yyyy.ddd,CLOCK=hh.mm.ss REPLY U, OR GMT/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 GMT parameter to change the value of the TOD clock:

```
R id,'[DATE=yyyy.ddd][,CLOCK=hh.mm.ss][,GMT]'
```

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 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 GMT time causes the system to reset the TOD clock and recalculate the local time value, using the new GMT and the system time zone constant.

Resetting local time does not affect 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.

If message IEA888A indicates that both GMT and local time values are incorrect, you should reset the 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 GMT. 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 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 4-533.

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]...) ]
```

```
Where s-option represents:

STRNAME=strname
  [,CONNAME=conname                                                    ]

  [,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 4-237.

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.

REPLY Command

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 cancelled 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 cancelled 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 4-26 summarizes the functions that the RESET command provides.

Table 4-26. 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 4-336

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.

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 CONTROL PROCESSING COMPLETE, after the command finally completes.

In a sysplex, if there is no z/OS level system active to process the RESET command, then RESET processing cannot be performed for that console. In that case the system will issue this message: IEE686I RESET COMMAND REJECTED, consname IS A SMCS CONSOLE.

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.

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=, SVRCLASS=, 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 police 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
```

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`.
  - 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 “Switching Workload Management Modes and Changing Resource States” on page 4-308 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 4-341.

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 T
     K S
     K V,USE
     MSGRT
     STOPTR
     TRACK
     V MSTCONS
     ```
   - 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
```

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.

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. However, any commands issued inside of ROUTE *ALL or ROUTE *sysgrpname* are not eligible for the MSGRT command.

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. Although IBM recommends 4-byte console IDs, we support 1-byte migration IDs.

  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
```

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                        }    {cc    }
            {*OTHER                            }    {cca   }
            {(sysname[,sysgrpname,sysname...])}    {name  }
                                                   {name-a}
```

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 or by a MSGRT 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, cc, cca, name***, or** *name-a*
The display area where the system is to display the command responses. You can specify the display area by area (*a*), console id (*cc*), or both (*cca*); or, you can specify the display area by area (*a*), console name (*name*), or both (*name-a*). For *cc*, you must specify a decimal number from 1 to 99.

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:

  ```
  RO sys1,D C,L=con1
  ```

  The following is not valid because the D T command does not support the L= operand:

  ```
  RO 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:

  ```
  RO 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 4-342.

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 master 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.

```
IEE421I RO *ALL,V 414,OFFLINE        FRAME  1     F      E   SYS=SYS1
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        FRAME LAST   F      E   SYS=SYS1
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 4-27 summarizes the information that the SEND command provides. Use it to access the pages on which you can find details about a particular use of the SEND command.

Table 4-27. 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 4-349
SEND ...,LOGON SEND ...,NOW	"Communicating with All Logged-On Terminal Users" on page 4-350
SEND ...,SAVE	"Saving Messages in the Broadcast Data Set" on page 4-352
SEND ...,LIST	"Listing the Notices Section of the Broadcast Data Set" on page 4-353
SEND ...,DELETE	"Deleting a Message from the Broadcast Data Set (Notices Section)" on page 4-354

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.

```
SE {'message'},{BRDCST            }
   {msgno    } {OPERATOR=routecode}
               {CN=console         }
```

The parameters are:

'message'
The message to be sent.

msgno
The number of the message in SYS1.BRODCAST to be sent. (See "Saving

Messages in the Broadcast Data Set" on page 4-352 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 3-5 on page 3-14).

CN=*console*
The console where the message is to be sent; *console* must be a one or two-digit decimal number between 0 and 99.

Note: When TSO/E 2.2 is installed, the CN keyword supports console names. The console name is 2 to 8 alphanumeric characters, the first character must be alphabetic or one of these characters, $, #, or @. MCS and SMCS consoles must be 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 4-111.

Example 1

To send message number 46 to console 12, enter:

```
se 46,cn=12
```

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 4-352. 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 received 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 4-352 for information about how to save and later send messages by message number.)

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.

```
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.BRODCAST 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 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, and highlighting 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 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 (APPC/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 run-time library services (RTLS) configuration.
- Dynamically specify which TSO/E parameter set should be activated.

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 4-28. 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. 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. The value can be changed by issuing the SET SMS=xx command and having MAXCFFEATURELEVEL specified in the IGDSMSxx parmlib member, or by using the SETSMS command with the MAXCFFEATURELEVEL keyword.

Syntax

The complete syntax for the SET command is:

```
T [[DATE=yyyy.ddd][,CLOCK=hh.mm.ss]][RESET]
  [,IPS=xx][,OPT=xx][,ICS=xx][,SMF=xx][,DAE=xx]
  [,MPF={(xx[,xx]...)|NO}]
  [,SLIP=xx][,PFK=xx][,IOS=xx][,EXS=xx]
  [,SMS=xx[,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]
```

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=, SCH=, GRSRNL=, CNGRP=, MPF=, OMVS=, or RTLS=, 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 GMT 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
The time zone constant which 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.

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*.

IPS=*xx*
The two alphanumeric characters indicating the IEAIPSxx member of the logical parmlib that contains the new parameters SRM is to use.

Note: The SET IPS command is not valid on systems operating in workload management goal mode. The command is supported on systems operating in workload management compatibility mode.

OPT=*xx*
The two alphanumeric characters indicating the IEAOPTxx member of the logical parmlib that contains the new parameters SRM is to use.

ICS=*xx*
The two alphanumeric characters indicating the IEAICSxx member of the logical parmlib that contains the new installation control specifications SRM is to use.

Note: The SET ICS command is not valid on systems operating in workload management goal mode. The command is supported on systems operating in workload management compatibility mode.

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=, ICS=, SMF=, and DAE= parameters affect the jobs in progress as well as the jobs read and scheduled after the command.

MPF=(*xx***[,***xx***]...)**
The two alphanumeric characters indicating the MPFLSTxx members of the logical parmlib that are concatenated to form the MPF table. xx can represent:

- 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
- 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 MPF parameter on 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 IGSMSxx member. (For a comparison of the SET SMS command with the SETSMS command, see Table 4-30 on page 4-406.)

You can use SET SMS=xx to specify an IGDSMSxx parmlib member containing 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 for EXTENDED 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 may have to issue the SET SMS=xx a second time to trigger the switch from NORMAL to EXTENDED sharing. Each of the systems will issue message IGW306I when it migrates to EXTENDED sharing.

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.

GRSRNL=(*xx***[,***xx***]...)**
The two alphanumeric characters indicating the GRSRNLxx member(s) of the logical parmlib that contain the specified GRS resource name lists. GRSRNL allows you to change the current RNLs specified in one or more GRSRNLxx parmlib member. 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, may become suspended or delayed, which can impair the performance of that critical application or the operating system itself.

For more detail on the use of the SET GRSRNL command, see *z/OS MVS Planning: Global Resource Serialization*.

Note: 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 and I/O timing processing.

The MIH timing intervals or the I/O timing intervals or the MIH timing intervals and the I/O timing intervals are changed ONLY for devices that have MIH statements coded in IECIOSxx.

Notes:

1. IECIOSxx may 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. Hot I/O processing parameters can only be changed 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 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.

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 SCHEDxx 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 SCHEDxx 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
- 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 4-390.

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 to add to the LPA following IPL, those to delete from the LPA, and threshhold values for minimum amounts of CSA storage that must still be 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 SETPROG command. See “SETPROG Command” on page 4-390. Note, however, that modules accessed via 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***[,***xx***...])**
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.

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.

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:

```
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 change the installation control and performance parameters according to the values found in the IEAICS00 and IEAIPS22 members of the logical parmlib, enter:

```
T ICS=00,IPS=22
```

Example 4

To restart SMF with the parameters found in the SMFPRMAA member of the logical parmlib, enter:

```
T SMF=aa
```

Example 5

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 6

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 7

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 8

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 9

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 10

To SET the console group definitions in the CNGRPAA member, enter:

```
SET CNGRP=AA
```

Example 11

To SET the console group definitions in the members CNGRPAA and CNGRPBB, enter:

```
SET CNGRP=(AA,BB)
```

Example 12

To remove all console group definitions from the sysplex, enter:

```
SET CNGRP=NO
```

Example 13

To change the MPFLSTxx member that builds the MPF table the system uses, enter:

```
SET MPF=06
```

Example 14

To change the MPFLSTxx members that builds the MPF table the system uses, enter:

```
SET MPF=(A1,A2,B4)
```

Example 15

To set the PROGxx member that the system uses to reference the APF list, enter:

```
SET PROG=03
```

Example 16

To set the current DIAGxx member to DIAG05, enter:

```
SET DIAG=05
```

Example 17

To change the product enablement policy with the parameters found in IFAPRDA2 and IFAPRDA3, enter:

```
SET PROD=(A2,A3)
```

Example 18

To change the RTLS configuration with the parameters found in parmlib members CSVRTLA2 and CSVRTLA3, enter:

```
SET RTLS=(A2,A3)
```

Example 19

To change to using the TSO/E parameters found in IKJTSOA1, enter:

```
SET IKJTSO=A1
```

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.

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

Use the SETGRS command to migrate a currently active GRS (global resource serialization) ring complex to a GRS star complex or to modify the current RESMIL or TOLINT values.

Syntax

The complete syntax for the SETGRS command is:

```
SETGRS {MODE=STAR                      }
       {[RESMIL=nnnnn][,TOLINT=nnnnn][,SYNCHRES={YES|NO}]}
         |RESMIL=OFF
```

Note: The installation's system programmer should direct use of this command.

Parameters

MODE=STAR
Directs the system to convert a GRS ring complex to a GRS star complex.

MODE=STAR is mutually exclusive with the RESMIL and TOLINT parameters.

RESMIL=*nnnnn* **| 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 value of RESMIL can be from 0 to 99999 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 *or* **NO**
Specifies whether synchronous reserve processing is to be activated.

Notes:

1. The system where you enter this command will control the migration.

2. Once GRS completes the transition to the star complex, the system issues the following message to indicate that the migration has completed and GRS is active for the complex:

   ```
   ISG334I GRS STAR COMPLEX INITIALIZATION COMPLETE
   ```

3. During processing of a SETGRS MODE=STAR command, no global resource requests (ENQ, DEQ, or RESERVE) will be processed. The length of time GRS requestors are suspended may be several minutes, because the GRS 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 GRS activity.
4. A SETGRS MODE=STAR request is valid if the following criteria are met:
   - GRS is running a ring complex.
   - All systems in the GRS ring complex support a star complex.
   - There are no systems in the GRS ring complex that are interconnected via GRS channel-to-channel support rather than the coupling facility.
   - All systems can access the ISGLOCK lock structure on the coupling facility.
   - The GRS 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.

Example

To migrate from a global resource serialization ring complex to star complex, enter:

```
SETGRS MODE=STAR
```

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 enable or disable the dynamic channel path management function, or to refresh the control unit model table for the single point of failure detection function in dynamic channel path management.

Syntax

The complete syntax for the SETIOS command is:

```
SETIOS DCM={YES|NO|REFRESH}
       MIH[,class=mm:ss[,class=mm:ss]...]
              [,MOUNTMSG={YES|NO}]
          [,DEV={([/]devnum[,[/]devnum]...)   },TIME=mm:ss,IOTIMING=mm:ss]
                {([/]lowdevnum-[/]highdevnum) }
          [,MSGONLY={YES|NO}]
```

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.

Parameters

The parameters are:

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.

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.

  (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.

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
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

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 LOAD*xx* 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, LPARNAME, 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.

SETLOGRC Command

Use the SETLOGRC command to change the logrec error and environmental recording medium originally specified in the IEASYSxx 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

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

```
SETOMVS [FORKCOPY=(COPY|COW)]
        [,IPCSEMNIDS=ipcsemnids]
        [,IPCSEMNOPS=ipcsemnops]
        [,IPCSEMNSEMS=ipcsemnsems]
        [,IPCMSGQBYTES=ipcmsgqbytes]
        [,IPCMSGNIDS=ipcmsgnids]
        [,IPCSHMMPAGES=ipcshmmpages]
        [,IPCSHMNIDS=ipcshmnids]
        [,IPCSHMNSEGS=ipcshmnsegs]
        [,IPCSHMSPAGES=ipcshmspages]
        [,IPCMSGQMNUM=ipcmsgqmnum]
        [,LIMMSG=[NONE|SYSTEM|ALL]]
        [,MAXASSIZE=maxassize]
        [,MAXCORESIZE=maxcoresize]
        [,MAXCPUTIME=maxcputime]
        [,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]
        [,PRIORITYPG=(n) | NONE] ;
        [,RESET=(xx)]
        [,STEPLIBLIST='stepliblist']
        [,SUPERUSER=superuser]
        [,SYNTAXCHECK=(xx)]
        [,TTYGROUP=ttygroup]
        [,USERIDALIASTABLE=useridaliastable]
        [,VERSION='string']
```

SETOMVS EXTENSIONS (sysplex exclusive)

