

GC28-1206-5  
File No. S370-40

**Program Product**

**MVS/Extended Architecture  
Operations:  
System Commands**

**MVS/System Product:**

|                       |                 |
|-----------------------|-----------------|
| <b>JES3 Version 2</b> | <b>5665-291</b> |
| <b>JES2 Version 2</b> | <b>5740-XC6</b> |

**IBM**

### **Sixth Edition (June, 1987)**

This is a major revision of GC28-1206-4. See the Summary of Amendments following the Contents for a summary of the changes made to this manual. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This edition, with Technical Newsletter GN28-1199 applies to Version 2 Release 2.0 of MVS/System Product 5665-291 and 5740-XC6 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. The previous edition still applies to Version 2 Release 1.7 and may be ordered using the temporary order number GT00-2130. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product in this publication is not intended to state or imply that only IBM's product may be used. Any functionally equivalent product may be used instead.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Department D58, Building 921-2, PO Box 390, Poughkeepsie, NY 12602. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

## Preface

This book describes how to use MVS/Extended Architecture (MVS/XA) system commands to run a processor in System/370 Extended Architecture mode under the MVS/Extended Architecture (MVS/XA) operating system. Although you can perform many of the tasks described in this book using either MVS/XA system commands or subsystem (JES2 or JES3) commands, this book describes MVS/XA system commands only. For information about subsystem commands and their functions, see *MVS/Extended Architecture Operations: JES2 Commands* or *MVS/Extended Architecture Operations: JES3 Commands*.

This book is intended for anyone using a console and MVS/XA system commands to control an MVS/XA system. It 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 MVS/XA system.

The basic tasks of running an MVS/XA system fall under two general tasks:

1. Operating the system itself
2. Operating consoles in the system

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” describes the tasks of running an MVS/XA 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” describes the consoles that MVS supports as operators’ consoles. It describes the characteristics and operations that you **cannot** define, including the operations that are common on all operator’s consoles.
- Chapter 3: “Defining and Changing Console Characteristics” continues the console descriptions of Chapter 2 by describing the console characteristics that you **can** define. It describes the commands and SYS1.PARMLIB parameters that operators and system programmers can use to tailor the consoles and console operations to the installation’s requirements.
- Chapter 4: “MVS/XA System Commands Reference” describes, in reference form, the function, syntax, and parameters of all the MVS/XA system commands that you use to control both the system and the MCS consoles.

In the back of this book, a glossary defines technical terms used in this book.

## Associated Publications

References to other books in this manual use abbreviated titles. The complete titles of books that you should use together with this manual are:

- *Advanced Communications Function for TCAM, Version 2 Operation*, SC30-3136
- *MVS/Extended Architecture: Cache Device Administration Guide*, GC26-4137.
- *MVS/Extended Architecture: Checkpoint/Restart*, GC26-4012
- *MVS/Extended Architecture Debugging Handbook: Volume 1*, LC28-1164; *Volume 2*, LC28-1165; *Volume 3*, LC28-1166; *Volume 4*, LC28-1167; *Volume 5*, LC28-1168
- *MVS/Extended Architecture Diagnostic Techniques*, LY28-1199
- *MVS/Extended Architecture Environmental Recording Editing and Printing (EREP) Program*, GC28-1178
- *MVS/Extended Architecture Installation: System Generation*, GC26-4009
- *MVS/Extended Architecture JCL*, GC28-1148
- *MVS/Extended Architecture Message Library: System Messages, Volumes 1 and 2*, GC28-1376 and GC28-1377
- *MVS/Extended Architecture Message Library: System Codes*, GC28-1157
- *MVS/Extended Architecture Message Library: Routing and Descriptor Codes*, GC28-1194
- *MVS/Extended Architecture MVS Configuration Program: Guide and Reference*, GC28-1335
- *MVS/Extended Architecture Operations: JES2 Commands*, SC23-0064
- *MVS/Extended Architecture Operations: JES3 Commands*, SC23-0063
- *MVS/Extended Architecture Resource Measurement Facility (RMF) Monitor I and II Reference and Users Guide*, LC28-1556
- *MVS/Extended Architecture System Programming Library: Initialization and Tuning*, GC28-1149
- *MVS/Extended Architecture System Programming Library: JES2 Initialization and Tuning*, SC23-0065
- *MVS/Extended Architecture System Programming Library: JES3 Initialization and Tuning*, SC23-0059
- *MVS/Extended Architecture System Programming Library: System Modifications*, GC28-1152
- *MVS/Extended Architecture System Programming Library: Service Aids*, GC28-1159

- *MVS/Extended Architecture System Programming Library: System Macros and Facilities: Volume 1*, GC28-1150; *Volume 2*, GC28-1151
- *MVS/Extended Architecture SYS1.LOGREC Error Recording*, GC28-1162
- *MVS/Extended Architecture System Programming Library: TSO*, GC28-0629
- *MVS/Extended Architecture System Programming Library: System Management Facilities*, GC28-1153
- *MVS/Extended Architecture System Programming Library: User Exits*, GC28-1147

In addition, use the following publications depending upon your installation's specific system configuration:

- *IBM 3081 Operator's Guide for the System Console*, GC38-0034
- *IBM 3083 Operator's Guide for the System Console*, GC38-0036
- *IBM 3084 Operator's Guide for the System Console*, GC38-0037
- *OS/VS2 MVS Planning: Global Resource Serialization*, GC28-1062
- *Advanced Communications Function for VTAM Version 2 Operation*, SC27-0612
- *OS/VS Mass Storage System Extensions Operation*, SH35-0040
- *IBM 2740/2741 Communications Terminal Operator's Guide*, GA27-3001
- *IBM 3270 Information Display System 3278 Display Station Operator's Guide*, GA27-2890
- *IBM 3270 Information Display System 3279 Color Display Station Operator's Guide*, GA33-3057
- *IBM 3270 Information Display System - 3274 Control Unit Planning, Setup, and Customizing Guide*, GA27-2827
- *IBM 3270 Information Display System - 3274 Control Unit Customizing Guide*, GA23-0065
- *IBM 3270 Information Display System Character Set Reference*, GA27-2837
- *IBM 3290 Information Panel Description and Reference*, GA23-0021
- *MVS/Operator Communication Control Facility: Installation and Operation*, SC24-5226



## Contents

|  |      |
|--|------|
| <b>Chapter 1. System Operations</b>                            | 1-1  |
| Starting, Loading, and Initializing the System                 | 1-2  |
| Preparing the System Hardware                                  | 1-2  |
| Loading the System Software                                    | 1-3  |
| Initializing the System Software                               | 1-3  |
| Setting the Time and Date                                      | 1-4  |
| Starting and Specifying Parameters for the Job Entry Subsystem | 1-5  |
| Building a Global Resource Serialization Complex               | 1-5  |
| Controlling the System   | 1-6  |
| Displaying the Current System Status                           | 1-6  |
| Displaying the Status of Devices and Availability of Paths     | 1-7  |
| Communicating among Several Consoles                           | 1-7  |
| Setting the Time and Changing the System Parameters            | 1-7  |
| Using the System Restart Function                              | 1-8  |
| Controlling Time-Sharing                                       | 1-9  |
| Controlling Jobs   | 1-10 |
| Starting a Job   | 1-10 |
| Stopping a Job   | 1-10 |
| Cancelling a Job   | 1-10 |
| Modifying a Job's Parameters                                   | 1-11 |
| Modifying a Job's Performance Group                            | 1-11 |
| Restarting a Job   | 1-11 |
| Controlling an External Writer                                 | 1-13 |
| Controlling Shared DASD  | 1-14 |
| Shared DASD Operating Guidelines                               | 1-15 |
| Specifying Shared DASD Mount Characteristics                   | 1-15 |
| Guidelines for Sharing Data Sets                               | 1-16 |
| Controlling a Global Resource Serialization Complex            | 1-17 |
| Controlling System Information Recording                       | 1-19 |
| Interacting with System Functions                              | 1-22 |
| Device Allocation  | 1-22 |
| Hot I/O Detection  | 1-24 |
| Device Boxing  | 1-24 |
| Responding to Failing Devices                                  | 1-25 |
| Reconfiguring Resources  | 1-26 |
| Alternate CPU Recovery   | 1-27 |
| Partitioning a 3084 or 3090 Model 400 Processor Complex        | 1-28 |
| Rebuilding a Global Resource Serialization Ring                | 1-31 |
| After a System Failure   | 1-31 |
| After a CTC Adapter Failure                                    | 1-33 |
| Quiescing the System   | 1-36 |
| Stopping the System  | 1-36 |
| <b>Chapter 2. Console Characteristics and Operations</b>       | 2-1  |