```
SETOMVS   FILESYS
         ,FILESYSTEM=filesystem
         ,AUTOMOVE=YES|NO|UNMOUNT
         ,SYSNAME=sysname|*       or
SETOMVS   FILESYS
         ,FILESYSTEM=filesystem
         ,AUTOMOVE=YES|NO|UNMOUNT   or
SETOMVS   FILESYS
         ,FILESYSTEM=filesystem
         ,SYSNAME=sysname|*       or
SETOMVS   FILESYS
         ,MOUNTPOINT=mountpoint
         ,AUTOMOVE=YES|NO|UNMOUNT
         ,SYSNAME=sysname|*       or
SETOMVS   FILESYS
         ,MOUNTPOINT=mountpoint
         ,AUTOMOVE=YES|NO|UNMOUNT   or
SETOMVS   FILESYS
         ,MOUNTPOINT=mountpoint
         ,SYSNAME=sysname|*       or
SETOMVS   FILESYS
         ,FROMSYS=sysname
         ,SYSNAME=sysname|*
```

Note: FILESYSTEM, **FROMSYS**, and **MOUNTPOINT** are mutually exclusive parameters. When you specify **FILESYS**, you must supply one of these three parameters.

Parameters

AUTOMOVE = YES | NO | UNMOUNT, FILESYS=filesys, FILESYSTEM=filesystem, FROMSYS=sysname, MOUNTPOINT=mountpoint, SYSNAME=sysname|*, and VERSION='nnnn' , which are described in this section, are parameters that are used in a sysplex environment where systems are exploiting shared HFS. For more information on shared HFS in a sysplex, see *z/OS UNIX System Services Planning*.

The parameters are:

AUTOMOVE=YES | NO | UNMOUNT
The AUTOMOVE, NOAUTOMOVE, and UNMOUNT parameters apply only in a sysplex where systems are participating in shared HFS. These parameters indicate what happens if the system that owns a file system goes down. AUTOMOVE indicates that ownership of the file system automatically changes to another system participating in shared HFS. NOAUTOMOVE indicates that ownership of the file system is not moved if the owning system goes down; as a result, the file system becomes inaccessible. UNMOUNT indicates that the file system should be unmounted if the system's owner should crash. This filesystem and any filesystems mounted within its subtree will be unmounted. AUTOMOVE is the default.

Note: **AUTOMOVE** is not allowed when moving multiple filesystems. Also, to ensure that the root is always available, use the default for AUTOMOVE.

FILESYS=filesys
In a sysplex environment, this parameter alerts the parser that commands which change mount attributes are forthcoming.

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 45 character alphanumeric field that denotes the name of the filesystem to be changed or moved. This filesystem name may be in the following form: 'OMVS.USER.JOE'. **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*.

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.

IPCMSGQBYTES = 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.

IPCMSGNIDS = ipcmsgnids
Specifies the maximum number of unique message queues in the system. The range is from 1 to 20 000. The default is 500.

IPCSHMMPAGES = ipcshmmpages
Specifies the maximum number of pages for a shared memory segment. The range is from 1 to 25 600. The default is 256.

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.

IPCSHMSPAGES = ipcshmspages
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.

IPCMSGQMNUM = ipcmsqgmnum
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).

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 proces 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).

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 initated

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 65 535.

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.

MAXMMAPAREA = maxmmaparea
Specifies the maximum amount of data space storage (in pages) that can be allocated for memory mappings of HFS files. Storage is not allocated until memory mappings are active.

The range is from 1 to 16 777 216. The default is 4 096.

MAXPROCSYS = maxprocsys
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 = maxprocuser
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 an z/OS UNIX System Services process. The range is 3 to 32 767;

MAXPTYS = maxptys
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 = maxsharepages
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.

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.

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.

PRIORITYPG = (n) | NONE
Specify from 1 to 40 performance group numbers, in a range from 1 to 999. If

you do not specify this statement, or if you specify NONE, no array is created for it. If you specify fewer than 40 values, the last value specified is propagated to the end of the array.

All performance groups specified on the PRIORITYPG statement must also be specified in the IEAIPSxx parmlib member.

Generally, we do not recommend that you set PRIORITYPG.

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.

Note: SETOMVS RESET accepts only a single parameter; SET OMVS accepts more than one parameter.

For more detailed information about the RESET parameter see *z/OS UNIX System Services Planning*.

Table 4-29. 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	No	Yes
FORKCOPY	Yes	Yes
FROMSYS	No	No
IPCMSGNIDS	Yes	Yes
IPCMSGQBYTES	Yes	Yes
IPCMSGQMNUM	Yes	Yes
IPCSEMNIDS	Yes	Yes
IPCSEMNOPS	Yes	Yes
IPCSEMNSEMS	Yes	Yes
IPCSHMMPAGES	Yes	Yes
IPCSHMNIDS	Yes	Yes
IPCSHMNSEGS	Yes	Yes
IPCSHMSPAGES	Yes	Yes
MAXASSIZE	Yes	Yes
MAXCORESIZE	Yes	Yes
MAXCPUTIME	Yes	Yes
MAXFILEPROC	Yes	Yes
MAXFILESIZE	Yes	Yes

Table 4-29. Acceptable Parameter Statements and Their Applicability (continued)

Parameter Statement	SET OMVS= (xx, yy, ...)	SETOMVS RESET= (xx)
MAXMMAPAREA	Yes	Yes
MAXPROCSYS	Yes	Yes
MAXPROCUSER	Yes	Yes
MAXPTYS	Yes	Yes
MAXSHAREPAGES	Yes	Yes
MAXTHREADS	Yes	Yes
MAXTHREADTASKS	Yes	Yes
MAXUIDS	Yes	Yes
MOUNT	No	No
MOUNTPOINT	No	No
NETWORK	No	Yes
PRIORITYGOAL	Yes	Yes
PRIORITYPG	Yes	Yes
ROOT	No	No
RUNOPTS	No	No
STARTUP_EXEC	No	No
STARTUP_PROC	No	No
STEPLIBLIST	Yes	Yes
SUBFILESYSTYPE	No	Yes
SUPERUSER	Yes	Yes
SYSCALL_COUNTS	Yes	Yes
SYSNAME	No	No
SYSPLEX	No	No
TTYGROUP	Yes	Yes
USERIDALIASTABLE	Yes	Yes
VERSION	Yes	Yes

STEPLIBLIST = 'stepliblist'

Specifies the path name of a hierarchical file system (HFS) 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 an OS/390 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 only syntax and does not verify that HFS data sets or mount points are valid.

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 HFS. 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 HFS. 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 HFS 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 an OS/390 group name. Slave pseudoterminals (ptys) and OCS rtys 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 hierarchical file system (HFS) 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 HFS. VERSION allows multiple releases and service levels of the binaries to coexist and participate in HFS sharing. A directory with the value *nnnn* specified on VERSION is dynamically created at system initialization under the sysplex root and is used as a mount point for the version HFS. This directory, however, is only dynamically created if the sysplex root HFS 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 HFS. 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 HFS, see ″Shared HFS in a Sysplex″ in *z/OS UNIX System Services Planning*.

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.

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 4-355 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.

DSNAME|LIBRARY=*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.

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                  }
```

Note: This command requires a /* */ around comments. Refer to “System Command Formats” on page 4-13 for further information.

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.

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}                                   }
                       {ASID=asid  }
               {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.

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.

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=*volser*
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 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 4-404.

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.

UNALLOCATE
Indicates that you want to undo all existing allocations obtained while processing active LNKLST sets. This also releases the SYSDSN ENQ.

Note: Once you have completed everything associated with the UNALLOCATE, you must specify SETPROG 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.

You must assure that while the allocations are not in effect the LNKLST data sets are not deleted or moved.

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 threshhold 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.

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.

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 | LNKLST]
                 [,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 use its normal search protocol (search LPA then 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

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
```

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:

```
SS 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 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.

For more information on the ACDS, SCDS, and COMMDS data sets, see *z/OS DFSMSdfp Storage Administration Reference* and *z/OS DFSMSdfp Diagnosis Reference*.

Note: If you are not sure about the differences between SET SMS or SETSMS, see Table 4-30.

Table 4-30. 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.	No default values. Unspecified parameters remain unchanged.

Some combinations of SETSMS parameters are not valid. Table 4-31 shows these incorrect combinations.

Note: The SCDS is a required parameter if the ACDS does not contain a valid configuration.

Table 4-31. 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.

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)
```

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 or SCDS.

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.

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.

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.

CACHETIME(*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(*nnnnn*)
Specifies that SMS is to allow *nnnnn* seconds (1 to 86399) to elapse between the production of SMS BMF records. The default is 3600 seconds.

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. While this parameter sets the *maximum* value, BMF will dynamically change the *actual* number of times 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 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.

Trace 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), JOBNAME (*), 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 nnnnnn***K,** *or nnn***M)**
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)**
SMS is to trace error events or all events. The default value from SMS initialization is ERROR. The new TYPE value replaces the previous value.

JOBNAME(*jobname or* ***)**
SMS is to limit tracing to the specific job named *jobname*. If you enter ***, *all* jobs are traced.

If you specify JOBNAME, omit ASID.

ASID(*asid or* ***)**
SMS is to limit tracing to the specified address space. If you enter ***, *all* address spaces are traced. If you specify ASID, omit JOBNAME.

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” on page 4-411 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” on page 4-411 for a complete list of the options.

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(*nnn* **or 3600)**
Specifies the number of seconds between recording SMF 42 records related to SMSVSAM coupling facility use (subtypes 15, 16, 17 and 18). SMF_TIME(YES) overrides this parameter.

DEADLOCK_DETECTION(*iiii,kkkk***)**
Specifies the deadlock detection intervals used by SMS.

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.

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

 TRACE    = OFF   SIZE = 16k        TYPE = ERROR
 JOBNAME  = *ASID=*
 TRACING EVENTS:
 MODULE   =  OFF    SMSSJF   =  OFF    SMSSSI   =  OFF    ACSINT   =  ON
 OPCMS    =  OFF    CONFC    =  ON     CDSC     =  OFF    CONFS    =  OFF
 MSG      =  ON     ERR      =  OFF    CONFR    =  OFF    CONFA    =  OFF
 ACSPRO   =  OFF    IDAX     =  OFF    DISP     =  OFF    CATG     =  OFF
 VOLREF   =  OFF    SCHEDP   =  OFF    SCHEDS   =  OFF    VTOCL    =  OFF
 VTOCD    =  OFF    VTOCR    =  OFF    VTOCC    =  OFF    VTOCA    =  OFF
 RCD      =  OFF    DSTACK   =  OFF
```

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)
```

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
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)
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
ALL	All of the options

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 ENF-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.

CAUTION:
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 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 MSTJXLxx'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 | 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 ENF-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 | 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 | SUB | 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.

- 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
```

SETXCF Command

Use the SETXCF command to control the cross-system coupling facility (XCF). Table 4-32 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 4-32. Summary of the SETXCF Command

Command	Topic
SETXCF COUPLE	"SETXCF COUPLE Command"
SETXCF FORCE	"SETXCF FORCE Command" on page 4-419
SETXCF MODIFY	"SETXCF MODIFY Command" on page 4-421
SETXCF PRSMPOLICY	"SETXCF PRSMPOLICY Command" on page 4-423
SETXCF START	"SETXCF START Command" on page 4-423
SETXCF STOP	"SETXCF STOP Command" on page 4-429

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 4-33. 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	Have sysplex scope only when you specify POLICY or REBUILD.

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 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 in the master catalog. You need to use parentheses only when you specify the volume. Once 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, other than sysplex data, stored in this couple data set.

The supported names are:
- ARM for automatic restart management
- CFRM for coupling facility resource management
- LOGR for system logger
- SFM for sysplex failure management
- WLM for workload management

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.

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 in the master catalog. If the volume is specified, the system does not use the master 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, 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, 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 resources related to structures in a coupling facility. The resources can be either structures actively in use in the sysplex or dumps associated with structures pending deallocation.

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 on a system that does not have connectivity to the coupling facility that contains the structure named, 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.

Attention: To reduce the risk of losing data, do not force the deletion of a structure unless you understand its 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 FORCE,
    {STRUCTURE,STRNAME=(strname[,strname]...)                          }

    {CONNECTION,STRNAME=strname,CONNAME={(conname[,conname]...)|ALL} }

    {STRDUMP,STRNAME=strname[,STRDUMPID=strdumpid]                     }

    {STRDUMPSERIAL,STRNAME=strname[,STRDUMPID=strdumpid]               }
```

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. The structure must have no active or failed-persistent connections to be deleted with the SETXCF FORCE command.

CONNECTION *or* **CON**
Directs the system to force the deletion of a failed-persistent connection. If this is the last connection to a non-persistent structure, delete the structure as well.

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.

CONNAME= *or* **CONNM=(***conname***[,***conname***]...)** *or* **ALL**
Directs the system to delete one more 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.

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.

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.

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.

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 PRSMPOLICY Command

Use the SETXCF PRSMPOLICY (*or* PRSMPOL) 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 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

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]...)]

             {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                   }
```

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

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.

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.

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 characters. 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***[,***cfname***]...)**
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.

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.

  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.

Note that before the rebuild process begins, you might need to change to 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.

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.)

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.
- 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.

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                          }

            {POLICY,TYPE=name                                  }

            {REBUILD,{POPULATECF=cfname}                       }
                     {DUPLEX,}
                     {STRNAME=(strname[,strname]...)}
                         {,KEEP=NEW|OLD}
                     {CFNAME=(cfname[,cfname]...)   }

            {ALTER,STRNAME=strname                             }
```

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.

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.

POLICY *or* **POL,TYPE=***name*
Directs the system to deactivate the active policy for the type of service named.

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=NEW|OLD
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 structue 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 IXLALTER programming interface.

STRNAME *or* **STRNM=***strname*
Identifies the structure for which to stop structure alter processing.

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 4-34. Summary of the SLIP Command

Command:	Topic:
SLIP SET	“Setting a SLIP Trap” on page 4-439
SLIP MOD	“Modifying an Existing SLIP Trap” on page 4-498
SLIP DEL	“Deleting an Existing SLIP Trap” on page 4-499

General information about the SLIP command is in the following topics:
- “Syntax”
- “Using SLIP Commands”
- “Processing of SLIP Commands” on page 4-434
- “Coding SLIP Command Parameters” on page 4-435

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 4-446
- “Syntax for an Instruction Fetch or Successful Branch SLIP SET PER Command” on page 4-447
- “Syntax for a Storage Alteration SLIP SET PER Command” on page 4-448
- “Syntax for the ACTION Parameters for the SLIP SET Command” on page 4-449
- “Modifying an Existing SLIP Trap” on page 4-498
- “Deleting an Existing SLIP Trap” on page 4-499

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 4-340 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 4-442.

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, REFBEFOR, 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. 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.
5. 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
```

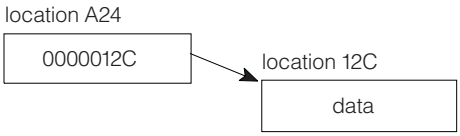
```
128 is a direct address.
% signs indicate 24-bit indirect addressing.
+4, +8, and +C are displacements.
```

In the expression

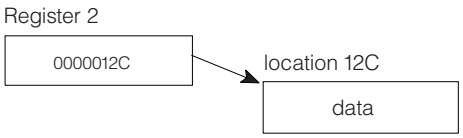
```
2R??+4?+8?+C
```

```
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.
```

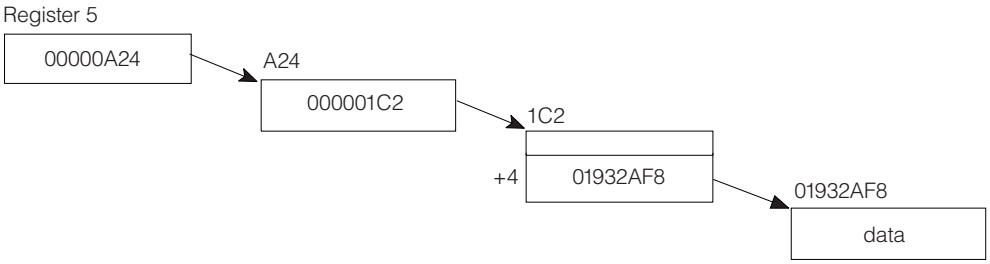
To refer to data when the address of the data is located at A24, specify `A24?`. Graphically:



To refer to data when the address of the data is in general purpose register 2, specify `2R%`. Graphically:

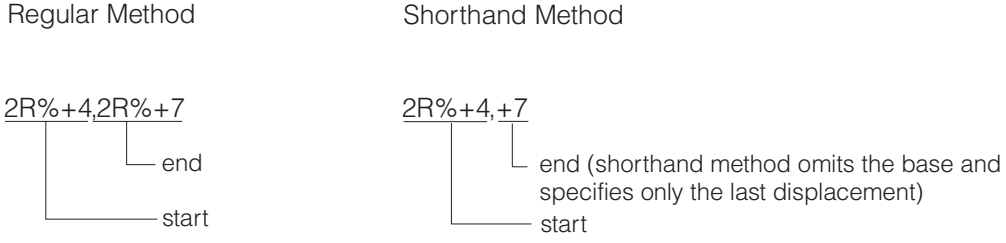


You can indicate as many levels of indirect addressing as necessary by following the initial indirect address with a corresponding number of percent signs or question marks. You can also include plus or minus displacement values. For instance, you can specify. `5R%%+4?`. Graphically:



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.

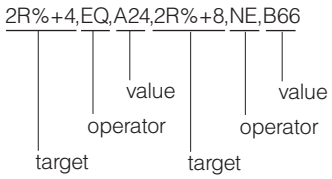


When you are using shorthand to specify an address range, make sure that the ending address is greater than the starting address. Because SLIP does not consider the base to include the last displacement, the displacement you specify for the ending address must be greater than the displacement you specify for the starting address.

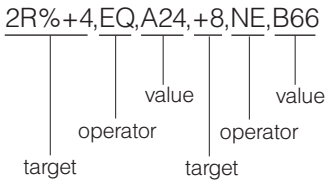
For example, **it would be incorrect** to specify a range as `2R%+4,+3`. An error condition exists because the ending address is less than the starting address.

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,



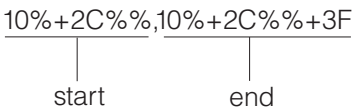
is written using shorthand form as



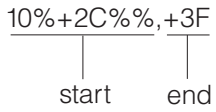
You must establish a direct or indirect base address before using displacements. In the following example, 2R specifies the contents of general purpose register 2, not an address. Therefore, the example using a +6 displacement is incorrect.

```
2R,EQ,C12,+6,NE,D01
```

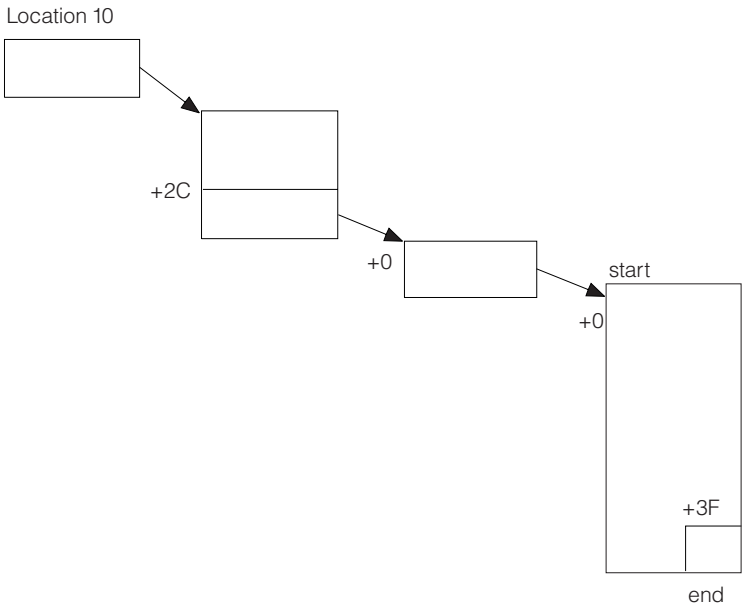
Indirect Addressing with the LIST, RANGE, SUMLIST and TRDATA Parameters: When indirect addressing is used with the LIST, SUMLIST, RANGE or TRDATA parameters, the indirect addresses point to the beginning and end of a field of data. The following example shows a starting and ending address:



The same example in shorthand is:



Graphically:



Qualifying Direct or Indirect Addresses to Address Spaces: You can qualify direct or indirect addresses in the DATA, REFBEFOR, 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
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=(3.3FC210,3FC217,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%)
```

Setting a SLIP Trap

The SLIP commands used to set SLIP traps are described in the following topics:

- “Structure of a SLIP SET Command” on page 4-440
  - Events
  - Conditions
  - Actions
  - Types of SLIP SET Parameters
- “Setting Effective SLIP PER Traps” on page 4-442
  - 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 4-446

- "Syntax for an Instruction Fetch or Successful Branch SLIP SET PER Command" on page 4-447
- "Syntax for a Storage Alteration SLIP SET PER Command" on page 4-448
- "Syntax for the ACTION Parameters for the SLIP SET Command" on page 4-449
- "SLIP SET Parameters" on page 4-455

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
  - 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 hiperspace.

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 4-35. 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 JOBNAME PSWASC ASID JSPGM PVTEP ASIDSA LPAEP PVTMOD COMP LPAMOD RANGE DATA MODE REASON DSSA NUCEP ERRTYP NUCMOD

Table 4-35. Summary of the SLIP SET parameters (continued)

Function:	Parameters:
Action parameters: These parameters specify what the system is to do when the trap matches.	IGNORE RECORD SYNCSVCD NODUMP RECOVERY TARGETID NOSUP REFAFTER TRACE NOSVCD REFBEFOR TRDUMP NOSYSA STDUMP WAIT NOSYSM STRACE NOSYSU SVCD The following parameters are options to tailor the action that the system is to do. ASIDLST REMOTE STRLIST DSPNAME REFAFTER SUMLIST JOBLIST REFBEFOR TARGETID LIST SDATA TRDATA
Trap control parameters: These parameters control the operation of the trap.	DISABLE IDGROUP PRCNTLIM ENABLE MATCHLIM
Specialized parameters:	DEBUG ID RBLEVEL END OK

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 4-443.*

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.
- 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.

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]...)               ]


 [,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 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
```

```
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
    [,CONNAME=conname]
    [,ACCESSTIME={ENFORCE | NOLIMIT}]
    [,LOCKENTRIES]
    [,USERCNTLS]
    [,EVENTQS]
    [,(EMCONTROLS={ALL | (list)})]
    [,({COCLASS | STGCLASS | LISTNUM}={ALL | (list)}]
      {[,ADJUNCT={CAPTURE|DIRECTIO}][,ENTRYDATA={UNSERIALIZE|SERIALIZE}]})
      {[,SUMMARY]                                                       })
```

```
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
    [,CONNAME=conname]
    [,ACCESSTIME={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
    [,CONNAME=conname]
    [,ACCESSTIME={ENFORCE | NOLIMIT}]
    [,LOCKENTRIES]
    [,USERCNTLS]
    [,EVENTQS]
    [,(EMCONTROLS={ALL | (list)})]
    [,({COCLASS | STGCLASS | LISTNUM}={ALL | (list)}]
      {[,ADJUNCT={CAPTURE|DIRECTIO}][,ENTRYDATA={UNSERIALIZE|SERIALIZE}]})
      {[,SUMMARY]                                                       })
```

```
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
    [,CONNAME=conname]
    [,ACCESSTIME={ENFORCE | NOLIMIT}]
    [,LOCKENTRIES]
    [,USERCNTLS]
    [,EVENTQS]
    [,(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
- SVCD

- SYNCSVCD
- 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.

NOSYSA
Suppresses all SYSABEND dumps requested while the system processes the error.

NOSYSM
Suppresses all SYSMDUMP 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

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 TRACE and TRDUMP keywords. For example, if you specify ACTION=(STOPGTF,TRACE), then SLIP produces a GTF trace records until the trap becomes disabled.

ACTION=SVCD

When the trap matches for an error or PER event, schedules an SVC dump for the current or failing address space. 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.

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.

Caution: 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. The SYNCSVCD parameter overrides DAE suppression.

This ACTION value is not valid for an error event trap.

The ASIDLST, DSPNAME, 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

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, 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
```

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* **| '***jobname***'[,** *asid* **| ,'***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 *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,'JOB1','JOB*')
```

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 hiperspace.

```
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 Udddd, 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. If you specify any of the following abend completion codes, the SLIP action is not taken: 11A, 12E, 15D, 15F, 200, 212, 25F, 279, 282, 42A, 430, 57D, 700, 72A, A00, B00, and E00.

   Each of these codes is originally a program check (code 0C4) that the system converts to a new value. If you want to specify a program check, use COMP=0C4 or ERRTYP=PROG. To avoid having the SLIP action occur for all program checks, you should also specify a program name, module name, or other condition.
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.

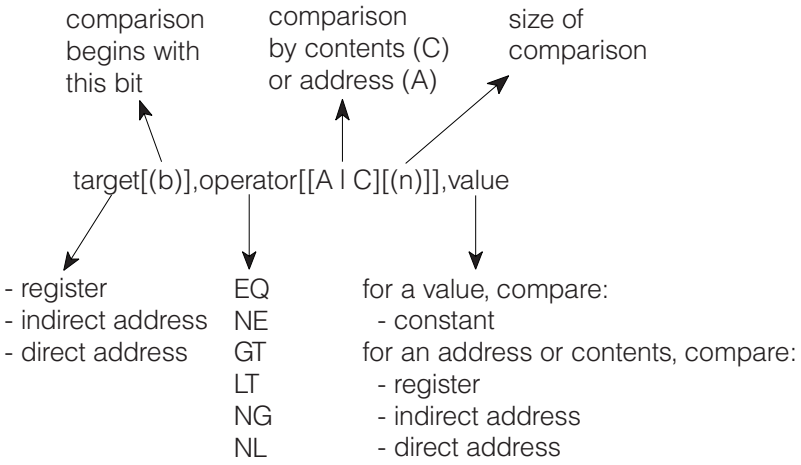
Abbreviation: C

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 4-438.)
- A GPR in the form xG or xR where x is 0 – 15.
- An indirect address (see “Indirect Addresses” on page 4-435)

If SLIP cannot establish addressability to the target location, SLIP issues message IEA413I and increments the counter for the trap.

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)
```

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
'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*
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)
```

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
```

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 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. 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 4-435.) 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%+2C%%,+3F)
```

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 SB 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=(IEAVTXYZ,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 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

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 message ID consists of the characters up to the first blank in the WTO, but is never more than 10 characters. The slip action processor gets control after SSI and MPF processing.

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 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=(IEAVTRTS,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.

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.

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) =(*'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 8 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 procesing 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. 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 4-435.

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.
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 REFAFTER or REFBEFOR.

The following syntax for a triplet is identical for REFAFTER and REFBEFOR.

target[(b)],**EQ**|**EQA**|**EQC**[(*n*)],*value*

The parameters in a *triplet* are:

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 4-435)

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 4-435)

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 REFBEFOR 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,LPAMOD=(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,LPAMOD=(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,LPAMOD=(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, SYNCSVCD, or WAIT.

The parameters within the REMOTE parameter are: UNCOND, COND, SYSLIST, ACTION, ASIDLIST, 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 SYNCSVCD, 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 ASIDLIST 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.

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.

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
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 SYNCSVCD: SQA.
- For ACTION=STDUMP or TRDUMP: NOSQA.

If you do not specify SDATA, the default SDATA options are:
- For ACTION=SVCD or SYNCSVCD: 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. Even when you specify SDATA, the system issues a SLIP dump command with the value of the QUIESCE option set by the CHNGDUMP command.

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

CONNAME=*conname*
In the STRLIST parameter for a coupling facility cache structure, requests the user registry information for this user be included in the dump. The *conname* is the name of a connected user. If the connected user represented by the *conname* 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 ACCESSTIME parameter of the IXLCONN macro is in effect.

When ACCESSTIME=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 ACCESSTIME=0 is specified on the IXLCONN macro and ACCESSTIME=ENFORCE is specified on the SLIP command, the structure will not be included in the dump.

When ACCESSTIME=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 ACCESSTIME
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).

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 ACCESSTIME=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.

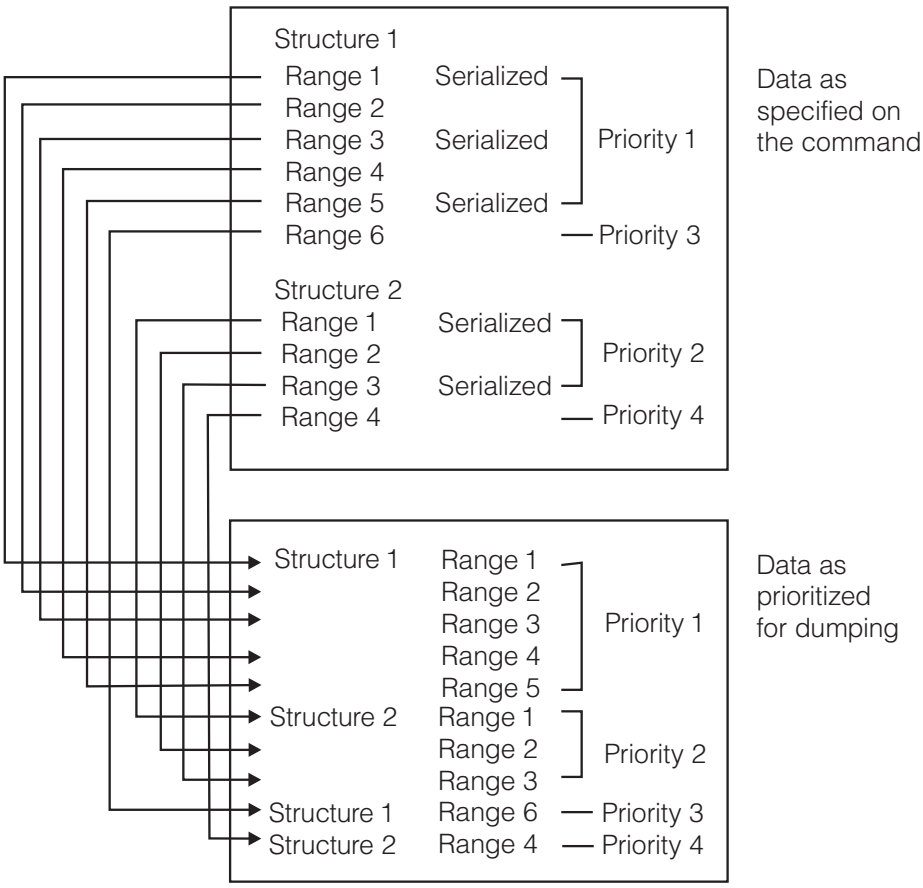
Abbreviation: SUM

Notes for the STRLIST Parameter:

1. A syntax error message is issued if STRNAME is not the first parameter in STRLIST.
2. If CONNAME and ACCESSTIME are specified more than one time for a structure, the first CONNAME and the last ACCESSTIME 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 4-435.) A starting address must be less than or equal to the ending address.

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 CVT. See “Dynamic PER Traps” on page 4-445 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 4-435.)

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

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 JOBNAME 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),JOBNAME=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),JOBNAME=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 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,F116),ACTION=WAIT,ML=1,END
```

Example 12

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 13

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 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 14

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 15

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 16

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 17

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 18

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 19

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 20

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 21

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

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.

CAUTION:
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 4-36. Summary of the START Command

Topic:	Command:
“Starting a System Task from a Console”	START membername
“Starting the APPC/MVS Address Space” on page 4-505	START APPC
“Starting the APPC/MVS Transaction Scheduler Address Space” on page 4-506	START ASCH
“Starting the DFSMS Offering” on page 4-507	START DFSMSPKG
“Starting the Generalized Trace Facility” on page 4-507	START GTF
“Starting the Library Lookaside (LLA) Address Space” on page 4-509	START LLA
“Starting the Object Access Method (OAM)” on page 4-510	START OAM
“Starting Resource Recovery Services (RRS)” on page 4-510	START RRS
“Starting the System Object Model (SOM) Subsystem” on page 4-511	START SOM
“Starting TSO/VTAM Time-Sharing” on page 4-512	START membername
“Starting the Virtual Lookaside Facility or Data Lookaside Facility” on page 4-513	START VLF or START DLF
“Starting an External Writer” on page 4-514	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 *devicetype* (or *devnum*), or *classes*, or *volumeserial*, 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
- 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 in a variety of ways. If you use the JOBNAME= parameter on the START command, the job name is assigned as indicated. Otherwise:

- If a JOB statement exists within the member, the system will assign the job name according to the JOB statement.
- If a JOB statement does not exist, the system will automatically assign the member name as the job name. In the command `START CICS`, CICS will be assigned as the job name.

An identifier may be specified on the START command (to identify the started task for the operator). In the command `START CICS.CICSA`, CICS is the member name and CICSA will be assigned as the identifier.

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 cataloged 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 MSTJXLxx data set, will be unavailable and will therefore not be found; the system will issue message IEFC612I.

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,JOBACCT=(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[,APPC=(nn[,nn]...[,L])]
```

The parameters are:

APPC,SUB=MSTR
Invokes the APPC/MVS procedure and creates the APPC/MVS address space.

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*.

L The system will display the parmlib statements on the operator's console as it processes them.

Starting the DFSMS Offering

Use the START DFSMSPKG command instead of re-ipling, to allow the system to read the DFSMS_OFFERING statements specified in the IGDDFPKG parmlib member. IGDDFPKG specifies the Data Facility Storage Management Subsystem (DFSMS/MVS) functional components that are licensed for use on the system.

The system will issue message IEA287I for each DFSMS_OFFERING statement.

```
S DFSMSPKG
```

There are no parameters for DFSMSPKG.

For more information about IGDDFPKG, see *z/OS MVS Initialization and Tuning Reference*. For more information about Message IEA287I, use LookAt (see "Using LookAt to look up message explanations" on page xvii) or refer to the *MVS System Messages* books.

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.

BLOK=
Reserves common storage buffers to collect GTF data.

*nnn***K** *or nn***M**
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=*nnnn***K**
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={***nnn***K** *or nn***M}**
The number of bytes of GTF data that will appear in a stand alone dump. The default is 40K.

SDUMP= *or* **SD={***nnn***K** *or nn***M}**
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={***nnn***K** *or nn***M}**
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 name of the subsystem that will process the task is the master subsystem. If you omit this parameter, the JES subsystem scheduler starts LLA. The resulting dependency on JES requires LLA to be stopped when stopping JES.

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*.