|   |      |
|---|------|
| General Characteristics of Display Consoles                     | 2-1  |
| Subsystems' Use of MCS Consoles                                 | 2-2  |
| Devices MVS/XA Supports as Consoles                             | 2-2  |
| Multiple-Console Configuration                                  | 2-3  |
| Features on Display Consoles                                    | 2-4  |
| Display Screen Areas  | 2-6  |
| Special Screen Characters                                       | 2-8  |
| Messages Sent to Display Consoles                               | 2-8  |
| Specific Characteristics of Display Consoles                    | 2-9  |
| 3277 Model 2 Display Console                                    | 2-9  |
| 3278 and 3279 Display Consoles                                  | 2-14 |
| 3290 Information Panel  | 2-23 |
| 3180 Model 1 Display Station                                    | 2-29 |
| Characteristics of 2740 Communications Terminal                 | 2-35 |
| Entering Commands or Responding to Messages on the 2740         | 2-35 |
| Responding to a Negative System Response on the 2740            | 2-35 |
| Changing Information on the 2740                                | 2-35 |
| Receiving Messages on the 2740                                  | 2-36 |
| Operations on Display Consoles in Full-Capability Mode          | 2-36 |
| Performing Basic Keyboard Actions                               | 2-36 |
| How to Enter Commands   | 2-37 |
| Entering Commands with the Keyboard                             | 2-37 |
| Entering Commands with Program Function Keys                    | 2-38 |
| Entering Commands with the Selector Pen                         | 2-42 |
| Changing Information in the Entry Area                          | 2-43 |
| Using the Hardcopy and System Logs                              | 2-45 |
| Using the Hardcopy Log  | 2-45 |
| Using the System Log  | 2-48 |
| Handling Consoles in Error Conditions                           | 2-48 |
| System Errors   | 2-48 |
| Console Hardware Errors   | 2-49 |
| Recovering Lost Messages  | 2-50 |
| Responding to Console Message Backups                           | 2-51 |
| Responding to a Master Console Failure                          | 2-54 |
| Responding to a No-Consoles Condition                           | 2-54 |
| Processing MVS Messages at the System Console                   | 2-56 |
| <b>Chapter 3. Defining and Changing Console Characteristics</b> | 3-1  |
| Using SYS1.PARMLIB Members to Define Your Consoles              | 3-2  |
| Using CONSOLxx to Define Console Characteristics                | 3-2  |
| Using MPFLSTxx to Control Messages                              | 3-5  |
| Using PFKTABxx to Define your PFKs                              | 3-5  |
| Controlling the Hardcopy Log                                    | 3-6  |
| Assigning the Hardcopy Log to a Device                          | 3-6  |
| Routing Messages to the Hardcopy Log                            | 3-7  |
| Sending Commands and Command Responses to the Hardcopy Log      | 3-8  |
| Making Console Assignments                                      | 3-9  |
| System Command Groups   | 3-9  |
| Assigning a Master Console or Secondary Console                 | 3-11 |
| Placing a Console in Offline Status                             | 3-11 |
| Interchanging Your Consoles on a Control Unit                   | 3-12 |
| Assigning Alternate Consoles                                    | 3-12 |
| Displaying Information About Console Characteristics            | 3-13 |
| Defining the Console Operating Mode                             | 3-15 |



|   |            |
|---|------------|
| Using a Status Display Console  | 3-18       |
| Using a Message Stream Console  | 3-18       |
| Controlling System Messages   | 3-19       |
| Defining the Message Processing Facility (MPF)                                  | 3-19       |
| Controlling the Action Message Retention Facility                               | 3-21       |
| Activating WTO and WTOR User Exit Routines                                      | 3-23       |
| Checking Message Processing, Retention, and Presentation Options                | 3-24       |
| Defining Routing Codes for a Console  | 3-24       |
| Defining Message Levels for a Console   | 3-26       |
| Controlling the Format of Messages  | 3-28       |
| Defining PFKs Using PFK Tables  | 3-28       |
| An Example of Defining a PFK Table  | 3-29       |
| Defining PFKs Using the CONTROL Command   | 3-31       |
| Deleting Messages from the Console Screen                                       | 3-32       |
| Defining Conversational or Non-conversational Message Deletion Mode             | 3-32       |
| Deleting Messages Manually in Conversational Mode                               | 3-33       |
| Verifying the Deletion Request  | 3-34       |
| Deleting Messages Manually in Nonconversational Mode                            | 3-34       |
| Defining the Scope of Message Deletion  | 3-35       |
| Deleting Retained Action Messages with the CONTROL Command                      | 3-36       |
| Numbering Messages to Aid Manual Message Deletion                               | 3-37       |
| Temporarily Suspending the Screen Roll  | 3-37       |
| Defining Automatic Message Deletion   | 3-38       |
| Finding Appropriate Message Deletion Specifications                             | 3-40       |
| Checking the Message Deletion Specifications                                    | 3-41       |
| Controlling System Status Displays  | 3-42       |
| Setting Up Display Areas on a Console   | 3-42       |
| Routing Status Displays to Display Areas  | 3-46       |
| Framing Status Displays   | 3-48       |
| Erasing Status Displays   | 3-48       |
| Establishing Dynamic Status Displays  | 3-49       |
| Changing the Time Interval for Updating a Dynamic Status Display                | 3-51       |
| Displaying Jobname, Data Set Status, and TSO Information                        | 3-52       |
| Adding Information to Mount and Demount Messages                                | 3-53       |
| Managing Messages with a Console Cluster  | 3-54       |
| Setting Up and Using a Master Console Cluster                                   | 3-55       |
| Setting Up an Alternate Console Chain   | 3-56       |
| Defining Routing Codes for the Consoles   | 3-57       |
| Defining the Operating Modes and the Message Levels for the Consoles            | 3-58       |
| Setting Up Display Areas  | 3-59       |
| Setting Up a TRACK Display  | 3-61       |
| Setting Message Roll Rates and Message Deletion Specifications for the Consoles | 3-62       |
| Directing Command Responses to Specific Consoles                                | 3-62       |
| Setting Up a Periodic Display of Outstanding Requests                           | 3-63       |
| Summary of Contents of CONSOLxx for the Cluster                                 | 3-64       |
| Defining PFKs for Console 2   | 3-65       |
| Summary of the PFK Definitions for the Cluster                                  | 3-69       |
| Activating the PFK Table  | 3-69       |
| Using the Master Console Cluster and Setting It Up Again                        | 3-70       |
| The 3290 as a Console Cluster   | 3-70       |
| <b>Chapter 4. MVS/XA System Commands Reference</b>                              | <b>4-1</b> |
| Command Syntax  | 4-8        |
| Command Format  | 4-9        |