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(ATRRRS) 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 cataloged 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
```

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.

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

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 4-37. Summary of the STOP Command

Topic:	**Command:**
"Stopping an Address Space"	STOP jobname.identifier
"Stopping an ASCH Initiator" on page 4-519	STOP ASCHINT
"Stopping the Data Lookaside Facility (DLF)" on page 4-519	STOP DLF
"Stopping the Library Lookaside (LLA) Address Space" on page 4-519	STOP LLA
"Stopping the Object Access Method (OAM) Address Space" on page 4-520	STOP OAM
"Stopping a System Object Model (SOM)" on page 4-520	STOP SOM
"Stopping the Virtual Lookaside Facility (VLF)" on page 4-521	STOP VLF

Syntax

The syntax for each variation of the STOP command is shown immediately preceding its respective parameter list.

STOP or P

Stopping an Address Space

You can use the STOP command to stop an address space. Note that you can communicate with the currently running program only if it was designed to recognize input from 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 job.

The name of an address space is determined based on whether the JOBNAME= keyword was specified on the START command.

If JOBNAME= was specified, *jobname* is the name assigned to the address space.

If JOBNAME= was not specified and the source JCL for the address space is:

- A ***job***, the system will use the job name provided with the JCL JOB statement.
- A ***procedure***, the system will use the member name as the job name.

identifier
The identifier assigned to the job or address space. (Refer to "Displaying Started Task Status" on page 4-156 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.
- *[/]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 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

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:

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 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

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.

Scope in a Sysplex

This command has sysplex scope only if 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 STOPMN command is:

```
PM {JOBNAMES[,L={a|cc|cca|name|name-a}]}
   {DSNAME                            }
   {SPACE                             }
   {STATUS[,L={a|cc|cca|name|name-a}]  }
   {SESS[,L={a|cc|cca|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, cc, cca, name, or name-a*
The console where the display is to appear unless routing instructions are in effect. See the MSGRT command in this chapter. If you omit this keyword, the display appears 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

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|cc|cca|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, cc, cca, name, or name-a*
The display area (*a*), console identifier (*cc*), both (*cca*), 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 unless routing instructions are in effect; see the MSGRT command in this chapter.

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 console 22, enter:

```
pt a,l=22a
```

Example 2

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 *z/OS MVS Recovery and Reconfiguration Guide*.

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 4-38 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 4-38. Summary of the SWAP Command

Command:	Topic:
SWAP devnum1,devnum2	"Operator-Requested DDR"
SWAP OFF SWAP ON	"System-Initiated DDR" on page 4-525

Syntax

The syntax for each variation of the 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 cancelled.
   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
```

Notes:

1. When the alternate “to” (*devnum2*) device is offline, the system takes the ‘from’ device offline:

   ```
   IEF281I devnum1 NOW OFFLINE
   ```
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 | ON}
```

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

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
  - The console's ability to receive undelivered messages

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                        }

  {CN={(consname1)           }}
      {(consname1,consname2)}
      {(consname1,consname1)}
```

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 its alternate or 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.

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 IEE686I.

(*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. This could be a MCS or SMCS console.

Notes:

In a sysplex:

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.

Example 1

To SWITCH console *MON1* to console *MON2*, enter:

```
SWITCH CN=(MON1,MON2)
```

Example 2

To SWITCH console *SERVE* to its alternate, enter:

```
SWITCH CN=SERVE
   or
SWITCH CN=(SERVE)
```

Example 3

To remove console *COMPUT*'s attributes from the console it has been switched to, enter:

```
SWITCH CN=(COMPUT,COMPUT)
```

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 only from the master console or another 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}                                      ]
      [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|nnnM}        ]
        [,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.

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.

*nnn***K**

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 64K 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 turn master trace on if it is not on already.

*nnn***K**
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, unless you specified a different table size in a previous TRACE command or, during system initialization, in a SCHEDxx parmlib member.

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 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 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

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 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.

*nnnn***K**
*nnnn***M**
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.

*nnnn*K is the buffer size in kilobytes, where nnnn is a decimal number from 1 to 9999. *nnnn*M 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 CTncccxx 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.

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 cataloged 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 name identifies the name associated with the environment creating the work request, which may reside anywhere within the network.

The connection name 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.

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.

PROC|PR=*proc name*
specifies the proc name to be used as a filter for tracing. Proc name identiies 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.

TRAN|T=*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.

USER|U=*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=(Y|N)
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.

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

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|cc|cca|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 4-147 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.

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, cc, cca, name, or name-a*

The display area (*a*), console identifier (*cc*), name (*name*), or both (*name-a* or *cca*) 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 (unless routing instructions are in effect; see the MSGRT command in this chapter).

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

Table 4-39 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, vector facilities, expanded storage, and central storage sections, see the CONFIG command.

Table 4-39. Summary of the VARY Command

Command:	Topic:
VARY CN,ACTIVATE	"Controlling Problem Determination Mode for the System Console" on page 4-544
VARY CN,DEACTIVATE	"Controlling Problem Determination Mode for the System Console" on page 4-544
VARY CN	"VARY CN command" on page 4-545
VARY ...,CONSOLE	"VARY CONSOLE command" on page 4-551
VARY ...,MSTCONS	"Changing the Master Console" on page 4-554
VARY ...,HARDCPY	"Controlling Hardcopy Processing" on page 4-555
VARY conspec...,ONLINE VARY conspec...,OFFLINE	"Placing a Secondary Console Online or Offline" on page 4-559
VARY devspec...,AUTOSWITCH	"Defining a Tape Device as Automatically Switchable" on page 4-560
VARY devspec...,ONLINE VARY devspec...,OFFLINE	"Placing an I/O Device or a Range of I/O Devices Online or Offline" on page 4-561
VARY GRS	"Controlling a Global Resource Serialization Complex" on page 4-565
VARY PATH	"Placing an I/O Path or Paths Online or Offline" on page 4-567
VARY SMS CFCACHE VARY SMS CFVOL VARY SMS FALLBACK VARY SMS,SHCDS VARY SMS,SHCDS CFRESET VARY SMS,SMSVSAM	"Changing the State of Coupling Facility Cache Structures and Volumes" on page 4-569
VARY SMS,DRIVE VARY SMS,LIBRARY	"Placing an Optical Drive or Library Online or Offline" on page 4-571
VARY SMS,LIBRARY	"Placing a System-Managed Tape Library Online or Offline" on page 4-571
VARY SMS,PDSE,ANALYSIS	"Analyzing the State of the PDSE Subsystem" on page 4-571
VARY SMS,PDSE,FREELATCH	"Releasing PDSE Latches" on page 4-572
VARY SMS,STORGRP VARY SMS,VOLUME	"Changing the SMS Status of a Storage Group or Volume" on page 4-572

Table 4-39. Summary of the VARY Command (continued)

Command:	Topic:
VARY SWITCH	"Placing a Switch Port Online or Offline" on page 4-574
VARY WLM,APPLENV	"Controlling an Application Environment" on page 4-575
VARY WLM,POLICY	"Activating a Service Policy" on page 4-578
VARY XCF	"Removing a System from the XCF Sysplex" on page 4-579

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
- VARY MSTCONS

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 4-40. Sysplex Scope for VARY Command

Command	Conditions
VARY CN	Has sysplex scope unless all of the following are true: • You issue VARY CN(*conspec*),ONLINE without specifying SYSTEM=. • You do not specify SYSTEM= in the CONSOLxx member of Parmlib that defines this console. • The console has never been active in the sysplex.
VARY ...,MSTCONS	Has sysplex scope when you issue VARY *conname*,MSTCONS. Also has sysplex scope when you issue VARY *devnum*,MSTCONS, but only if you use a common IODF for the specified device across the sysplex.
VARY SMS, STORGRP|VOLUME	Has sysplex scope under these conditions only: • You specify (*storgrp|volume*,ALL) and all systems in the sysplex are in the same SMS complex. • You specify (*storgrp|volume*, *system*) where *system* 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 *z/OS MVS Planning: Workload Management* 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 or console ids, 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.

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

Note: If you did not specify message level for the system console in CONSOLxx, the system console receives all messages except broadcast messages.

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 id or 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.

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.

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 id or name. An asterisk (*) indicates the system console on which the command is entered. You can enter the command from any authorized console (master console control) 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.

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
- Define an alternate console
- Change a console's alternate group definition

The consoles specified on these commands must be defined as consoles in the CONSOLxx Parmlib member. Extended MCS consoles may 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]... )}
       [,ALTGRP={name|{*NONE*}]
       [,AMSCOPE=([*][,name[,name]...])]
       [,AUTH={ALL|INFO|MASTER|([SYS][,IO][,CONS])}]
       [,AROUT=(rtcode[,rtcode]...)]
       [,DMSCOPE=([*][,name[,name]...])]
       [,DROUT=(rtcode[,rtcode]...)]
       [,LOGON={OPTIONAL|REQUIRED|AUTO|DEFAULT}]
       [,LU={luname|*NONE*}]
       [,MSCOPE={(*ALL)|{([*][,name[,name]...])}]
       [,OFFLINE|,ONLINE[,SYSTEM=sysname][,FORCE]]
       [,ROUT={ALL|NONE|(rtcode[,rtcode]...)}]
       [,UD={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.
2. The consoles with MASTER authority do not replace the master console. The master console and designated UD consoles receive all the action and WTOR messages that can not be displayed on any other console. The master console also receives, at minimum, routing codes 1 and 2 (master console action messages).

CN

Change the indicated authority for the specified console id(s) or console name(s). When you are uncertain of the current console identifiers, enter a DISPLAY CONSOLES command; see “Displaying Console Status Information” on page 4-111 for further information on this command.

conspec1

Represents one of the following:

consid The 2-digit console id associated with the console device (for example, 04)

nnnnnnnn The name of the console device as specified in the CONSOLxx Parmlib member CONSOLE statement (such as TAPECNTL)

***** The console that you are currently issuing commands from.

ALTGRP=

Indicates the requested alternate console group for the console.

name

The name of the group to be set as the alternate for this console. The alternate group specifies those consoles eligible to backup this console for a console failure. You define the group using the CNGRPxx parmlib member and it can contain MCS, SMCS and extended MCS consoles. For more information about specifying console groups, see the description of the CNGRPxx parmlib member in *z/OS MVS Initialization and Tuning Reference*. The group you specify must be active. If the console you specify is an MCS or SMCS console and has had an alternate console or alternate console group defined, this command replaces the previous console definition with an alternate group definition.

Note: If you specify a subsystem console, the system will issue error message IEE312I.

NONE

The system will remove any alternate group definitions or alternate console (for MCS or SMCS consoles) for the console you specify.

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=

This command requires MASTER console authority and may be protected with a security product OPERCMDS resource class with the MVS.VARYAUTH.CN profile, which requires CONTROL authority.Table 3-4 on page 3-6 lists the commands and their associated groups.

MASTER
The specified console(s) are authorized to enter all system operator commands.

ALL
The specified console(s) are authorized to enter INFO, SYS, IO, and CONS 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.

AROUT=*rtcode*

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 (*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.

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 console authority and may be protected with a security product OPERCMDS resource class with the MVS.VARYLOGON.CN profile, which requires CONTROL authority.

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. If a different LU is specified for an active console, then message IEE045I or *NONE* is issued. If the console is not active, then any LU may be specified. This command requires MASTER console authority. It may be protected with the RACF OPERCMDS resource class with the MVS.VARYLU.CN profile, which requires CONTROL authority.

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.

*
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)

Migration Considerations
In MVS/ESA SP 5.1, VARY CN(...),ONLINE became the recommended command for MCS console activation. The VARY CONSOLE command is still supported, but is no longer being enhanced. IBM recommends using the VARY CN command instead of VARY CONSOLE.

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.

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. The master console always receives messages with routing codes 1 and 2, even when NONE is specified. See Table 3-5 on page 3-14 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.

UD=
Whether the specified console can receive undelivered messages.

Y The specified console will receive undelivered messages.

N The specified console will not receive undelivered messages. This is the default value for all consoles, except the master console.

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 processors SY3 and SY4, enter:

```
VARY CN(CON3),MSCOPE=(SY3,SY4)
```

Example 3

To add SY2 to the list of processors which will send unsolicited messages to console CON3, enter:

```
VARY CN(CON3),AMSCOPE=SY2
```

Example 4

To assign the alternate group NEWYORK to the console BETSY, enter:

```
VARY CN(BETSY),ALTGRP=NEWYORK
```

Example 5

To remove any alternate group definitions from console PETE, enter:

```
VARY CN(PETE),ALTGRP=*NONE*
```

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:

```
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 master console, 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.

Migration Considerations

In MVS/ESA SP 5.1, VARY CN(...),ONLINE became the recommended command for MCS console activation. The VARY CONSOLE command is still supported, but is no longer being enhanced. IBM recommends using the VARY CN command instead of VARY CONSOLE.

The syntax of the VARY CONSOLE command is:

```
V {conspec2|(conspec2[,conspec2]...)},CONSOLE
     [,ALTCONS=conspec2]
     [,AROUT=(rtcode[,rtcode]...)]
     [,AUTH={ALL|MASTER|INFO|([SYS][,IO][,CONS])}]
     [,DROUT=(rtcode[,rtcode]...)]
     [,ROUT={ALL|NONE|(rtcode[,rtcode]...)}]
```

Notes:

1. If you specify one device number, one AUTH=operand, or one routing code, you can omit the parentheses.
2. The consoles with MASTER authority do not replace the master console. The master console and designated UD consoles receive all the action and WTOR messages that cannot be displayed on any other console. The master console also receives, at minimum, routing codes 1 and 2 (master console action messages).

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=
The system command groups that the console is authorized to enter. This parameter is accepted only from master authority consoles. Table 3-4 on page 3-6 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. The master console always receives messages with routing codes 1 and 2, even when NONE is specified. See Table 3-5 on page 3-14 for a list of routing codes.

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.

AROUT=*rtcode*
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 (*rtcode*) or ranges of routing codes (*rtcode-rtcode*) on the same AROUT operand.

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*) or ranges of routing codes (*rtcode-rtcode*) on the same DROUT operand.

ALTCONS=*conspec2*
conspec2 is one of the following:

[/]*devnum*
The device number of the console device.

nnnnnnnn
The name of the console device.

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.

If this parameter is omitted, the alternate console is unchanged.

If the console you specify is an MCS or SMCS console and has had an alternate console or alternate console group defined, this command replaces the previous console definition with an alternate group definition.

Note: In a sysplex environment, you cannot assign an alternate console from another processor in the sysplex except by console name. When you specify a device number the system assumes the device is physically attached to the processor that issues the command.

Example 1

To make consoles with device numbers 00C and 009 active and the console with device number 01FA as the alternate console for both, enter:

```
v (00c,009),console,altcons=/01fa
```

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 master console, the tape pool, and the tape library, enter:

```
v /01fa,console,auth=(cons,sys),rout=(1,3,5)
```

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

The console, 00C, and console 009 are to become active consoles with 01F as the alternate console for both.

```
v (00c,009),console,altcons=01f
```

Example 6

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 master console, the tape pool, and the tape library.

```
V 01f,console,auth=(cons,sys),rout=(1,3,5)
```

Example 7

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 8

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)
```

Changing the Master Console

Use this form of the VARY command to switch the sysplex master console function to another MCS or SMCS console. If you are not sure of the console assignments currently in effect, enter the DISPLAY CONSOLES command (see “Displaying Console Status Information” on page 4-111.)

You can issue the VARY MSTCONS command for a secondary console to establish a master console if you need operator intervention to complete RACF initialization. Prior to RACF initialization, even if your installation requires operators to log on before issuing commands, the VARY MSTCONS command is accepted from a secondary console to establish a master console.

The syntax of the VARY ,MSTCONS command is:

```
V {[/]devnum},MSTCONS
  {conname  }
```

[/]*devnum*
: The device number of the console device the systems is to assign as the master console. This device must be a full capability console. 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.

conname
The name of the console device the system is to assign as the master console. This device must be a full capability console. This can be a MCS or SMCS console.

MSTCONS
The system is to switch the master console.

Example 1

To assign the console device with device number 3E1 as the new master console, enter:

```
V 3E1,MSTCONS
```

Example 2

To assign the console device with device number ABCD as the new master console, enter:

```
V /ABCD,MSTCONS
```

Example 3

To assign the console name CON1 as the new master console, enter:

```
V CON1,MSTCONS
```

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 a SYSLOG or MCS printer console to be the hardcopy medium
- Stop the hardcopy medium.
- Activate or deactivate the operations log.

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, and the DEVNUM parameter specifies the hardcopy medium and/or the operations log.

The syntax of the VARY HARDCPY command is:

```
V [[/]devnum],HARDCPY[,CMDS|,NOCMDS|,STCMDS|,INCMDS]
   |conname          [,AROUT=(rtcode[,rtcode]...)]
   |SYSLOG           [,DROUT=(rtcode[,rtcode]...)]
   |OPERLOG          [,ROUT={ALL|NONE            }]
                            {(rtcode[,rtcode]...)}
                     [,UD={Y|N} ]
                     [,OFF[,UNCOND]]
```

[/]*devnum*
The device number of the device the system is to use for the hardcopy log. The device must be an active, nondisplaying console.

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.

SYSLOG
The system log is to become the hardcopy medium.

OPERLOG
The operations log is to be activated or deactivated.

When you omit the *devnum*, SYSLOG, and OPERLOG operands, the system changes what goes into the hardcopy log, rather than the hardcopy log device.

conname
The name of the console device the system is to use for the hardcopy log. The device must be an active, nondisplaying console.

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.

When OFF is specified without UNCOND, the following applies: If you specify *conname*, *dev*, or SYSLOG, you must specify the currently active hardcopy medium. If you specify OPERLOG, the operations log must be active.

If you do not specify *conname*, SYSLOG, or OPERLOG, the system defaults to the hardcopy medium (*conname* or SYSLOG) if it is active; otherwise, the system rejects the command. The system will not deactivate the operations log unless OPERLOG is specified.

If there are few consoles active and none of these consoles is a display console, either a hardcopy log or an 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 identified by *devnum* even if there is no other active hardcopy medium.

- If *devnum* is SYSLOG, and it was 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 *devnum* is either OPERLOG or a printer console, no messages are saved.

Note: If *devnum* 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 messsages 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.

UD=
Specifies whether UD messages (those with descriptor codes 1, 2, 3, 11, or 12, and WTORs) that are not received by another console are to be delivered to the system console (for those processor models that support a system console).

N The system console receives UD messages not sent to any other console.

Y The system console does not receive UD messages not sent to any other console.

Note: The system rejects the UD parameter if a system console is not installed.

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

When you enter the following command, device 01F, if it is active, becomes the hardcopy device and receives all system and operator communications:

```
V 01f,HARDCPY,ROUT=ALL
```

Example 4

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 5

To have the hardcopy message set recorded on the operations log, enter:

```
V OPERLOG,HARDCPY
```

Placing a Secondary Console Online or Offline

Use the following form of the VARY command to make a secondary console online as an I/O device or offline. This command can be used to make an SMCS console offline, however, it cannot be used to make an SMCS console online. You cannot move the master console or the hardcopy output device online or offline without first assigning these functions to other consoles, as described earlier in this command.