|   |       |
|---|-------|
| CANCEL Command  | 4-10  |
| CHNGDUMP Command  | 4-13  |
| Dump Options and Modes  | 4-13  |
| Changing the Dump Mode and Options  | 4-13  |
| Setting the Dump Modes and Options  | 4-15  |
| Removing Options from or Resetting the System Dump Options Lists                            | 4-17  |
| Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP Dump Types                              | 4-19  |
| Resetting the Dump Mode to ADD and the Dump Options to Initial Values                       | 4-22  |
| Example Showing How CHNGDUMP Commands Affect Dump Modes and Options                         | 4-22  |
| CONFIG Command  | 4-25  |
| Reconfiguring the System Directly   | 4-26  |
| Reconfiguring the System in Response to a Configuration Display                             | 4-32  |
| Reconfiguring the System with a CONFIGxx Member of SYS1.PARMLIB                             | 4-34  |
| CONTROL Command   | 4-35  |
| Changing Display Area Specifications  | 4-37  |
| Deleting Retained Action Messages   | 4-38  |
| Halting the Printing or the Display of a Status Display                                     | 4-39  |
| Controlling Displays in Areas   | 4-40  |
| Removing Information From the Screen  | 4-41  |
| Activating, Deactivating, or Displaying the Status of the Action Message Retention Facility | 4-42  |
| Changing or Displaying the Number of Allowed WTO and WTOR Message Buffers                   | 4-43  |
| Changing or Displaying the Status of WTO User Exit IEAVMXIT                                 | 4-44  |
| Changing a PFK Definition   | 4-45  |
| Rerouting Message Queues  | 4-47  |
| Changing or Displaying Message Deletion and Format Specifications                           | 4-48  |
| Changing or Displaying Time Intervals for Dynamic Displays                                  | 4-51  |
| Changing the Operating Mode of a Console  | 4-52  |
| Selecting the Message Levels for a Console  | 4-53  |
| DEVSERV Command   | 4-55  |
| DISPLAY Command   | 4-57  |
| Displaying Page and Swap Data Set Information   | 4-61  |
| Displaying Device Status and Allocation   | 4-63  |
| Displaying System Configuration Information   | 4-65  |
| Displaying Console Status Information   | 4-70  |
| Displaying CONTROL Command Functions  | 4-73  |
| Displaying Dump Options or Dump Data Set Status   | 4-73  |
| Displaying System Activity  | 4-76  |
| Displaying System Requests  | 4-79  |
| Displaying SMF Data   | 4-83  |
| Displaying Commands Defined for PFKs  | 4-84  |
| Displaying the Local Time and Date and the Greenwich Mean Time and Date                     | 4-85  |
| Displaying SLIP Trap Information  | 4-85  |
| Displaying the Domain Description Table   | 4-86  |
| Displaying Message Suppression, Retention, Color, Intensity, and Highlighting Options       | 4-87  |
| Displaying Global Resource Serialization Information  | 4-88  |
| DUMP Command  | 4-94  |
| Specifying Dump Options   | 4-94  |
| DUMPDS Command  | 4-98  |
| FORCE Command   | 4-102 |
| HALT Command  | 4-105 |
| LOG Command   | 4-106 |
| MODE Command  | 4-107 |

|  |       |
|--|-------|
| Displaying Recording and Monitoring Status   | 4-108 |
| Controlling the Recording of System Recovery and Degradation Machine Check Interruptions | 4-108 |
| Controlling the Recording of Hard Machine Check Interruptions                            | 4-110 |
| MODIFY Command   | 4-112 |
| Modifying Job Parameters   | 4-112 |
| Starting TSO/TCAM Time-Sharing   | 4-113 |
| Stopping TSO/TCAM Time-Sharing   | 4-113 |
| Modifying TSO/VTAM Time-Sharing  | 4-114 |
| Specifying Data Set Selection Criteria for an External Writer                            | 4-115 |
| Causing an External Writer to Pause  | 4-116 |
| Building and Replacing LNKLST Directories  | 4-117 |
| Communicating with Catalog Address Space   | 4-117 |
| MONITOR Command  | 4-119 |
| MOUNT Command  | 4-121 |
| MSGRT Command  | 4-123 |
| Stopping Message Routing   | 4-125 |
| PAGEADD Command  | 4-126 |
| QUIESCE Command  | 4-128 |
| REPLY Command  | 4-129 |
| Replying to System Information Requests  | 4-130 |
| Replying to System Requests During Recovery Processing                                   | 4-130 |
| Specifying System Parameters   | 4-131 |
| Setting the Time-of-Day Clock  | 4-132 |
| Specifying SMF Options   | 4-133 |
| Specifying Dump Options  | 4-134 |
| RESET Command  | 4-135 |
| SEND Command   | 4-136 |
| Communicating with Other Operators   | 4-137 |
| Communicating with Specified Users   | 4-138 |
| Communicating with All Terminal Users  | 4-139 |
| Saving Messages in the Broadcast Data Set  | 4-140 |
| Listing the Notices Section of the Broadcast Data Set                                    | 4-141 |
| Deleting a Message from the Notices Section of the Broadcast Data Set                    | 4-141 |
| SET Command  | 4-142 |
| SETDMN Command   | 4-145 |
| SETSMF Command   | 4-147 |
| SLIP Command   | 4-148 |
| Setting a Non-PER SLIP Trap  | 4-151 |
| Setting an Instruction Fetch or Successful Branch PER Trap                               | 4-152 |
| Setting a Storage Alteration PER Trap  | 4-153 |
| Keyword Relationships  | 4-153 |
| Keywords for SLIP SET Commands   | 4-154 |
| SLIP SET Examples  | 4-174 |
| Modifying a SLIP Trap  | 4-175 |
| Deleting a SLIP Trap   | 4-176 |
| SLIP Keyword Summary   | 4-177 |
| Indirect Addressing Used with SLIP   | 4-178 |
| START Command  | 4-183 |
| Starting a Job from the Console  | 4-183 |
| Reading a Job from a DASD Device or Tape   | 4-184 |
| Starting a Writer  | 4-184 |
| Starting GTF   | 4-185 |
| Starting TSO/VTAM Time-Sharing   | 4-187 |

|  |       |
|--|-------|
| Starting the LNKLST Lookaside (LLA) Address Space    | 4-188 |
| STOP Command   | 4-189 |
| Terminating the LNKLST Lookaside (LLA) Address Space | 4-190 |
| STOPMN Command                                       | 4-191 |
| STOPTR Command                                       | 4-192 |
| SWAP Command   | 4-193 |
| Operator-Requested DDR                               | 4-193 |
| System-Initiated DDR                                 | 4-194 |
| SWITCH Command                                       | 4-195 |
| TRACE Command  | 4-196 |
| TRACK Command  | 4-198 |
| UNLOAD Command                                       | 4-200 |
| VARY Command   | 4-201 |
| Assigning and Controlling MCS Consoles               | 4-203 |
| Changing the Master Console                          | 4-206 |
| Controlling the Hardcopy Log                         | 4-206 |
| Changing the Online or Offline Status of Resources   | 4-209 |
| WRITELOG Command                                     | 4-217 |

**Glossary** X-1

**Index** X-15

## Figures

- 1-1. Example of a Two-System Global Resource Serialization Ring 1-18
- 1-2. Identifying the Resources in a 3084 Processor Complex 1-28
- 1-3. Identifying the Resources in a 3090 Model 400 Processor Complex 1-29
- 1-4. Output of the DISPLAY M=SIDE Command 1-30
- 2-1. Devices MVS/XA Supports as Consoles 2-3
- 2-2. Available Features of Display Consoles 2-6
- 2-3. Comparison of the Display Screens of Full-Capability and Output-Only Display Consoles 2-7
- 2-4. 3277 Model 2 Display Console with Optional Features Attached 2-11
- 2-5. 3278 Model 4 Display Console with Optional Features Attached 2-16
- 2-6. 3290 Information Panel 2-24
- 2-7. 3290 Four-Part Console with Keyboard 2-26
- 2-8. 3180 Model 1 Display Station 2-30
- 2-9. Keyboard of the 3180 Model 1 2-32
- 2-10. IBM Defaults for PFKs 2-39
- 2-11. First Line of a Hardcopy Log Message 2-46
- 2-12. Hardcopy Devices that MVS/XA Assigns 2-47
- 2-13. Output of DISPLAY CONSOLES,BACKLOG Command 2-52
- 3-1. Comparison of System Commands and CONSOLE Statements in CONSOLxx 3-3
- 3-2. Comparison of System Commands and INIT Statements in CONSOLxx 3-4
- 3-3. Comparison of VARY HARDCOPY Commands and HARDCOPY Statements in CONSOLxx 3-4
- 3-4. System Command Groups 3-10
- 3-5. Sample Alternate Console Chain 3-12
- 3-6. Output of the DISPLAY CONSOLES,A Command 3-14
- 3-7. Format of the 3277 Model 2 Screen in Message Stream Mode 3-16
- 3-8. Message Routing Codes 3-25
- 3-9. PFKTAB01 Member of SYS1.PARMLIB. 3-30
- 3-10. Default Values for SEG Parameter in CONSOLxx 3-35
- 3-11. Sample Screen Format Showing Two Display Areas 3-43
- 3-12. Display Area Sizes for Consoles 3-44
- 3-13. Example of Setting Up and Using Display Areas 3-45
- 3-14. Display Areas on Consoles in the Console Cluster 3-60
- 3-15. 3290 Information Panel as a Console Cluster 3-70
- 4-1. System Command Summary 4-2
- 4-2. System Command Syntax 4-8
- 4-3. System Command Format 4-9
- 4-4. Example of How CHNGDUMP Commands Affect Dump Modes and Options 4-23
- 4-5. Summary of the DISPLAY Command 4-57
- 4-6. Displaying System Activity: Information for the LIST Operand 4-76
- 4-7. Displaying System Activity: Information for the ALL Operand 4-77
- 4-8. MODE Parameters Allowed for Machine Check Interruptions 4-107



## Summary of Amendments

**Summary of Amendments  
for GC28-1206-5  
As Updated November 16, 1987  
by Technical Newsletter GN28-1199**

This Technical Newsletter consists of changes:

- Adding support for the MVS command delimiter.
- Removing support for the JES3 action message retention facility.
- Adding a new parameter to the MODIFY CATALOG command.

The MVS command delimiter, defined at system initialization, allows the operator the option of entering multiple commands in the entry area.

Removing support for the JES3 action message retention facility is done in conjunction with JES3 changes and without any loss of function for the MVS operator. The MCS action message retention facility can process all messages, JES3 and MVS, on all systems.

The new parameter on the MODIFY CATALOG command is END. This will allow the catalog address space to end processing a specified service task while:

- using the REDRIVE option to continue to process a specified catalog request by a different service task
- or using the NOREDRIVE option to end any catalog requests in process at the time.

**Summary of Amendments  
for GC28-1206-5  
for MVS/System Product Version 2 Release 2.0**

This edition contains changes for MVS/System Product Version 2 Release 2.0. The changes include the following:

- The CLOCKxx member of SYS1.PARMLIB can be used to prompt the operator to set the time and date.
- PFKs are defined in PFK tables that reside in the PFKTABxx member of SYS1.PARMLIB. The tables are controlled through the CONSOLxx member of SYS1.PARMLIB and the CONTROL and SET commands.
- The operator can use the CONTROL command to:
  - Include the name of the system that sent the message along with each message
  - Assign the definitions in a specified PFK table to a specified console

- Change the maximum number of WTOR messages in the system
- Delete one or more outstanding critical eventual action messages
- The operator can use the DISPLAY command to display:
  - The page and swap data sets that the system is currently using
  - Information about specific SLIP traps
  - Added information about the hardcopy log
  - Outstanding messages by routing codes, by keynames, by message id, and by system
  - PFK definitions available to a console by PFK table and by console id
- Through the DUMP command, the NIP hardcopy buffer data can be dumped.
- The operator can use the MSGRT command to route the DISPLAY ASM command.
- The operator can use the SET command to:
  - Establish the PFK tables available for a console
  - Change the commands SLIP is to process
- The SLIP command has the following changes:
  - STDUMP, a new keyword to create SLIP system trace records while the trap is enabled and schedule an SVC dump when the trap is disabled or deleted
  - Changes to the DATA, PVTEP, PVTMOD, and WAIT keywords
- The operator can use the VARY command to:
  - Route messages to consoles and to the hardcopy log by ranges of routing codes
  - Add and remove specified routing codes that send messages to consoles and to the hardcopy log
- MVS no longer supports the 2250 model 3 display unit, and the 3277 model 1 and 3278 model 1 display stations as operators' consoles
- JES3 operators can use MCS consoles to run JES3

The information that was formerly in Part 2 is now in Chapters 2 and 3. Chapter 2 includes information about the consoles that IBM supports as operator consoles, and procedures for operating them. Chapter 3 includes information about how to tailor the characteristics of consoles and operations of consoles.

A summary of the parameters in the CONSOLxx member of SYS1.PARMLIB is included in Chapter 3. Additionally, throughout the book, the operands on some MVS commands are associated with the corresponding parameters in CONSOLxx.

Support for MVS/XA DFP includes a new operand on the MODIFY command, the CATALOG operand, which communicates with the catalog address space.

In addition, minor technical and editorial changes have been made throughout.



**Summary of Amendments  
for GC28-1206-4  
for MVS/System Product Version 2 Release 1.7**

This edition contains changes to support MVS/System Product Version 2 Release 1.7. The changes include:

- Description of the partitioning of the 3090 Model 400 processor complex and extended storage elements on the 3090 Model 400
- The new ESTOR operand on the CONFIG command, which allows the operator to configure extended storage elements
- New displays of configuration information in response to parameters on the DISPLAY M command

This edition contains changes to support the Vector Facility Enhancement. Support for the Vector Facility includes the following:

- The VF keyword on the CONFIG command allows the operator to reconfigure one or more Vector Facilities attached to online processors.
- The VFON and VFOFF operands on the CONFIG CPU,ONLINE command also allows the operator to reconfigure Vector Facilities.
- The system response to the DISPLAY M,CPU command changes slightly to include the status of Vector Facilities.
- The VS keyword on the MODE command allows the system to monitor the machine checks caused by a Vector Facility.

This edition also includes the description of the 3081 display station - a new operator console.

In addition, minor technical and editorial changes have been made throughout.



## Chapter 1. System Operations

The tasks of starting, running, and stopping an MVS/Extended Architecture (MVS/XA) system involve controlling the MVS/XA system software and most installation hardware, including processors, channel paths, and I/O devices. This book is for persons who need information about these tasks. They include:

- Those who need to know the different parts of the system and how to make them work best for the installation
- Those who develop procedures for the daily operations, including system programmers or 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
- Those who must predict problems and set up procedures for handling them

Controlling MVS/XA involves issuing commands on a console and responding to messages that appear on the console screen. To start the system, you use two different devices as consoles. The first device is the **hardware console** (or system console), which is connected to the processor. From this console, you load the system software and specify the nucleus. Later, during normal operations, this console is used by system engineers to monitor and service the hardware.

You probably know the second device as the **master console**. Actually, it does not become the master console until it is capable of accepting MVS/XA commands. When you first use this device to initialize the system, it is controlled by initialization programs (and is sometimes called the NIP console). You use it to give these programs certain initial values, specify an alternate master catalog, and, perhaps, set the time and date. This device can then become a master console — capable of accepting the MVS/XA commands described in this book — when MVS starts up.

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

Chapter 2 of this book describes the physical characteristics of consoles and techniques for operating the consoles. Chapter 3 describes how to define and change the characteristics of the multiple console service (MCS) consoles. Chapter 4 describes, in reference form, all the MVS/XA system commands that you can use to control both the system and MCS consoles.

The tasks of operating an MVS/XA system that are described in Chapter 1 include:

- Starting, loading, and initializing the system
- Building a global resource serialization complex
- Controlling the system
- Controlling time-sharing
- Controlling jobs
- Controlling an external writer
- Controlling shared DASD
- Controlling a global resource serialization complex
- Controlling system information recording
- Interacting with system functions
- Responding to failing devices
- Reconfiguring system resources
- Rebuilding a global resource serialization ring
- Quiescing the system
- Stopping the system

## Starting, Loading, and Initializing the System

Before the system can do work, you must:

1. Prepare the system hardware
2. Load the system software
3. Initialize the system software
4. Set the time and date, as required
5. Start the job entry subsystem (JES2 or JES3) and specify all job entry subsystem parameters

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

### 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 real storage configuration.
4. Mount all volumes the system needs for its work and turn on the drives you use.
5. Turn on power for all devices you plan to use as multiple-console support (MCS) consoles.
6. Switch into the configuration all control units for devices that the system needs.

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

## Loading the System Software

Once the system hardware is ready, you can use the hardware console to load the system software. Loading the system has several different names, depending upon when you do it and what it does. The first load after system generation or any load during which the link pack area is reloaded is called a “cold start.” A load after you turn the processor power on, where the segment and page tables for the link pack area are rebuilt, is called a “quick start.” A load after a system failure, where the segment and page tables for the link pack area are rebuilt but virtual I/O is saved, is called a “warm start.”