```
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.

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-' 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.

OFFLINE
The system is to take the specified device(s) offline.

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 (281,2822,283),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.

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.

nnnnnnnn
The name of a console device as specified in the CONSOLxx Parmlib member CONSOLE statement.

O-*nnnnnnnn*
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.

[/]*lowdevnum***-[/]***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 cancelled.
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.

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. Any devices targeted by a VARY OFFLINE command will enter and remain in the pending offline state. This will initially be identified by message IEF524I. Should allocation recovery message IEF238D remain outstanding for fifteen (15) minutes, the system will issue message IEF525E and repeat it at fifteen minute intervals until the operator either provides a response or cancels the job.
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. If a long job is running or the system is waiting for work, enter a START command specifying the deallocation procedure, DEALLOC, thereby allowing the system to take devices offline.

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 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 except the master console offline.

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 xvii) 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
```

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. Requestors 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.
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 4-574.)

```
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 may 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.

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:
- Online (either allocated or unallocated)
- 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 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 these commands is sysplex wide. For more information, see *z/OS DFSMSdfp 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 may 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 4-572.

MONDS(dsname{,dsname...}),ON|OFF

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 in 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.

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 FALLBACK 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.

FALLBACK
Is used as the last step in the disablement procedure to fall back from SMSVSAM processing. For the SMSVSAM fallback procedure, see *z/OS DFSMSdfp Storage Administration Reference*.

TERMINATESERVER
Abnormally 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.

Use this command only for specific recovery scenarios that require the SMSVSAM server to be down and not to restart automatically.

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 FORCEDELETELOCKSTRUCTURESMSVSAMYES before the command takes effect.

Use this command only in the event of a volume loss.

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 Reference* 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 Reference* for specific information about how to use this command, including the command syntax.

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 may 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]
      {                                                 } {ENABLE|E }
      {{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.

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.

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.

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.

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 unit. 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.

Note that this command affects only managed device paths. Nonmanaged paths must be varied online or offline separately.

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 4-567.

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 = 003990.0CC.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 = 003990.0CC.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* 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 may 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 *MVS Planning: Workload Management* for more information on operational considerations for application environments and the role of the VARY WLM,APPLENV=*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.

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 4-41 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 4-41. 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.

Table 4-41. Goal Mode and Compatibility Mode Actions for VARY WLM,APPLENV (continued)

Operation	Goal Mode Systems		Compatibility Mode Systems
	Automatic Control: JCL Procedure Defined to WLM	**Manual Control: No JCL Procedure Defined to WLM**	
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.

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.

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.

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
```

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.

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
```

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.