Load the system as follows, using the SYSCTL (CC012) frame:

1. Specify the target processor (CP0, CP2, CP1, or CP3) for initialization.
2. Specify the device number of the device that contains the system residence volume.
3. If you want to load an alternate nucleus, specify the alternate nucleus identifier (0-9, A-Z) in the first position of the load parameter field. If you do not specify an alternate nucleus identifier, the system loads a standard (or primary) nucleus.
4. If you want to load an alternate I/O configuration, specify the two-digit I/O configuration identifier (0-9, A-Z) in the second and third position of the load parameter field. If you do not specify an alternate I/O configuration, the system loads the I/O configuration that has the identifier 00.
5. Select the operator load function.

Selecting the 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 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 *Message Library: System Codes*. The processor operations manual tells you how to display the PSW.

## 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 the NIP console during an initialization process. Although some of the values are defined at system generation time, you must supply others as system parameters in response to the following message from the system:

```
IEA101A SPECIFY SYSTEM PARAMETERS FOR RELEASE xx.yy.zzz
```

So that you don't have to enter parameters one at a time, you usually respond to this message with IEASYSxx members of the partitioned data set SYS1.PARMLIB. Each IEASYSxx member can contain, in a special form, many different initialization parameters. You can respond to message IEA101A with IEASYSxx members in one of two ways:

1. Perform an ENTER action to specify the default parameters in IEASYS00.
2. To specify one or more alternate sets of parameters in IEASYSxx, such as IEASYSaa and IEASYSbb, enter:

```
REPLY 00,SYSP=(aa,bb,...)
```

The parameters in the alternate sets supplement or override the default parameters in IEASYS00.

If you need more detailed information on SYS1.PARMLIB and its members, see "System Initialization" in *Initialization and Tuning*.

### Specifying an Alternate Master Catalog

During each initialization, the system issues the following message:

```
IEA347A SPECIFY MASTER CATALOG PARAMETER
```

You must respond to this message. Respond as follows:

- 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 the master catalog, enter two characters for *nn*. The two characters may be any combination of A-Z and 0-9.

### 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 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=85.301,CLOCK=22.31.53  
*00 IEA888A LOCAL DATE=85.301,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, 1985 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, 1985, reply as follows:

```
R 00,DATE=85.302,CLOCK=13.00.00,GMT
```

The system responds with:

```
IEA888A GMT    DATE=85.302,CLOCK=13.00.00
*00 IEA888A LOCAL DATE=85.301,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=85.302
```

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
```

The following message appears signalling that the console is now a master console and is controlled by MCS:

```
IEE152I ENTER CANCEL D C,K
```

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.

## 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 on starting JES, see either *Operations: JES2 Commands* or *Operations: JES3 Commands*.

## Building a Global Resource Serialization Complex

To build a global resource serialization complex, your installation must coordinate the IPL procedures of the systems that are to be part of the complex. Your installation selects a system to start the global resource serialization complex. The system that starts the complex also starts the global resource serialization ring. All other systems defined in the same global resource serialization complex must join the complex. A system that joins the complex also joins the global resource serialization ring.

IPL the system selected to start the complex with `GRS=START`. The IBM-supplied default for the `GRS=` option system parameter is `JOIN`. Your installation can either override the parameter for the system that is starting the complex (as well as for any systems that are not going to participate in the global resource serialization complex) or IPL with the default. If your system is to start the complex and you IPL with the default, message `ISG006I` appears on your console. This message tells you that you cannot join a complex because a complex does not exist. In response to message `ISG009D`, which follows message `ISG006I`, you can reply `START`. This action has the same effect as specifying `START` for the `GRS` system parameter. The system now starts both the complex and the ring.

Once one system has started the complex, other systems can join the complex by IPLing with GRS=JOIN. If more than one system is IPLed with GRS=START or replies START to message ISG009D, only the first system to reply START starts the complex. All other systems receive message ISG005I, indicating that a complex already exists, followed by message ISG009D, asking them to IPL again or to reply to the message with JOIN or NONE.

Before a system can join the complex, the global resource serialization ring must exist. If a failure disrupts the ring before or while a system is trying to join the complex, message ISG006I appears indicating that no active global resource serialization system exists. The ring must be rebuilt before the system that is trying to join the complex can do so.

After the complex is built, there is no operational distinction between the system that starts the complex and any system that joins the complex.

## Controlling the System

To control the operating system effectively, you need to know the current status of the system. Knowing status information, such as the number of active jobs and teleprocessing functions, enables you to take appropriate actions to operate the system efficiently and to correct potential problems.

MVS/XA has system and subsystem commands that display job and system status either when you request it 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.

The system restart function provides another way to control certain system functions.

## Displaying the 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 make sure that you are keeping 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 then use the information in the display to take any necessary actions.

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.

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.



## 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) adapters
- Direct access storage devices (DASDs)
- Graphic devices
- Magnetic tape units
- Communication equipment
- Unit record devices

This command displays the status of the devices and the jobnames and ASIDs of device users. Knowing the jobs and ASIDs using a particular device allows you to determine whether you can take a particular device offline.

The **DISPLAY M** command allows you to keep track of the availability of channel paths and devices on these paths.

If your system includes MVS/XA DFP Version 2, 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/XA 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.

## 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.

Using the **SEND** command, you can communicate with operators of other consoles in your system, and you can send messages to remote terminals.

## 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).

## Using the System Restart Function

You can use the system restart function basically to:

- Restart the system after you have entered a QUIESCE command (described under “Quiescing the System” later in this part)
- Restart the system from a restartable wait state that is specified in *Message Library: System Codes*

You can also use the restart function to restart the system when it behaves abnormally, as when it enters an invalid wait state or a disabled loop. An invalid wait state exists when the wait state code in the PSW (IC) is not listed in *Message Library: System Codes*. Symptoms of a disabled loop are:

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

The procedure for performing the system restart function -- which loads a new PSW from storage location 0 -- ranges from pressing a RESTART key on the operator’s console to specifying one of several restart actions on an operator frame. The system restart function basically causes work to resume on a system that is quiesced or in a valid restartable wait state. Using the SYSCTL (CC012) frame to restart the system, however, you can specify restart reasons and tailor restart processing to fit other situations:

- When you suspect that a unit of work is causing an invalid wait state or a disabled loop and you cannot terminate the suspected unit of work by using the CANCEL or FORCE commands, request the restart function with REASON 0. In response, the system displays the following information about the unit of work in progress:
  - Address space identifier (ASID)
  - Job name
  - Step name
  - Current PSW
  - Whether the system is in task, SRB, wait, or “other” mode

To avoid terminating jobs unnecessarily, check that this information describes the unit of work you suspect has a problem. If it does, reply ABEND to abnormally terminate the interrupted program and invoke the necessary recovery routines. In contrast, if the displayed information does not describe the unit of work that you suspect has a problem, reply RESUME to end further restart processing and allow the interrupted work to continue. 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

- Request the restart function with REASON 1 when you suspect a system problem that is not related to the work currently in progress. In response, the system diagnoses and repairs some problems that might be causing it to behave abnormally. Among its actions, the system:
  - Checks system dispatchability. If the system is non-dispatchable, it 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.

*Note:* Performing the restart function with REASON 1 might cause the system to immediately terminate some address spaces. Request the restart function with REASON 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 1. (The message you receive if there are no batch jobs or time-sharing users is an example of this notification.) You only get this information, however, if you request the restart function with REASON 1:

- On a processor that can communicate with the master console or its first designated alternate
- When the master console or its first designated alternate is a display console

If, for any reason, 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 you take any further action.

## 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 the system concurrently, and the execution of each job is controlled primarily by the remote terminal user. Thus, time-sharing can be defined as the shared, conversational, and concurrent use of a computing system by a number of users at remote terminals.

Time-sharing in MVS/XA is provided by TSO, formerly an option of the control program (hence, the name time-sharing option) and now a standard feature. A time-sharing environment in which communication between TSO and the terminal is controlled by the virtual telecommunications access method (VTAM) is called TSO/VTAM. TSO/TCAM is a time-sharing environment in which communication between TSO and the terminal is controlled by the telecommunications access method (TCAM).

You start and stop TSO/VTAM by using the START and STOP commands; you start and stop TSO/TCAM by using the MODIFY command. You can modify TSO/VTAM by using the

MODIFY command; you cannot modify TSO/TCAM. Other commands that control time-sharing are the same for both environments.

For more information about VTAM, see *VTAM Operation*. For more information about TCAM, see *TCAM Operation*.

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, use the SEND command. With it, you can:

- 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.BROADCAST, has mail and notices sections.

## Controlling Jobs

Jobs are a basic unit of work for the system. 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.

### 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. For a description of the internal reader, see *SPL: System Modifications*.

### 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.

## Modifying a Job's Parameters

Using the MODIFY command, you can change the characteristics of a job. The programmer must have set up the job so that some of its parameters can be changed and must give you the new parameters.

## Modifying a Job's Performance Group

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

## 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 real 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, he 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 *MVS/XA JCL*.

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 re-IPL 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 re-IPL 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 *Checkpoint/ Restart*.

## Controlling an External Writer

Use an external writer to write SYSOUT data to a tape or a DASD, or to write a SYSOUT data set with a SYSOUT JCL parameter that specifies a special installation-supplied writer routine. You can also use an external writer to write output to any QSAM-supported devices attached to the processor; these devices are called local devices. You cannot use an external writer to write SYSOUT data to remotely-attached devices; an external writer always writes to local devices.

To start an external writer, enter a START command that specifies either the IBM-supplied procedure (XWTR) or an installation-supplied procedure. You can specify the device on the START command and optionally specify whether the external writer should begin immediately to process all output directed to local devices and enqueued in the output classes assigned to the external writer. If one or more output classes are specified in the START command or on the EXEC statement in the cataloged procedure used to start the external writer, the external writer begins to process immediately. If no classes are specified, the external writer waits for you to assign data selection criteria with the MODIFY command.

Using the MODIFY command, you can cause the external writer to select data sets according to output class, job-id, special forms name, and destination. You need specify only those parameters that are to be changed. Parameters omitted from the MODIFY command keep their current value as set by defaults or a previous MODIFY command. Using the MODIFY command, you can also cause the external writer to pause for operator intervention.

If you use the immediate-processing option, specify output classes on the EXEC statement or the START command. The absence of a program name in the SYSOUT parameter of the DD JCL statement indicates that the standard writer is being used. If you do not use the immediate-processing option (classes are not specified), the defaults are any class, any destination, any job, any forms, and any writer routine. However, these options are not used until a MODIFY command is entered.

When no particular forms are specified, forms-mounting is done on demand as directed by the external writer. When no particular writer routine is specified, writer routines are invoked as required by the external writer, subject to your authorization. You are asked to authorize the use of any routine not previously authorized for this external writer. If you do not want to authorize the routine, you can either delete the data set or cause it to be written by the standard writer (STDWTR).

You cannot specify print trains and forms control buffers as data set selection criteria. These two parameters, plus the 3525 interpret feature, are specified on the DD statement in the cataloged procedure used to start the external writer (and, possibly, to override the START command) and apply to all data sets written by this writer. To separate output by print train, forms control, and punch card interpretation, you must direct the SYSOUT to separate output classes.

For more information on external writers, see *MVS/XA SPL: System Modifications*.

## Controlling Shared DASD

The shared direct access storage device (DASD) option allows multiple systems to access common data residing on direct access storage devices. This sharing is accomplished through a hardware feature of the DASD control unit together with the reserve/release function of the operating system or through the global resource serialization function of the operating system. The reserve function reserves specified devices for use only by the system issuing the "RESERVE." The release function removes a specified device from reserved status, allowing the device to be reserved by another system.

The shared DASD option is included in the operating system during system generation. It allows up to eight systems to share a pool of 3330/3333, 3330V, 3340/3344, 3350, 3375, or 3380 direct access storage devices. Two systems can share a pool of 2305 model II direct access storage devices. Up to two paths from a system to a control unit are allowed.

The advantages of the shared DASD option include:

- Reducing the amount of time you have to spend moving volumes from one system to another.
- Minimizing the updating of data sets. You have to update only one instead of two or more duplicates.
- Simplifying scheduling. Unless the job has other special requirements, you can run a job needing a specific data set on a shared device on any of the sharing systems.

The system uses a multichannel hardware switch to control the reserving and releasing of the shared device and data. The multichannel switch allows a common control unit to be switched on a first-come, first-served basis among several channel paths, each possibly from a different system. (On systems with 3330 devices, make sure any hardware toggle switches marked "tagged" and "untagged" are always set to the "tagged" position.)

If the control unit or device is busy with one of the other sharing systems, an attempt to initialize the system from a shared device fails. Continue to retry the initialization until it is successful; the retry has no adverse effects on the other systems. Likewise, in a single system environment, system reset only resets the functions of a shared control unit or device that belongs to the system that issues the system reset. Any function related to another system remains undisturbed. A selective or malfunction reset has no effect on device reservations or status.

If, when you initialize a system having shared DASD, a second system has reserved the shared device, the initializing system waits one and a half seconds and then issues this message:

```
* id IOS120A DEVICE ddd SHARED. REPLY 'CONT' or 'WAIT'
```

The recommended reply is 'WAIT'. 'WAIT' causes the system to wait until the device is released by the system that is currently using it. If your system waits more than one minute, re-IPL. If the device is still reserved on the other system when you re-IPL, message IOS120A is issued again. Reply 'CONT' and the device is marked offline to the system performing the IPL. If you reply 'CONT', you might be unable to establish the proper combination of mount characteristics (listed later in this section under "Specifying Shared DASD Mount Characteristics"). Therefore, it is recommended that you run *no* jobs on the IPL system that might require the unavailable device, and notify your system programmer of the problem.



These requirements do not apply to the 3330V. See *Mass Storage System Extensions Operation* for more information about the 3330V.

*Note:* You must reply WAIT for 3344-emulated 3340 devices and 3350-emulated 3330 devices that are to be marked permanently resident by the volume attribute list (VATLST) facility.

## Shared DASD Operating Guidelines

When operating a system with shared DASD, be aware of the following considerations:

- Whenever you change the mount characteristics of a shared volume, the status of a device, or both, you must maintain a valid combination of mount characteristics across all sharing systems. Remember that the VARY OFFLINE command is always the first step of the mount procedure and that the procedure must be done in parallel on all sharing systems.
- If there is a hardware malfunction on a device other than the system residence device, you must vary the failing device offline on all sharing systems. You can then mount the shared volume on another shared device, if one is available, as long as you follow parallel mount procedures on all sharing systems.
- Normally, if there is a reserved device on a path, you cannot remove that path with a VARY path or CONFIG channel path command. You can, however, release a reserved device and remove a path to it if you issue CONFIG CHP,OFFLINE,FORCE. If you try to remove a path to a reserved device with any other CONFIG command or with a VARY command, the system issues message IEE379I or IEE719I and does not execute the command.

## Specifying Shared DASD Mount Characteristics

Volume characteristics, device status, and volume mounting and demounting are all affected by the shared DASD option. One of the following combinations of mount characteristics and device status must be in effect for each device that is physically shared among processors.

| System A             | Other Systems  |
|----------------------|--|
| Permanently resident | Permanently resident   |
| Reserved             | Reserved   |
| Removable            | Offline—Non-JES3 devices   |
| Removable            | Removable—JES3-managed devices   |
| Offline              | Permanently resident, reserved, or removable (In JES2, if a device is removable in one system, it must be offline in all others) |

After system initialization, a valid set of mount characteristics must be established before the system begins device allocation. The system programmer can make sure that a valid set of mount characteristics exists for a newly-initialized system by specifying the mount characteristics of shared devices in the VATLST member of SYS1.PARMLIB. The mount characteristics of a volume, the status of a device, or both, can be changed in one system as long as the resulting combination across all systems is valid.

You must put volumes that are not permanently resident in reserved status using the MOUNT command before starting jobs needing the shared volume. To change volumes that are reserved, you must:

1. Use the VARY command to put the device offline to each sharing system and wait for the offline message in each system. The device does not go offline until the message is issued. If no jobs are in progress, the offline message does not appear on the console. Issue a START DEALLOC command to receive the message that the device is offline.
2. Use the MOUNT command to notify each sharing system of the unit where the new volume is being placed, and to put the volume in reserved status.
3. Use the MOUNT command to mount the volume.

Jobs requiring this new volume should not be selected by an initiator before the volume is mounted. You can hold up job selection by one of the following:

1. Using the TYPRUN=HOLD parameter on the job card.
2. Using the appropriate subsystem command.
3. Assigning the job to a job class and not activating that class for subsystem scheduling.

After the volume is mounted, use a job entry subsystem command to let the job be selected in cases 1 and 2; activate the class for subsystem scheduling in case 3.

When you want a shared non-JES3 device to be allocated by only one system, the operator of each system sharing the device should use the VARY command to place the device offline to their systems.

## Guidelines for Sharing Data Sets

Any of your installation's own data sets can be shared. In addition, the following system data sets can be shared:

- Private catalogs
- Volume tables of contents (VTOCs) on all shared volumes
- JES2 and JES3 spool data sets

The following data sets can be shared as long as concurrent updating is not attempted:

- SYS1.IMAGELIB
- SYS1.LINKLIB if all sharing systems have the same modules with common name configurations
- SYS1.LPALIB if all sharing systems have the same modules with common name configurations
- SYS1.PROCLIB when not on the IPL volume
- SYS1.BROADCAST through global resource serialization
- SYS1.UADS through global resource serialization

The following system data sets cannot be shared:

- Master catalog
- Page data sets
- PASSWORD
- SYS1.DUMPxx
- SYS1.LOGREC
- SYS1.MANx
- SYS1.NUCLEUS
- SYS1.STGINDEX
- SYS1.SVCLIB

## Controlling a Global Resource Serialization Complex

Using the DISPLAY GRS and VARY GRS commands, you can monitor and maintain a global resource serialization complex.

DISPLAY GRS helps you to monitor the complex because it displays system status, CTC adapter status, and resource and contention information. For example, Figure 1-1 shows a ring consisting of systems A and B, which are connected by CTC 9C0. If you issued DISPLAY GRS,ALL from system B, you might get the following response:

```
ISG020I  hh.mm.ss  GRS  STATUS  [id]
          LIST  TYPE  QNAME  RNAME
          INCL  GEN  SYSDSN
          EXCL  GEN  SYSCTLG
          EXCL  SPEC  SYSDSN  PASSWORD
          EXCL  SPEC  SYSDSN  SYS1.BROADCAST
          EXCL  SPEC  SYSDSN  SYS1.DAE
          EXCL  GEN  SYSDSN  SYS1.DUMP
          EXCL  SPEC  SYSDSN  SYS1.LOGREC
          EXCL  GEN  SYSDSN  SYS1.MAN
          EXCL  SPEC  SYSDSN  SYS1.NUCLEUS
          EXCL  GEN  SYSDSN  SYS1.PAGE
          EXCL  SPEC  SYSDSN  SYS1.STGINDEX
          EXCL  SPEC  SYSDSN  SYS1.SVCLIB
          EXCL  SPEC  SYSDSN  SYS1.UADS
          EXCL  GEN  SYSIGGV2  PAGE08.CATALOG
          EXCL  GEN  SYSIGGV2  PAGE13.CATALOG
          EXCL  SPEC  SYSVTOC  SPOOL1
NO ENTRIES EXIST IN THE RESERVE CONVERSION RNL
NO RESOURCE CONTENTION EXISTS
SYSTEM      STATE      COMM      SYSTEM      STATE      COMM
B           ACTIVE
LINK       STATUS      TARGET    LINK       STATUS      TARGET
9C0        IN-USE      A
```

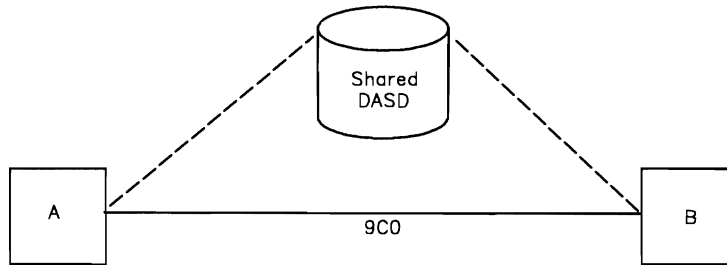
The response to the DISPLAY GRS,ALL command includes:

- Contents of the SYSTEM inclusion, SYSTEMS exclusion, and RESERVE conversion resource name lists (RNLs)
- Resource contention information
- System information
- Local CTC adapter status

If you want more specific information about your complex, use the following DISPLAY GRS operands:

| OPERANDS   | DISPLAY                                     |
|------------|---|
| SYSTEM     | System information                          |
| LINK       | CTC adapter status                          |
| CONTENTION | Resource contention information             |
| RNL=       | Contents of one or all RNLs                 |
| RES=       | List of major names or resource information |

You can route the output from DISPLAY GRS using the MSGRT command.



**Figure 1-1. Example of a Two-System Global Resource Serialization Ring**

VARY GRS helps you to maintain the complex because it lets you reconfigure the ring in response to changing conditions, such as a system or CTC adapter failure. The VARY GRS command is primarily used in recovery situations. You can use VARY GRS to:

- Temporarily suspend (QUIESCE) a system's participation in the global resource serialization ring
- Cause a quiesced global resource serialization system to resume (RESTART) its participation in the ring
- Rebuild (RESTART) a disrupted global resource serialization ring; that is, a group of inactive systems
- Remove (PURGE) a quiesced system from the global resource serialization complex

Specifically, if you want to temporarily suspend a system's participation in a ring, issue the VARY GRS,QUIESCE command. Users on the quiesced system retain control of all global resources that they own and continue to wait for global resources they requested but do not yet own. However, processing of all new requests for global resources made by requestors on the quiesced system is suspended. A quiesced system can resume its participation in the ring by means of the VARY GRS,RESTART command. Once the quiesced system rejoins the ring, processing of all previously-suspended requests for global resources is resumed.

An inactive system is one whose participation in the global resource serialization ring has been suspended because a system or CTC adapter failure has disrupted the ring. An inactive system can rebuild the global resource serialization ring using the VARY GRS,RESTART command. Any systems that do not join the new ring as a result of this command become quiesced systems if they can communicate with systems in the ring.

If you want to remove a system from the complex, issue VARY GRS,PURGE from another system. In response to this command, all global resources owned by the system being purged are released, and all outstanding requests for global resources made by the system are deleted.

Thus, indiscriminate use of the VARY GRS,PURGE command can cause resource integrity problems. Notify the system programmer if any resources are held by the system being purged. A purged system must be re-IPLed with GRS=JOIN to rejoin the global resource serialization complex and ring.

See the VARY command in Chapter 4 for a complete explanation of the syntax and use of VARY GRS.

## 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:

- The hardcopy log
- The system log
- System management facilities (SMF)
- System trace
- The generalized trace facility (GTF)
- Master trace

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

### The Hardcopy Log

The hardcopy log is a record of all system message traffic. In a JES3 system, the console's hardcopy log is always written on the system log. In a JES2 system, the hardcopy log usually is written on the system log but may be written on a printer console, if the installation chooses. The hardcopy includes:

- Operator commands
- System responses to operator commands
- System requests
- Operator responses to system requests
- Most messages appearing on active consoles
- Messages suppressed by the message processing facility (MPF)
- Messages suppressed from all consoles by the LEVEL parameter in the CONSOLxx member of SYS1.PARMLIB or the CONTROL V,LEVEL command

## The System Log

The system log is a direct access data set that stores messages and commands until they reach a threshold specified at system initialization. When the threshold is reached, the system uses a system writer to print the contents of the system log. You can force the current system log data set to be queued for printing before the threshold is reached by issuing the WRITELOG command.

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 problem programs using a write to log (WTL) macro instruction
- Descriptions of unusual events that you enter using the LOG command
- Hardcopy log data

To learn how to use the system log and the hardcopy log, see “Using the Hardcopy and System Logs” in Chapter 2.

## 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. An installation can use this information to evaluate system workload, resource use, and performance. System-related SMF routines record information on workload, configuration, and paging activity. Job-related SMF routines record information on processor time, SYSOUT activity, and data set activity for each job, job step, and TSO user.

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. The system programmer must create each new SMF data set, as described in *System Management Facilities*. At system initialization time, SMF uses the primary data set as the active recording data set unless it is full. The primary data set is the first data set listed in the DSNAME parameter of the SMFPRMxx member of SYS1.PARMLIB. 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.

To speed execution, it is a good practice to submit the dump job before it is needed, hold it in the job queue, and release it when necessary. Sample JCL, as well as more information on the IEFU29 exit, appears in *System Management Facilities*.

**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 can use the HALT EOD and SWITCH SMF commands 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 HALT or SWITCH command. (See the DISPLAY, HALT, and SWITCH commands in Chapter 4.)

**Restarting SMF:** Because SMF runs in its own address space, you can restart SMF with the SET SMF command. When you enter the SET SMF command, the following 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
```

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

## 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 the TRACE command in Chapter 4. For more information on using system trace, see *Diagnostic Techniques*.