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
```

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 it 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 Volume 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.

© Copyright IBM Corp. 1988, 2002

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.

© Copyright IBM Corp. 1988, 2002

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
- DFSMSdfp
- ES/3090
- ES/9000
- ESCON
- Hiperspace
- IBM
- IBMLink
- IMS
- MVS
- MVS/ESA
- OS/390
- PR/SM
- RACF
- Resource Link
- RETAIN
- RMF
- SecureWay
- SOM
- SOMobjects
- SP
- Sysplex Timer
- System Object Model
- VTAM
- z/Architecture
- z/OS
- zSeries
- 3090
- 400

UNIX is a registered trademark of The Open Group in the United States and other countries.

NetView is a trademark of International Business Machines Corporation or of Tivoli Systems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Index

Special Characters

* (asterisk)
  specified on identifier of system command 4-284
  specified on jobname.identifier of system command 4-21, 4-304, 4-306, 4-517
  specified on parameter of system command 1-17, 4-153, 4-286, 4-287, 4-288, 4-289, 4-290, 4-311, 4-313

Numerics

2250-3 display unit
  program function key (PFK)
    displaying number 4-61
3277-2 display station
  back-tab key 2-17
  in message stream mode 3-11
    illustration 3-11
  PFK
    displaying number 4-61
  selector pen 2-12
3490 tape device 4-561
3990-3
  SETSMS command
    DINTERVAL parameter 4-408

A

A (RELEASE command) 4-2
accessibility A-1
ACDS parameter
  SETSMS command 4-406
ACR (alternate CPU recovery)
  machine check 4-279
action
  action for a SLIP trap 4-441
action message
  backup 3-10
  deleting
    eventual action 4-59
    immediate action 4-59
    retained immediate 4-59
  displaying information 3-17
  requeued 2-19
  retained
    deleting 4-59
  retained by action message retention facility 3-13, 3-17
  retention facility 4-59
    activating 4-64
    cancelling 4-64
    changing 4-64
    deactivating 4-64
    deleting 4-60
    displaying status 4-64
action message retention facility
  cancelling 3-17
action message retention facility *(continued)*
  deactivating 3-17
  description 3-17
  displaying action messages not retained 3-18
  retrieving an action message 3-17
ACTION parameter
  SLIP command 4-455
ACTIVATE command 4-15
  example 4-18
  specifying FORCE 4-17
  summary table 4-2
  syntax 4-15
ACTIVATE parameter
  SETSSI command 4-413
activating a workload management service policy
  controlling 1-20
activation
  action message retention facility 3-17
  general WTO installation exit IEAVMXIT 3-18
ADD dump mode 4-25
ADD parameter
  SETSSI command 4-412
address
  direct 4-435
  indirect 4-435
  wraparound 4-481
address compare in SLIP
  definition 4-467
address space
  qualifying direct or indirect address 4-438
Advanced Program-to-Program Communication/MVS (APPC/MVS)
  address space information
    changing 4-355
AFTER parameter
  in PROGxx parmlib member 4-398
alarm
  no-consoles condition 2-18
alternate console
  assigning 3-7, 4-545
  definition 2-2
  switch 2-17
alternate console group
  selecting a master console 3-9
alternate nucleus
  load 1-5
alternate path 4-51, 4-563
APF (authorized program facility)
  listing authorized libraries 4-361
APF list
  updating 4-390
apostrophe
  use 4-1
ARM (automatic resource management)
  switching couple data containing ARM information 4-417
ASID|A on DISPLAY command 4-172

© Copyright IBM Corp. 1988, 2002

assign
command group 4-547, 4-552
MCS (multiple support console) console 4-545
routing code 4-547, 4-552
assignment
alternate console 3-7
master console 3-7
message level 3-14
new master console 3-8
routing code 3-14
ATBOTTOM parameter
in PROGxx parmlib member 4-397
ATTOP parameter
in PROGxx parmlib member 4-397
audible alarm
description 2-3
authorization
changing console 3-7
authorization to system commands 3-5
automatic
AVR (automatic volume recognition) 1-32
automatic message deletion
assigning 4-74
automatic restart 1-23
automatic restart management
DISPLAY XCF command 4-230
automatically switchable tape devices
change the AUTOSWITCH attribute 4-560
definition 1-27
obtain the status of 1-29, 4-219
operational considerations 1-27
automation
effects of altering console attributes 3-1
auxiliary storage
adding 4-322
deleting 4-324
draining 4-324
replacing 4-324
AVR (automatic volume recognition) 1-32

B

base address 4-436
base/displacement shorthand in SLIP command 4-436
binary compare in SLIP
definition 4-466
blank screen
error response 2-15
response 2-17
BMFTIME parameter
SETSMS command 4-408
box a device 1-33
BPXPRMxx parmlib member
changing, with SETOMVS 4-380
branch
SLIP trap 4-447
BRL on DISPLAY command 4-173
broadcast
message
level 4-80
routing 4-80
broadcast data set 1-21
broadcast message
description 3-14
buffer shortage 2-19, 2-20, 2-22

C

C command
See CANCEL command
cache
subsystem 4-322
CACHETIME parameter
SETSMS command 4-408
cancel
job 4-19
status display for an inactive console 2-16
CANCEL action
how to perform 2-8
perform 2-7, 2-11, 2-13, 2-15
CANCEL command 1-22, 4-19
example 4-22
inactive console 2-16
summary table 4-2
syntax 4-20
CAPS|C on DISPLAY command 4-173
catalog address space 4-283
information
displaying 4-295
service
requesting 4-295
CD command
See CHNGDUMP command
central storage
job 1-23
moving online or offline 4-47
specifying configuration 1-2
CF command
See CONFIG command
CFRM (coupling facility resource managment)
switching couple data containing CFRM
information 4-417
change
dump mode and option 4-26
status
of secondary console 4-559
system commands a console can receive 4-545
change information in the entry area 2-13
changing authorization
console 3-7
channel path
placing online or offline 4-49
reconfiguring 4-49
characters
valid command 4-1
checkpoint/restart 1-23, 1-24
effect on SLIP PER trap 4-444
CHNGDUMP command 4-25
DEL operand
syntax 4-26
RESET operand
syntax 4-32

CHNGDUMP command *(continued)*
scope in a sysplex 4-26
SET operand
syntax 4-34
summary table 4-2
class C1 commands 1-36
class C2 commands 1-36
class M1 commands 1-34
class M2 commands 1-34
clock
setting 4-330
CLOCKxx member of SYS1.PARMLIB
OPERATOR PROMPT parameter 1-8
CLOCKxx parmlib member
OPERATOR PROMPT parameter 4-331
CLPA (create link pack area)
option 4-322
COFDLFxx member
keyword
MAXEXPB 4-302
PCTRETB 4-302
color
description 2-3
color options
displayed 3-18
command
check PFK command 2-10
enter
using PFK (program function key) 2-9
using selector pen 2-12
enter with selector pen 2-13
entry error correction 2-8
formats 4-13
how to enter
using keyboard 2-7
MODIFY
example 4-68
routing 4-340
select for hardcopy message set 3-23
syntax 4-13
command authority 3-5
command characters
valid 4-1
command delimiter 2-8
command entry 2-8
command entry error correction 2-8
command example
CONTROL A 4-59
CONTROL C,A 4-60
CONTROL C,D 4-61
CONTROL C,E 4-60
CONTROL C,I 4-60
CONTROL D 4-62
CONTROL M,AMRF 3-17, 4-64
CONTROL M,LOGLIM 4-65
CONTROL M,MLIM 4-66
CONTROL M,RLIM 4-66
CONTROL M,UEXIT 3-18, 4-68
CONTROL N,PFK 3-20
CONTROL Q 4-74
CONTROL S 3-16, 4-77
command example *(continued)*
CONTROL V,LEVEL 3-16
CONTROL V,USE 3-12
DEVSERV P 4-92
DEVSERV SMS 4-92
DISPLAY 4-146, 4-147, 4-182, 4-183, 4-184, 4-222, 4-223
DISPLAY PFK 3-19
DUMP 4-248
FORCE 1-26, 4-267
IEE094D 4-249
LOG 4-273
LOGOFF 4-274
MODE 4-280, 4-282
MODIFY 4-287, 4-288, 4-289, 4-294, 4-305, 4-308, 4-309, 4-310, 4-313
MODIFY CATALOG 4-301
MONITOR 4-316
MOUNT 4-318
MSGRT 4-320, 4-321
PAGEADD 4-323
PAGEDEL 4-325
REPLY 4-329
RESET 4-338
ROUTE 4-345
SEND 4-349, 4-352, 4-354
SET 3-20, 4-363, 4-365
SETDMN 4-367
SETETR 4-368
SETGRS 4-370
SETIOS 4-374
SETLOAD 4-376
SETPROG EXIT 4-392, 4-395
SETPROG LNKLST 4-399
SETSMF 4-404
START 4-505, 4-509, 4-515
STOP 4-518
STOPMN 4-522
STOPTR 4-523
SWAP 4-525
TRACE 4-534
TRACK 4-540
VARY 3-8, 3-9, 3-14, 3-22, 3-24, 4-554, 4-555, 4-560, 4-569, 4-577, 4-578, 4-579
VARY HARDCPY 3-23, 4-558
WRITELOG 4-580
command flooding 1-34
command group 3-5
assigning 4-547, 4-552
figure showing 3-7
summary 3-6
COMMNDxx parmlib member 4-434
common storage tracking
SET command 4-362
communication
console 1-10
operator-to-operator 4-348
terminal user 4-350
time-sharing user 4-349
component trace 1-27, 4-531

component trace *(continued)*
option
replying 4-331
specifying 4-331
CONFIG command 4-45
example 4-52, 4-54, 4-55
MSGRT command routing 4-319
summary table 4-2
configuration
changing 1-23
CONS command authority 3-5
console
activity
lack 2-15
activity, lack 2-15
alternate console group 2-2
changing authorization 3-7
console group 3-8
console use 3-10
control command 3-7
description 2-2
device 2-1
error condition 2-15
failure 3-8
hardware at IPL 1-2
hardware error 2-16
inactivity
cause 2-15
response 2-16
MCS (multiple console support)
definition 2-2
use 2-2
MCS at IPL 1-2
NIP at IPL 1-2
operation 2-1
recovery 2-15, 2-17, 2-18
response to an inactive 2-15
switching 3-8
varying offline 2-23
console area
L= operand 2-5
console assignment
making 3-5
console attributes
alteration and effects on automation 3-1
switching with SWITCH CN command 3-9
console characteristics
changing 3-1
defining 3-1
displaying 3-4
console group 3-8
arrangement 3-8
assigning 3-8
illustration of chain 3-8
selection 3-8
console in error condition 2-15
console inactivity
cause 2-15
console message backup
response 2-19
console operating mode
definition 3-10
setting up message stream console 3-10
setting up status display console 3-10
console operation
full-capability mode
changing 4-79
displaying 4-79
output-only mode
changing 4-79
displaying 4-79
console operation mode
effect on display screen 2-4
console screen
information
removing 4-62
console specification
altering 4-74
changing 4-74
current 4-74
displaying 4-74
console-to-console
status display
routing 4-319
sending 4-319
CONSOLxx member of Parmlib
CMDLEVEL parameter 4-555
DEVNUM parameter 4-555
ROUTCODE parameter 4-555
CONSOLxx member of SYS1.PARMLIB
ALTERNATE parameter 3-1
AMRF parameter 3-2
example 3-17
AREA parameter 3-1
AUTH parameter 3-1
CMDELIM parameter 3-2
CMDLEVEL parameter 3-3
CON parameter 3-1
DEL parameter 3-1
DEVNUM parameter 3-3
example 3-9
general description 2-2
HARDCOPY statement 3-3
INIT statement 3-2
LEVEL parameter 3-1
example 3-15
MFORM parameter 3-1
MLIM parameter 2-19, 3-2
MONITOR parameter on the CONSOLE statement 3-1
MONITOR parameter on the INIT statement 3-2
MPF parameter 3-2
MSGRT parameter 3-1
PFK (program function key) parameter 3-2
PFKTAB parameter 3-1, 3-21
relationship with CONTROL command 3-1
RLIM parameter 2-19, 3-2
RNUM parameter 3-1
ROUTCODE parameter 3-1, 3-3
RTME parameter 3-1
SEG parameter 3-1

CONSOLxx member of SYS1.PARMLIB *(continued)*
UD parameter 3-3
UEXIT parameter 3-2
USE parameter 3-1
UTME parameter 3-1
CONSOLxx parmlib member
MLIM parameter 2-22
MSGRT parameter 4-319
RLIM parameter 2-22
contents compare in SLIP
definition 4-467
control
message format 2-6, 3-16
message processed by message processing facility (MPF) 3-16
status displays on a status display console 3-10
system message 3-13
CONTROL command 4-57
A
example 4-59
syntax 4-58
C,A
example 4-60
syntax 4-59
C,CE
syntax 4-59
C,D
example 4-61
syntax 4-60
C,E
example 4-60
C,I
example 4-60
syntax 4-59
D
example 4-62
syntax 4-61
duplicate line
K Q 4-74
E
example 4-63
syntax 4-62
example 2-12
extended MCS console
limitation 4-57
M,AMRF
example 3-17, 4-64
syntax 4-64
M,LOGLIM
example 4-65
syntax 4-65
M,MLIM
example 4-66
syntax 4-65
M,RLIM
example 4-66
M,UEXIT
example 3-18, 4-68
syntax 4-68
MSGRT command routing 4-319
CONTROL command *(continued)*
N,PFK
example 3-20, 4-72
syntax 4-71, 4-72
Q
example 4-74
syntax 4-73
relationship to CONSOLE statement in CONSOLxx 3-1
S
example 3-16, 4-77
syntax 4-74
scope in a sysplex 4-58
SMCS APPLID
SMCS 4-69
status display
framing 4-61
summary table 4-2
T
syntax 4-78
V,LEVEL
example 3-16, 4-81
syntax 4-80, 4-81
V,USE
example 3-12, 3-13
syntax 4-79
CONTROL Q command
example 2-16
CONTROL T command
example 2-16
controlling
system 1-9
conversational mode
enter command assigned to PFK 2-11
enter command with selector pen 2-13
message deletion 4-74
cancelling 4-75
PFK 4-71
PFK (program function key) 3-19, 3-20
COPYFROM parameter
in PROGxx parmlib member 4-396
coupling facility resource management
DISPLAY XCF command 4-230
critical action
message
level 4-80
routing 4-80
critical eventual action
message
level 4-80
routing 4-80
critical eventual action message
description 3-14
cross system coupling facility (XCF)
displaying information 4-225
cross-system coupling facility (XCF) 4-415
active configuration
saving 4-415
alternate data set couple
defining 4-415

cross-system coupling facility (XCF) *(continued)*
data set couple
controlling 4-415
SETXCF command 4-423
signal path
modifying 4-415
starting 4-415
stopping 4-415
transport class
defining 4-415
deleting 4-415
modifying 4-415
XCF PR/SM policy 4-423
cursor
description 2-3
move 2-7
cursor detect action
perform 2-7
CVIO (clear virtual I/O)
option 4-322

D

D command
See DISPLAY command
DAE (dump analysis and elimination)
parameter 4-355, 4-358
suppression of the dump 4-457
data lookaside facility (DLF)
starting data lookaside facility (DLF) 4-514
data removal
data from entry area 2-15
data set
clearing 4-259
dump
adding 4-254, 4-255
allocation of, automatic 4-254
clearing 4-254
deleting 4-259
disable automatic allocation 4-258
enable automatic allocation 4-258
name-pattern 4-261
date
setting 4-330, 4-356
DD (data definition) statement 1-30
DDR (dynamic device reconfiguration) 1-37, 4-524
operator-requested DDR 4-524
DEACTIVATE parameter
SETSSI command 4-413
DEALLOC keyword 4-563
deallocation procedure 4-563
dedicated tape device
definition 1-27
DEFAULT statement 3-4
deferred restart 1-23
define
master console 4-554
routing codes for console 4-550, 4-552
definition
command for PFK (program function key) 3-19, 3-20
definition *(continued)*
routing codes for a console 3-13
status display console 3-10
degradation interruption
machine check
controlling recording 4-277
recording 4-277
delete
SLIP trap 4-499
deletion
message
from message stream console 3-10
delimiter of command 2-8
device
allocation 1-30
assignment 1-31
boxing 1-33
I/O device online or offline 4-561
place online or offline 4-561
supported and not supported by DDR 1-37
device status
displaying 1-10
DEVSERV command 4-83
channel path status
displaying 4-83
complete syntax 4-83
device status
displaying 4-83
example 4-92
SMS-managed device status
displaying 4-83
summary table 4-2
use 1-10
DINTERVAL parameter
SETSMS command 4-408
direct address in indirect addressing 4-435
disability A-1
disable status
meaning 4-572
disabled loop 1-18
displacement in indirect addressing 4-435
display
action messages awaiting action 3-17
action messages not retained 3-18
active job 1-23, 1-24
color options 3-18
console characteristics 3-4
console operating mode 3-5
console specifications 3-5
console status 3-5
current system status 1-9
device status 1-10
hardcopy information 3-24
highlighting options 3-18
information about system log 3-15
intensity options 3-18
job information 1-9, 4-539
logged-on TSO/E (Time Sharing Option/Extension) users 1-21
message deletion specifications 3-5
message levels not assigned to any console 3-15

display *(continued)*
message processing facility (MPF) message processing 3-18
number of allowed WTO message buffers 3-5
number of allowed WTOR message buffers 3-5
PFK (program function key) definition 3-5
routing codes not assigned to any console 3-15
status of installation exit 3-18
system activity 1-9
time and date 2-16
DISPLAY command 4-97
ASID|A operand 4-172
BRL operand 4-173
CAPS|C operand 4-173
CONSOLES
example 3-15, 3-24
sample output 3-9
displaying active service policy 4-221
displaying general IPL information used by the system 4-146
displaying IPL information 4-146
displaying PARMLIB information 4-182
displaying RTLS information 4-199
displaying the logical parmlib setup for a system 4-182
displaying workload management mode in effect 4-221
example 4-222, 4-223
FILE|F operand 4-173
LIMITS|L operand 4-173
M 1-10
MSGRT command routing 4-319
OMVS operand 4-171
OPTIONS|O operand 4-173
PF
example 3-19
PFK (program function key)
example 2-10
PID operand 4-172
R
example 4-60
RESET operand 4-173
scope in a sysplex 4-98
summary table 4-2
SUMMARY|S operand 4-172
U
use 1-10
U operand 4-172
use 1-10
VSERVER|V operand 4-173
display console
features 2-3
DISPLAY CONSOLES command
example 2-21
sample 2-21
sample output 2-21
DISPLAY IPLINFO command
example 4-146, 4-147
DISPLAY PARMLIB command
example 4-182, 4-183, 4-184
display screen area
on console 2-4
DISPLAY SSI command
ALL parameter 4-211
DYNAMIC parameter 4-211
FUNC parameter 4-211
L parameter 4-212
LIST parameter 4-211
SSI parameter 4-211
STATUS parameter 4-212
SUBSYS parameter 4-212
syntax 4-211
DISPLAY WLM command
syntax 4-221
DISPLAY XCF command
syntax 4-225
displaying device status and allocation
the IPL volume 4-219
DLF (data lookaside facility)
changing 4-301
defined 4-513
displaying 4-302
modifying 4-302
stopping address space 4-519
DLF (data lookaside facility) address space
stopping 4-519
DLF (data lookaside facility) data object
starting 4-513
DLOG (JES3 system log) 3-24
domain description table
changing 4-366
displaying 4-118
DS command
See DEVSERV command
DSNAME parameter
in PROGxx parmlib member 4-397
dump
clearing 4-259
data set
adding 4-254, 4-255
clearing 4-254
deleting 4-259, 4-261
disable automatic allocation 4-258
enable automatic allocation 4-258
name-pattern 4-261
mode and option 4-25
option
specifying 4-237
resources
adding 4-255
deleting 4-259, 4-260
DUMP command
example 4-248
option
replying 4-332
specifying 4-332
summary table 4-2
syntax
complete 4-236
dump options list
for SYSABEND dump type 4-25

dump options list *(continued)*
for SYSMDUMP dump type 4-25
for SYSUDUMP dump type 4-25
DUMPDS command
summary table 4-2
dynamic device allocation 1-31
dynamic display
time interval
changing 4-78
displaying 4-78
dynamic exits
display 4-188
update 4-392, 4-395
dynamic LPA
managing 4-399
Dynamic LPA module information
display 4-192
dynamic parmlib concatenation
dynamic Changing
dynamic changing 4-376

E

E command
See RESET command
enable status
meaning 4-572
enablement policy
changing 4-355
display 4-186
end line
system checking 2-16
ENTER action
perform 2-7
enter command
using keyboard 2-7
entry area
blank out 2-15
change information 2-13
character insertion 2-14
description 2-4
error correction 2-14
error code 1-5
error event
trap 4-440
error message
response 2-16
error recovery 1-30
in SMF 1-26
switch data set 1-26
error response
blank screen 2-15
console error condition 2-15
error message 2-15
system error 2-15
eventual action
message
deleting 4-59, 4-60
eventual action message
description 3-14
example of command
CONFIG 4-52, 4-54, 4-55
CONTROL D command 2-12
CONTROL E command 2-12
DISPLAY CONSOLES 2-21
DISPLAY PFK 2-10
DISPLAY T 2-16
SEND command 4-1
TRACK 2-14
extended highlight capability
description 2-3
external interrupt key
use in master console failure 2-17
external writer
data set selection
specifying criteria 4-310
MODIFY command
data set selection 4-310
starting 4-514

F

F command
See MODIFY command
failure
console 2-19, 3-8
fetch
SLIP trap 4-447
FILE|F on DISPLAY command 4-173
flagged
message
deleting 4-63
FORCE command 1-22
caution 4-264
example 1-26, 4-267
summary table 4-2
syntax
complete 4-265
using 4-264
FORCE operand of CONFIG command
cautions about using 1-33, 4-50
FORCE operand of VARY command
cautions about using 1-33
cautions about using FORCE with OFFLINE 4-564
foreign host
definition 1-28
full-capability console
console 3-11
defined 3-11
full-capability mode
cursor on console 2-3
display screen for, illustration 2-4
keyboard on console 2-3
operation on display console 2-7

G

G command
See SWAP command
general trace service 1-27

GETMAIN/FREEMAIN/STORAGE tracing
SET command 4-362
global resource serialization
removing a system from complex 4-565
VARY GRS command 4-565
global resource serialization (GRS)
resource name list
changing 4-355
GMT (Greenwich mean time) 4-330
GRS (global resource serialization)
resource name list
selecting 4-359
GTF (generalized trace facility) 1-26
starting 4-507

H

halt
TRACK display 4-523
HALT command 1-38, 4-269
complete syntax 4-269
summary table 4-2
hard interruption
machine check
controlling recording 4-277
monitoring 4-277
hardcopy medium
bypassing a device 2-23
changing status 4-556
controlling in CONSOLxx 3-3
device, bypassing 2-23
displaying information 3-24
message 4-80
type of system-assigned 3-24
hardcopy message set
minimum routing codes for 4-557
selecting a command 3-23
selecting messages 3-22
hardcopy processing
controlling 4-555
defaults used for 3-4
varying 4-555
hardware configuration definition (HCD)
defining a device 2-23
hardware console
definition 1-2
to process MVS message traffic 2-22
hardware malfunction 1-32
HCD (hardware configuration definition)
device definition 2-1
highlighting of message 4-170
displaying options in effect 3-18
hot I/O detection 1-32

I

I command
See SWITCH command
I/O configuration
loading 1-5
I/O control command 3-7
I/O device
place online or offline 4-561
IEA101A message 1-7
IEA116A message 4-334
IEA347A message 1-7
IEA886A message 4-330
IEA888A message 4-331
IEA903A message 4-331
IEACMDxx parmlib member 4-434
IEASLPxx parmlib member 4-434
IEASYSxx member of SYS1.PARMLIB 1-7
IEAVMXIT installation exit
description 3-18
displaying status 3-18
status
activating 4-68, 4-74
changing 4-68, 4-74
displaying 4-68, 4-74
IEE094D message 4-332
IEE100E message 4-50
IEE131D message 4-50, 4-51
IEE250I message 4-82
IEE360I message 1-26
IEE507D message 4-50, 4-51
IEE508E message 4-51
IEE828E message 3-15
IEE922I message 4-79
IEE980I message 1-26
IEF225D message 1-23
IEF238D message 1-33
IEFAUTOS structure
on coupling facility 1-28
IFASMFDP dump program 1-26
immediate action
message
deleting 4-59
level 4-80
routing 4-80
immediate action message
deleting 4-60
IMSI 1-4
inactive console
characteristic 2-15
response 2-15
indirect address 4-435
indirect addressing
DATA parameter 4-437
LIST parameter 4-437
RANGE parameter 4-437
REFAFTER parameter 4-437
REFBEFOR parameter 4-437
SUMLIST parameter 4-437
TRDATA parameter 4-437
indirection indicator in indirect addressing 4-435
INFO command authority 3-5
informational
message
level 4-80
routing 4-80
informational command 3-7

informational message
description 3-14
initialization message suppression indicator (IMSI) 1-4
initialization of system software 1-7
inline commands 1-36
installation exit routine
description of use in processing a message 3-18
instruction fetch
SLIP trap 4-447
instruction line
description 2-4
Integrated Cryptographic Feature (ICRF)
bringing online 4-46
taking offline 4-46
Integrated Cryptographic Service Facility/MVS (ICSF/MVS)
bringing online 4-46
taking offline 4-46
intensity
description 2-3
interception
of system event 4-433
interruption
machine check
controlling recording 4-277
monitoring 4-277
INTERVAL parameter
SETSMS command 4-408
IO command authority 3-5
IOACTION command
summary table 4-2
IOS,MIH command options and functions
ASM 4-145
IPL (initial program loader) 1-5
procedure 1-5
program 1-5
IPS (installation performance specification)
changing 1-18, 4-357
IRA200I message 4-322
IRA201I message 4-322
ISG334I 4-370
IXL126I message 4-51
IXL127I message 4-52

J

JES event trace facilities 1-25
JES2
message routing code 3-13
procedure for starting 1-9
JES3
message routing code 3-13
procedure for starting 1-9
using MCS console 2-1
job
cancelling 1-22
controlling 1-21
displaying active 1-23, 1-24
passing information 1-22, 4-283, 4-285
restart 1-22
start 1-22
job *(continued)*
stop 1-22

K

K command
See CONTROL command
keyboard A-1
keyboard action 2-7

L

L command
See LOG command
L= operand
L= description 2-5
LIBRARY command 4-272
summary table 4-2
library lookaside
displaying information 4-158
library lookaside (LLA)
defined 4-509
starting LLA address space 4-509
LIMITS|L on DISPLAY command 4-173
LIMMSG parameter in BPXPRMxx 4-383
link pack area expansion 1-24
LLA (library lookaside)
directory
modifying 4-303
stopping address space 4-519
LLA (LNKLST lookaside)
LNKLST directory
modifying 4-68
LLA address space
starting 4-509
stopping 4-519
LNKLST concatenation
display 4-191
LNKLST directory
modifying 4-68
LNKLST set
activating 4-398
locating a routine 4-398
load
alternate nucleus 1-5
I/O configuration 1-5
IML (initial microprogram) 1-2
parameter field 1-5
process 1-5
load function 1-5
LOAD parameter 1-7
locked keyboard
response 2-17
LOG command 3-24, 4-273
complete syntax 4-273
example 4-273
summary table 4-2
logger
See system logger
LOGOFF command
complete syntax 4-274

LOGOFF command *(continued)*
example 4-274
summary table 4-2
LOGON command 1-6
complete syntax 4-275
LOGON panel 4-275
summary table 4-2
logrec
action for a SLIP trap 4-441
recording by SLIP trap 4-457
loop
disabled 1-18
lowercase character
use 4-1
LU (logical unit)
NOSCHED LU 4-104

M

M command
See MOUNT command
machine check
controlling status
monitoring 4-281
recording 4-281
degradation interruption
controlling recording 4-277
displaying status
monitoring 4-282
recording 4-282
hard interruption
controlling recording 4-277, 4-278
monitoring 4-277, 4-278
interruption
controlling recording 4-277, 4-281
monitoring 4-277, 4-281
monitoring status
controlling recording 4-280
system recovery
controlling recording 4-277
master catalog
specifying an alternate 1-7
MASTER command authority 3-5
master console 2-2
assigning 3-7
bypassing 2-23
changing assignment to another console 4-555
description 2-2
failure recovery 2-17
reassigning 3-8
routing a message 3-14
selecting from an alternate console group 3-9
sending a message 3-14
switching assignment to another console 3-8
vary 4-554
master console failure 2-17
master console-only command 3-7
master trace 1-27, 4-530
match
for SLIP trap 4-440
MAXCFFEATURELEVEL parameter
SETSMS command 4-407
MCS (multiple console support)
consoles at IPL 1-2
MCS (multiple console support) console
assigning 3-5
use 2-2
MCS console
at IPL 1-2
attribute definition 2-2
defining in CONSOLxx member of SYS1.PARMLIB 2-2
subsystems' use 2-1
message
automatic deletion
assigning 4-74
backup
displaying 4-73
controlling format 3-16
deleting 4-59
SYS1.BRODCAST notice section 4-354
error response 2-15, 2-16
eventual action
deleting 4-60
flagged
deleting 4-63
format 2-6
MPF-processed
controlling 4-355
MPF-suppressed 4-355
not requeued 2-19
number
controlling 4-61
deleting 4-62
operator-to-operator 4-348
processed by an installation exit 3-18
processed by IEAVMXIT 3-18, 4-68
processed by message processing facility (MPF) 3-16
queue
rerouting 4-73
recovery 2-19
roll rate
setting 4-74
routing
broadcast 4-80
critical action 4-80
critical eventual action 4-80
descriptor code 4-80
immediate action 4-80
informational 4-80
instruction 4-319
message level 4-80
system failure 4-80
write to operator 4-80
SYS1.BRODCAST
saving 4-352
terminal user 4-350
time-sharing user 4-349
message area
description 2-4

message buffer shortage 2-21
message control 2-16
message deletion
automatic mode
assigning 4-74
defining 4-74
conversational mode 4-74
nonconversational mode 4-74
roll mode
assigning 4-74
defining 4-74
roll-deletable mode
assigning 4-74
defining 4-74
specification
defining 4-74
displaying 4-74
verifying 4-74
message display
changing
color 4-358
highlighting 4-358
intensity 4-358
color
changing 4-358
control
MPFLSTxx member 4-358
highlighting
changing 4-358
message format
changing 3-16
controlling 2-6, 3-16
message identifier 2-6
message level
assigning 4-81
assigning to a console 3-14
console
assigning 4-80
displaying message levels not assigned to any console 3-15
message number
deleting 4-62
message presentation
controlled by MPFLSTxx member 3-16
displaying options in effect 3-18
message processing facility (MPF) 4-170
changing 4-358
controlling message presentation 3-16
description 3-16
displaying status 3-18
ending message suppression 4-358
message queue
rerouting 4-73
message roll rate
setting 4-74
message routing
cancelling 4-321
stopping 4-321
message routing code 3-13
message stream console
defined 3-10
message stream mode
deleting messages from a console 3-10
display screen for, illustration 2-4
message traffic
lack 2-15, 2-18
MIH
MIH time options and functions 4-142
MIH (missing interruption handler)
time interval
dynamic changing 4-371
missing interruption handler (MIH)
time interval
changing 4-355
MMS (MVS message service)
allocated resource
freeing 4-359
processing
ending 4-359
MN command
See MONITOR command
MODE command
example 4-280, 4-282
machine check
interruption 4-277
machine check interruption
controlling recording 4-281
summary table 4-2
modify
SLIP trap 4-498
MODIFY CATALOG command
example 4-301
MODIFY command 1-21, 4-283
BPXOINIT 4-294
catalog address space
information 4-295
service 4-295
changing WLM modes
compatibility mode 4-308
goal mode 4-308
DLF
changing 4-301
DLF parmlib member
replacing 4-302
DLF status
displaying 4-302
example 4-68, 4-287, 4-288, 4-289, 4-309, 4-310, 4-313
external writer
causing pause 4-312
specifying data set selection 4-310
fork() service
restart 4-290, 4-307
shut down 4-290, 4-307
jobname,APPL 4-287
LLA directory
replacing 4-303
network file system server 4-304, 4-305
OAM
managing 4-307
OE Threads
terminate 4-290, 4-307

MODIFY command *(continued)*
OMVS 4-308
pass information
z/OS UNIX System Services application 4-287
passing information to a job 1-22
passing information to a job or started task 4-283, 4-285
summary table 4-2
TSO/TCAM time sharing
starting 4-288
stopping 4-288
TSO/VTAM time sharing
modifying 4-289
z/OS UNIX initiators
shut down 4-290, 4-307
MONITOR command
complete syntax 4-315
example 4-316
MSGRT command routing 4-319
scope in a sysplex 4-315
summary table 4-2
to track terminal user 1-21
tracking a job 1-9
MONITOR command> 4-315
MONITOR displays at console failure 3-8
MOUNT attribute 1-31
MOUNT command 1-31, 1-32
complete syntax 4-317
example 4-318
scope in a sysplex 4-317
summary table 4-2
MOUNT command> 4-317
mount tape drive
with a partitionable processor 4-49
move
online or offline
central storage sections 4-45
channel path 4-45, 4-50
processor 4-45
secondary console 4-559
Vector Facility (VF) 4-45
MPFLSTxx member of SYS1.PARMLIB 3-18
MR command
See MSGRT command
MSGRT command 4-319
command routing
CONFIG 4-319
CONTROL 4-319
MONITOR 4-319
STOPMN 4-319
STOPTR 4-319
TRACK 4-319
complete syntax 4-319
description 1-10
example 4-320
extended MCS console
limitation 4-319
message routing
cancelling 4-321
stopping 4-321
MSGRT command 4-319 *(continued)*
ROUTE *ALL and ROUTE systemgroupname commands
limitation 4-319
routing instruction
removing 4-319
status display
routing 4-319
summary table 4-2
multiple-console
configuration 2-2
support
for MCS console 2-2
multiple-line messages at console failure 3-8
MVS message service (MMS)
changing 4-359
parameter
changing 4-355
SET command 4-359
starting 4-359
stopping 4-359

N

NAME parameter
in PROGxx parmlib member 4-396
network file system server
collecting messages for debugging 4-305
collecting messages for trace 4-305
diagnosing problems 4-305
MODIFY command 4-304
NIP (nucleus initialization program) completion 2-22
NIP console 1-7
definition 1-2
no-consoles condition 2-18, 2-19
notification 2-18
recovery 2-18
selecting a master console from an alternate console group 3-9
no-match
for SLIP trap 4-440
NODUMP dump mode 4-25
nonconversational mode
enter command with selector pen 2-13
entering command assigned to PFK 2-12
message deletion 4-74
cancelling 4-75
PFK 4-71
PFK (program function key) 3-19, 3-20
nucleus
alternate
loading 1-5
identifier 1-5
primary 1-5
standard 1-5

O

OAM (object access method)
managing 4-307
starting 4-510

OAM (object access method) *(continued)*
starting OAM 4-510
starting OAM address space 4-510
OAM (Object Access Method)
stopping address space 4-520
OAM address space
stopping 4-520
offline
varying console 2-23
offline device 2-1
OLTEP (online test executive program) 1-31
online device 2-1
operator 1-6
operator information area
description 2-4
operator load function 1-5
operator-to-operator
communication 4-348
message 4-348
options for CHNGDUMP command 4-29
OPTIONS|O on DISPLAY command 4-173
out of line display area
specification
altering 4-58
changing 4-58
displaying 4-58
output-only mode
defined 3-10
OVER dump mode 4-25

P

P command
See STOP command
PA command
See PAGEADD command
page data set
adding 4-322
deleting 4-324
draining 4-324
replacing 4-324
PAGEADD command 4-322
complete syntax 4-323
example 4-323
summary table 4-2
PAGEDEL command 4-324
auxiliary storage
deleting 4-324
draining 4-324
replacing 4-324
complete syntax 4-325
example 4-325
summary table 4-2
PAGNUM system parameter 4-322
parallel access volume
boxed alias of a parallel access volume 1-33
parameter
job entry subsystem 1-2
on SLIP command 4-441
parmlib
ADYSETxx member 4-358
parmlib *(continued)*
APPC/MVS member
changing 4-355
ASCH member
changing 4-355
CNGRP member
changing 4-355
CONSOLxx member
AMRF parameter 4-64
AREA parameter 4-58
CON parameter 4-74
CONSOLE statement MONITOR parameter 4-315
DEL parameter 4-74
INIT statement MONITOR parameter 4-315
LEVEL parameter 4-80
LOGLIM parameter 4-65
MFORM parameter 4-74
MLIM parameter 4-65
MPF parameter 4-358
PFK parameter 4-358
RLIM parameter 4-65
RNUM parameter 4-74
RTME parameter 4-74
SEG parameter 4-74
UEXIT parameter 4-68
USE parameter 4-79
UTME parameter 4-78
DIAGxx member
selecting through DIAG=xx 4-362
GRSRNLxx member
selecting GRS resource name list 4-359
IEAICSxx member 4-357
IEAOPTxx member 4-357
IECIOSxx member 4-371
IGDSMSxx member
selecting through SMS=xx 4-358
MMSLSTxx member
selecting through MMS=xx 4-359
MPFLSTxx member
changing 4-358
setting 4-358
PFKTABxx member
changing 4-358
PROGxx member 4-390
selecting through PROG=xx 4-361
SMFPRMxx member 4-357, 4-404
path
varying online or offline 4-567
PD command
See PAGEDEL command
PER (program event recording) event
trap 4-440
PER trap
dynamic PER traps 4-445
monitor with checkpoint/restart 4-444
set 4-442
performance group 4-335
changing job 1-22
PFK (program function key)
assigning 4-72

PFK (program function key) *(continued)*
command
changing 4-71, 4-72
defining 4-71, 4-72
conversational mode 4-71
defining a command 3-19, 3-20
definition error 2-9
display command defined for 2-10
display line on console 2-4
display on 3277-2 display station 2-12
displaying number
2250-3 display unit 4-61
3277-2 display station 4-61
in conversational mode 3-19, 3-20
initial definition 2-9
keyboard 2-3
nonconversational mode 4-71
system installation definition 2-9
table
assigning 4-72
defining 4-71
use to enter command 2-9
PFK (program function key) parameter 3-21
PFK (program function key) table
defining 3-21
definition 3-19, 3-21
description 3-21
PFK (program function key) table 3-21
PFK display line
example 2-4
PFK table
display definition 2-10
PFKTABxx member of SYS1.PARMLIB
changing 3-21
description 3-21
example 2-11
PFK (program function key) parameter 3-21
PID on DISPLAY command 4-172
place
channel path online or offline 4-49
processor online or offline 4-46
storage online or offline 4-47
PM command
See STOPMN command
pointer in indirect addressing 4-435
PPT (program properties table)
effects of the SET SCH command 4-361
modifying 4-361
previous command
retrieve 2-7
primary nucleus 1-5
primary SMF data set 1-25
print the system log 3-23, 3-24
procedure
started task 1-24
process resource / system manager (PR/SM)
SETXCF command 4-423
XCF PR/SM policy 4-423
processor
placing online or offline 4-46
reconfiguring 4-46
product enablement policy
changing 4-355, 4-362
display 4-186
program function keyboard
description 2-3
PRSMPOLICY option
SETXCF command 4-423
PSW (program status word) 1-5, 1-18
PT command
See STOPTR command

Q

quiesce
global resource serialization system 4-566
system 1-38
QUIESCE command 1-18, 1-38
complete syntax 4-327
summary table 4-2
quiesce status
meaning 4-572

R

R command
See REPLY command
reassignment
master console 3-8
reconfigure
channel path 4-49
DDR (dynamic device reconfiguration) 1-37
I/O device 4-561
I/O path 4-567
processor 4-46
storage 4-47
record
system event 1-26
recovery
error 1-30
from inactive console condition 2-15
from master console failure 2-17
hot I/O 1-32
lost message 2-19
no-consoles condition 2-18
registered product
display 4-186
remove
channel path 4-49, 4-50
last path to allocated, TP, in-use, or reserved device 4-49, 4-50
last path to allocated, TP, in-use, reserved device, or coupling facility 4-50
last path to coupling facility 4-49
tape or DASD volume 4-541
REPLY command 4-328
complete syntax 4-329
dump option
specifying 4-237
example 4-329
scope in a sysplex 4-329
SET command 4-330

REPLY command *(continued)*
short form 4-329
specifying system symbols 4-328
summary table 4-2
to invoke automatic restart 1-23
reserve a volume 1-31
reset
dump options to initial value 4-32
dump type to ADD 4-32
system dump options list 4-26
RESET command 4-335
changing performance group 4-336
changing service class 4-336
example 4-338
forcing hung MCS console offline 4-335
quiesce problem job or address space 4-336
resume quiesced job or address space 4-336
scope in a sysplex 4-335
summary table 4-2
RESET on DISPLAY command 4-173
resource allocation 1-30
resources
dump
adding 4-255
deleting 4-259, 4-260
response
blank screen 2-17
console message backup 2-19
error message 2-16
hardware error 2-16
inactive console condition 2-15
locked keyboard 2-17
master console failure 2-17
no-consoles condition 2-18
WTO buffer shortage 2-19
WTOR buffer shortage 2-22
restart
automatic 1-23
deck 1-23
deferred 1-23
global resource serialization ring 4-566
job 1-22
SMF (system management facilities) 1-26
system 1-18
using 1-18
RESTART key 1-18
restore
screen after hardware error 2-16
restrictions
Define PC command 4-340
retained action
message
deleting 4-59
retained immediate
message
deleting 4-59
retention facility
action message
activating 4-64
cancelling 4-64
changing 4-64
retention facility *(continued)*
action message *(continued)*
deactivating 4-64
deleting 4-60
displaying status 4-64
retrieve previous command 2-7
RMAX value
effect on short form of REPLY command 4-329
roll mode
message deletion
assigning 4-74
defining 4-74
roll-deletable mode
message deletion
assigning 4-74
defining 4-74
roll-deletable mode of message deletion
on message stream console 3-10
ROUTE command
complete syntax 4-340, 4-343
example 4-345
specifying system symbols 4-342
summary table 4-2
system-to-system
directing 4-340
routing
message 3-14
by routing code 3-13, 4-545
routing code
assigning to a console 3-14, 4-545
description 3-14
displaying routing codes not assigned to any console 3-15
for hardcopy message set 4-557
routing codes 4-114
listing 3-13
RRS (resource recovery services)
starting 4-510

S

S command
See START command
SCDS parameter
SETSMS command 4-407
SCHEDxx parmlib member 4-531
scratch volume 1-31, 1-32
screen
erasing 4-62
screen area
L= operand 2-5
on console of different operating mode 2-4
screen character
description 2-6
SD command
See SETDMN command
SDUMP dump type 4-25
SE command
See SEND command
secondary console
placing online or offline 4-559

secondary SMF data set 1-25
selector pen
  defining a command 3-19
  description 2-3
  enter command 2-12
  enter command in conversational mode 2-13
  enter command in nonconversational mode 2-13
  use 2-7
send
  message to console 4-545
SEND command 1-10
  example 4-1, 4-349, 4-352, 4-354
  for time-sharing communication 1-21
  operator-to-operator
    communication 4-348
    message 4-348
  scope in a sysplex 4-348
  summary table 4-2
  terminal user
    communication 4-350
    message 4-350
  time-sharing user
    communication 4-349
    message 4-349
SEND command> 4-348
service class 4-335
  changing job 1-22
set
  dump mode and option 4-34
SET command 4-355
  complete syntax 4-356
  DIAG parameter 4-362
  example 3-16, 3-20, 4-363
  GRSRNL parameter 4-359
  MMS parameter 4-359
  PROG parameter 4-361
  relationship to CONSOLE statement in CONSOLxx 3-2
  REPLY command 4-330
  RLS_MAXCFFEATURELEVEL parameter 4-359
  scope in a sysplex 4-355
  SMS parameter 4-358
  summary table 4-2
set time and date 1-18
SETDMN command 4-366
  complete syntax 4-366
  example 4-367
  summary table 4-2
SETETR command
  complete syntax 4-368
  enabling 4-368
  example 4-368
  PORT=n 4-368
  summary table 4-2
SETGRS command
  complete syntax 4-369
  example 4-370
  MODE=STAR 4-369
  summary table 4-2
SETIOS command
  complete syntax 4-371
SETIOS command *(continued)*
  example 4-374
  summary table 4-2
SETLOAD command
  complete syntax 4-376
  example 4-376
  summary table 4-2
SETLOGRC command
  changing the recording medium 4-378
  summary table 4-2
SETOMVS command 4-380
  syntax 4-380
SETPROG command
  summary table 4-2
  syntax 4-390
SETPROG EXIT command
  example 4-392, 4-395
SETPROG LNKLST command
  example 4-399
SETRRS CANCEL command
  complete syntax 4-403
  example 4-403
SETRRS command
  summary table 4-2
SETSMF command
  complete syntax 4-404
  example 4-404
  summary table 4-2
SETSMS command
  ACDS parameter 4-406
  BMFTIME parameter 4-408
  CACHETIME parameter 4-408
  comparison with SET SMS= 4-405
  complete syntax 4-405
  DINTERVAL parameter 4-408
  INTERVAL parameter 4-408
  MAXCFFEATURELEVEL parameter 4-407
  parameter
    incorrect combination 4-406
  SCDS parameter 4-407
  scope in a sysplex 4-406
  summary table 4-2
SETSSI command
  ACTIVATE parameter 4-413
  ADD parameter 4-412
  CONSNAME parameter 4-414
  DEACTIVATE parameter 4-413
  INITPARM parameter 4-414
  INITRTN parameter 4-414
  SUBNAME parameter 4-413
  summary table 4-2
  syntax 4-412
SETXCF command
  controlling XCF 4-415
  PRSMPOLICY option 4-423
  scope in a sysplex 4-415
  summary table 4-2
  syntax 4-415
  XCF PR/SM policy 4-423
shortcut keys A-1

shorthand for base/displacement in SLIP command 4-436
SL command
See SLIP command
SLIP command 4-433
complete syntax 4-433
error event SLIP trap 4-446
IF SBT PER trap 4-447
indirect addressing 4-435
non-PER trap 4-446
parameter relationship 4-441
SA PER trap 4-448
SAS PER trap 4-448
set effective PER trap 4-442
SET example 4-495
summary table 4-2
syntax for a PER trap 4-447
SLIP trap
instruction fetch 4-447
SAS PER 4-448
storage alteration 4-448
storage alteration by STURA 4-448
successful branch 4-447
SMF (system management facilities) 4-203
SMF (system management facility)
option
changing 4-333
replying 4-333
SMS (storage management subsystem)
ACDS
replacing 4-405
active configuration
saving 4-405
BMF record 4-408
cache control unit 4-408
CF processing 4-569
changing status of storage group or volume 4-569, 4-572
new configuration
activating 4-405
PDSE subsystem 4-571
placing a system-managed tape library online or offline 4-571
Releasing PDSE latches 4-572
restarting 4-358
starting 4-358
synchronization interval
changing 4-405
trace option 4-408
varying an optical drive online or offline 4-571
varying an optical library online or offline 4-571
SOM
START 4-512
special screen character
description 2-6
SRM (system resource manager)
parameter
changing 4-355
SSI parameter
DISPLAY SSI command 4-211
standard nucleus 1-5
start
Advanced Program-to-Program Communication/MVS (APPC/MVS) 4-505
ASCH (APPC/MVS transaction scheduler) 4-506
DFSMSPKG 4-507
DLF (data lookaside facility) 4-513
external writer 4-514
GTF (generalized trace facility) 4-507
job 4-501
LLA address space 4-509
OAM (object access method) 4-510
RRS 4-510
starting 4-505, 4-506, 4-507
starting APPC/MVS address space 4-505
starting APPC/MVS transaction scheduler 4-506
starting DFSMSPKG 4-507
terminal control address space (TCAS) 4-512
TSO/Virtual Telecommunications Access Method (TSO/VTAM) 4-512
VLF (virtual lookaside facility) 4-513
START command 1-21, 1-22, 1-27, 4-501
example 4-505, 4-509, 4-512, 4-515
summary table 4-2
START RRS command
complete syntax 4-510
example 4-511
started task
coding 1-24
controlling 1-24
modifying 1-24
passing information 1-22, 4-283, 4-285
starting 1-24
starting from console 4-501
started tasks
displaying status 4-157, 4-158
status command
controlled by TRACK command 4-539
status display
at console failure 2-19
console-to-console
routing 4-319
console-to-console routing
controlling 4-319
controlled by CONFIG command 1-10, 4-45
controlled by DISPLAY command 1-9
controlled by TRACK command 1-9
controlling 3-10
halting 4-60
printing
halting 4-60
updating 4-61
status display console
defined 3-10
status display mode
display screen for, illustration 2-4
setting up a console 3-10
stop
I/O on a channel path quickly with FORCE operand of CONFIG 4-50
monitoring of terminal use 1-21
OAM address space 4-520

stop *(continued)*
system 1-38
TRACK display 4-523
VLF (virtual lookaside facility) 4-521
STOP command 1-21, 1-22, 1-27, 4-516
complete syntax 4-519
example 4-518
summary table 4-2
stop routine 1-22
STOPMN command 4-522
complete syntax 4-522
example 4-522
MSGRT command routing 4-319
scope in a sysplex 4-522
stopping monitoring of terminal use 1-21
summary table 4-2
STOPTR command 4-523
complete syntax 4-523
example 4-523
MSGRT command routing 4-319
scope in a sysplex 4-523
summary table 4-2
use 1-10
storage
placing online or offline 4-47
reconfiguring 4-47
storage group
varying status 4-573
subsystem cache 4-322
successful branch
SLIP trap 4-447
summary of system command 4-2
SUMMARY|S on DISPLAY command 4-172
SWAP command 1-37, 4-524
example 4-525
summary table 4-2
swapping device of unlike density 1-37
switch
console 2-17
SWITCH
varying online or offline 4-574
SWITCH CN command 3-9
SWITCH command 1-26, 4-526
complete syntax 4-526
scope in a sysplex 4-526
summary table 4-2
switching console attributes 3-9
switching workload management modes
controlling 1-20
symbolic group name 1-32
symbolic links
in a sysplex 4-389
symbols
in commands 1-14
syntax
command 4-13
SYS command authority 3-5
SYS1.BRODCAST 4-348
contents 1-21
message
saving 4-352
SYS1.BRODCAST *(continued)*
notice section
deleting message 4-354
listing 4-353
SYS1.BRODCAST data set
managing with the SEND operator command 4-348
SYS1.DUMPxx 4-254
SYS1.NUCLEUS 1-7
SYS1.PARMLIB
CONSOLxx member
MSGRT parameter 4-319
SYSABEND dump type 4-25
SYSLOG (system log) 2-22, 3-23, 3-24
message class 3-24
SYSMDUMP dump type 4-25
sysplex
BPXPRMxx VERSION statement 4-389
scope
CHNGDUMP command 4-26
CONTROL command 4-58
definition 1-11
DISPLAY command 4-98
MONITOR command 4-315
MOUNT command 4-317
REPLY command 4-329
RESET command 4-335
SEND command 4-348
SET command 4-355
SETSMS command 4-406
SETXCF command 4-415
STOPMN command 4-522
STOPTR command 4-523
summary of commands 1-11
SWITCH command 4-526
TRACK command 4-539
UNLOAD command 4-541
VARY command 4-543
SETOMVS AUTOMOVE paramater 4-381
SETOMVS FILESYS paramater 4-381
SETOMVS FILESYSTEM paramater 4-381
SETOMVS FROMSYS parameter 4-381
SETOMVS MOUNTPOINT parameter 4-382
SETOMVS SYSNAME parameter 4-388
sysplex failure management
DISPLAY XCF command 4-230
switching couple data containing sysplex failure management information 4-417
sysplex scope
CHNGDUMP command 4-26
CONTROL command 4-58
definition 1-11
DISPLAY command 4-98
MONITOR command 4-315
MOUNT command 4-317
REPLY command 4-329
RESET command 4-335
SEND command 4-348
SET command 4-355
SETSMS command 4-406
SETXCF command 4-415
STOPMN command 4-522

sysplex scope *(continued)*
STOPTR command 4-523
summary of commands 1-11
SWITCH command 4-526
TRACK command 4-539
UNLOAD command 4-541
VARY command 4-543
system
control command 3-7
controlling 1-9
dump options list 4-25
error condition 2-15
information recording
controlling 1-25
IODF (I/O definition file) 1-3
IPL volume 1-3
monitoring 1-30
operator frame 1-18
record activity 3-22
recovery
controlling recording 4-281
machine check interruption 4-281
recovery processing 2-22
residence volume 1-3, 1-5, 1-38
response, lack 2-15
restart 1-18
stop 1-38
trace 1-26
system command group 3-7
assigning 4-547, 4-552
system console
definition 1-2
system control frame 1-3
system failure
message
level 4-80
routing 4-80
system information request
replying 4-329
responding 4-329
system log
control 3-22
displaying information 3-15
message appearing 3-15
printing 4-580
purpose 3-23
queuing 4-580
starting 4-580
stopping 4-580
system logger
DISPLAY XCF command 4-230
displaying information 4-160
must be active for START RRS command 4-510
switching couple data containing system logger information 4-417
system management facilities (SMF)
description 1-25
system message
controlling 3-13
system parameter 1-18
entering 4-333
system parameter *(continued)*
replying 4-333
specifying 4-333
system parameter change 1-18
system parameters 1-7
system recovery
WTOR message
replying 4-330
system request
WTOR message
replying 4-330
System Security WTORs
replying 4-330
system symbols
in commands 1-14
system task
starting from console 4-501
system trace 4-530
system-assigned hardcopy medium 3-24
system-initiated DDR 4-525
switching a data set 4-526
system-to-system
ROUTE command
directing 4-340
SYSUDUMP dump type 4-25

T

T command
See SET command
tape
blank 1-32
label 1-32
label verification 4-318
unlabeled 1-32
tape device
define automatically switchable 4-560
in a sysplex 1-27
tape drive
mounting with a partitionable processor 4-49
tapemark 1-32
TCAS (terminal control address space) 4-289
Telecommunications Access Method (TCAM) 4-288
terminal control address space (TCAS)
start 4-512
terminal user
communication 4-350
message 4-350
terminate
job 4-19
MOUNT command 4-19
time-sharing user 4-19
z/OS UNIX process 4-19
time
setting 4-330
time zone constant 4-357
time-sharing user
communication 4-349
message 4-349
TOD (time-of-day) clock
setting 1-8, 4-330

TR command
See TRACK command
trace
action for a SLIP trap 4-441
component 1-27
master 1-27
SMS trace option 4-408
system 1-26
TRACE command 1-27
complete syntax 4-529
example 4-534
summary table 4-2
trace table 1-27
TRACK command 4-539
complete syntax 4-539
example 2-14, 4-540
MSGRT command routing 4-319
scope in a sysplex 4-539
summary table 4-2
use 1-9
trap
of system event 4-433
TSO (time sharing option) 4-288
TSO/E (time sharing option / extended) 4-288
TSO/E (Time Sharing Option/Extension) 1-21
controlling 1-21
TSO/TCAM
time sharing
starting 4-288
stopping 4-288
TSO/Virtual Telecommunications Access Method (TSO/VTAM)
start 4-512
TSO/VTAM
time sharing
modifying 4-289
TSO/VTAM (Time Sharing Option/Virtual Telecommunications Access Method)
modifying 1-21

U

U command
See UNLOAD command
U on DISPLAY command 4-172
unlabeled tape 1-32
unload
tape or DASD volume 4-541
UNLOAD command 1-31, 4-541
complete syntax 4-541
scope in a sysplex 4-541
summary table 4-2
uppercase character
use 4-1
USE attribute 1-31

V

V command
See VARY command
value compare in SLIP
definition 4-467
vary
console 4-545
console offline 2-23
devices defined as 3270 model X 2-23
global resource serialization system 4-565
I/O device 4-561
I/O path online or offline 4-567
master console 4-554
secondary console 4-559
SMS (storage group or volume) 4-569, 4-572
tape device as automatically switchable 4-560
VARY CN command
description 4-545
VARY command 1-33
ALTCONS
example 3-9
ALTGRP
example 3-9
complete syntax 4-580
CONSOLE
example 3-14
example 4-554, 4-558, 4-560, 4-569, 4-577, 4-578, 4-579
HARDCPY
example 3-22, 3-23, 3-24
limitations with extended MCS console 4-543
MSTCONS
example 3-8, 4-555
relationship to HARDCOPY statement in CONSOLxx 3-3
scope in a sysplex 4-543
SMS (storage management subsystem) 4-569, 4-572
SMS (storage management subsystem) PDSE subsystem 4-571
SMS (storage management subsystem) Releasing PDSE latches 4-572
SMS (storage management subsystem) storage group or volume 4-571
SMS placing a system-managed tape library online or offline 4-571
summary table 4-2
SWITCH 4-574
VARY command> 4-542
VARY CONSOLE command
description 4-551
rejected during no-consoles condition 2-18
VARY HARDCOPY command
UD
example 3-15
VATLSTxx parmlib member 1-31
Vector Facility (VF)
bringing online 4-46
taking offline 4-46
virtual lookaside facility (VLF)
starting virtual lookaside facility (VLF) 4-514

Virtual Telecommunications Access Method (VTAM) 4-283
TCAS
modifying 4-289
VIRTUAL=REAL (V=R) 1-23
VLF (virtual lookaside facility)
defined 4-513
starting 4-513
stopping 4-521
stopping VLF (virtual lookaside facility) 4-521
volume
mounting 1-31, 4-317
varying volume status 4-573
VSERVER|V on DISPLAY command 4-173

W

W command
See WRITELOG command
wait state
not valid 1-18
restartable 1-18, 1-32, 1-33
warning line
description 2-4
wildcard
* (asterisk)
specified on identifier of system command 4-284
using in operator commands 1-17
WLM (workload management)
DISPLAY XCF command 4-230
switching couple data containing WLM information 4-417
WRITELOG command 3-23, 4-580
complete syntax 4-580
example 4-580
summary table 4-2
to force printing of SYSLOG 3-24
WTL (write to log)
buffer 4-65
SYSLOG buffer 4-65
WTL buffer
number allowed
changing 4-65
WTL macro 3-23
WTL SYSLOG buffer
number allowed
changing 4-65
displaying 4-65
WTO (write to operator)
buffer
shortage 4-73
message
level 4-80
routing 4-80
message buffer 4-65
WTO (write to operator) buffer 2-18, 2-19
WTO (write-to-operator) buffer
shortage 2-18, 2-19
WTO installation exit
IEAVMXIT
changing 4-68
WTO installation exit *(continued)*
IEAVMXIT *(continued)*
displaying status 4-68
WTO message
buffer limit 2-19
controlled by installation exit IEAVMXIT 3-18
WTO message buffer
number allowed
changing 4-65
displaying 4-65
WTOR (write to operator with reply)
buffer 4-65
WTOR (write to operator with reply) buffer
shortage 2-19
WTOR message
buffer limit 2-19
WTOR message buffer
number allowed
changing 4-65
displaying 4-65

X

XCF PR/SM policy
SETXCF command 4-423

Z

Z command
See HALT command

Readers' Comments — We'd Like to Hear from You

z/OS
MVS System Commands

Publication No. SA22-7627-03

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

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.

Name _____ Address _____

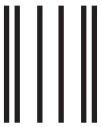
Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
SA22-7627-03

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 P384
2455 South Road
Poughkeepsie, NY
12601-5400

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Program Number: 5694-A01

Printed in U.S.A.

SA22-7627-03