## 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 on starting and stopping GTF, see the START and STOP commands in Chapter 4. For information on using GTF, see *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 same information as the hardcopy log. 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 hardcopy 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 *Diagnostic Techniques*.

## Interacting with System Functions

Most resource allocation, error recovery, and system monitoring functions in MVS/XA 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 cataloged procedures
- The MOUNT command
- TSO LOGONs

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 member of SYS1.PARMLIB. 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 SYS1.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 be assigned 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 *Operations: 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 device number assigned to each device.
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 IOS109I and attempts to recover from the hot I/O condition. (See *SPL: System Modifications* 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:

```
IOS111A - HOT NON-RESERVED DIRECT ACCESS DEVICE (Wait state 111)
IOS112A - HOT RESERVED DIRECT ACCESS DEVICE (Wait state 112)
IOS110A - 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 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. See *Message Library: System Messages* for an explanation of the messages and to *Message Library: 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/XA 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 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 removes the last path to the device

*Note:* Because you might release a reserve 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.

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 I/O and allocations.

*Note:* You cannot take a boxed device out of the boxed state by replying with the device name to the allocation recovery message, IEF238D.

## 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.
- Devices of the 3330V type. See *Mass Storage System Extensions Operation* for information about the 3330V.
- 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.

## Reconfiguring Resources

A computer installation consists of one or more configurations. A configuration is any combination of resources, such as processors, storage elements, and channel paths, on which an operating system can run. **Reconfiguration** is the process of dynamically adding hardware units to, or removing hardware units from, a configuration. Resources can be added to the configuration (placed online) to make them available to the system or removed from the configuration (taken offline) to make them unavailable to the system. An installation might reconfigure its resources for one of the following reasons:

- To service or repair some of the resources
- To isolate a failing item to prevent it from interfering with other functioning resources
- To change the resources available to the processor as processing requirements change

Reconfiguration has two parts: logical and physical. **Logical reconfiguration** is the process that allows or prevents the use of a resource by the operating system; **physical reconfiguration** is the process that allows or prevents the use of a resource by the hardware. The CONFIG command with the CPU, CHP, STOR, ESTOR, and VF keywords causes both physical and logical reconfiguration of the CPU, channel paths, real and extended storage, and Vector Facilities. When you issue a CONFIG command, the control program performs the logical reconfiguration and requests that the service processor perform the physical reconfiguration. In addition, you can issue a VARY command to cause the logical reconfiguration of I/O devices or I/O paths.

The actions you perform to reconfigure the system depend on the type of processor or **processor complex**. A processor complex, such as the 3083, 3081, or 3090 model 400, contains one, two, or four central processing units (CPUs).

- A **uniprocessor** or UP has one CPU.

Depending on the particular type of processor complex, an installation can configure offline a real storage element, real storage ranges, channel paths, and devices. In a UP system, the main purpose of reconfiguration is to take failing resources offline, an action that allows the system to continue operation.

- A **multiprocessor (MP)** has more than one CPU.

Depending on the particular type of processor complex, an installation can configure offline CPUs, Vector Facilities, real storage, extended storage, channel paths, and devices. In addition, the 3084 and 3090 model 400, which contain four CPUs, can be **partitioned**, or separated into two independent configurations. "Partitioning a 3084 or 3090 Model 400 Processor Complex," describes the procedures for creating two configurations in a processor complex.

If a reconfiguration attempt fails, the control program tries to return the resource to its original state and issues a message that explains what state the resource is in and why the reconfiguration attempt failed.

To change the configuration of the resources, use the following **CONFIG** and **VARY** commands:

| RESOURCE                 | COMMAND            |
|--------------------------|--------------------|
| CPU                      | CONFIG CPU         |
| Channel path             | CONFIG CHP         |
| Real storage             | CONFIG STOR        |
| Real storage element     | CONFIG STOR(E=x)   |
| Extended storage element | CONFIG ESTOR(E=id) |
| Vector Facility          | CONFIG VF          |
| I/O device               | VARY devnum        |
| I/O path                 | VARY PATH          |

For example, if you wanted to take a storage element offline, you would issue the **CONFIG STOR(E=x),OFFLINE** command.

*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).

Physical reconfiguration might not be supported for all hardware units by all processor types. While MVS/XA might support logical reconfiguration of a resource, the hardware might not support physical reconfiguration of the same resource. For example, the 4381 processor does not support physical reconfiguration of any resource; however, MVS/XA running on that processor supports logical reconfiguration of the CPU but not logical reconfiguration of real storage. Refer to the appropriate *Functional Characteristics* manual for physical reconfiguration characteristics.

## Alternate CPU Recovery

Alternate CPU recovery (ACR) attempts to maintain the availability of the system when one of the processors is unable to continue. The functioning processor attempts to perform the work that was in progress on the failing processor until the system can continue normal processing with the failing processor logically offline. You are informed of the loss of processor functions. Tasks requiring the lost functions are terminated or not allowed to run. If the system is unable to continue processing because of an unrecoverable error, and an error occurs during ACR processing, the system enters a wait state. Refer to *Message Library: System Codes* to find out what to do.

## Partitioning a 3084 or 3090 Model 400 Processor Complex

A processor complex that consists of four CPUs can be partitioned. These processor complexes — the 3084 and 3090 model 400 — can run either in **single-image mode**, where all of the CPUs, storage elements, and devices are in a single configuration, or in **physically-partitioned mode**, where the hardware units are divided into two configurations. An installation can use partitioning as an operational convenience or as an aid to recovery. In the first case, partitioning allows one operating system to run in one configuration, or **side**, and another in the other side. For example, an installation could run MVS/XA in one side and VM/SP in the other. In the second case, an installation could give control of a failing side to service personnel for diagnosis and repair and still have the other side operating normally.

The two sides of a 3084 and 3090 processor complex are known to hardware sometimes as side A and side B and sometimes as side 0 and side 1, but always to MVS/XA as side 0 and side 1.

*To partition a 3084 processor complex into two configurations*, first determine which side you want to take offline. Side 0 and side 1 (known to hardware as Side A and Side B) can contain the following reconfigurable resources and identification numbers:

| SIDE 0 (or Side A)   |   | SIDE 1 (or Side B)   |   |
|----------------------|---|----------------------|---|
| CPU                  | 0 | CPU                  | 2 |
| Real storage element | 0 | Real storage element | 2 |
| I/O engine           | 0 |                      |   |
|                      |   | CPU                  | 3 |
|                      |   | Real storage element | 1 |
|                      |   | Real storage element | 3 |
|                      |   | I/O engine           | 1 |
|                      |   |                      |   |

**Figure 1-2. Identifying the Resources in a 3084 Processor Complex**

For example, to take side 0 offline, issue the following commands from the master console in this order:

```
CONFIG CHP(ALL,0),OFFLINE,UNCOND
CONFIG CPU(0,2),OFFLINE
CONFIG STOR(E=0),OFFLINE
CONFIG STOR(E=2),OFFLINE
```

You might have to remove some devices from side 0 before the channel paths can go offline. Use either a **VARY** command for each device or, as in the preceding example, specify the **UNCOND** option when taking the channel paths offline with the **CONFIG** command. The **UNCOND** option specifies that the system is to take the channel path offline, even if it is the last path to an allocated device.

After the last channel path is removed from side 0 (or I/O engine 0), the system automatically takes I/O engine 0 offline. When no CPUs, storage elements, or channel paths remain on side 0, the system takes side 0 offline. Then, use the side 1 hardware console to place the 3084 in physically-partitioned mode. To use side 0, perform a power-on-reset from the side 0 hardware console and IPL side 0.

When you want to operate the 3084 processor complex in single-image mode again, you can bring down the operating system on side 0 and use the side 1 hardware console to place the 3084 in single-image mode. To bring online the resources for side 0, issue the following commands in this order on the master console:

```
CONFIG STOR(E=0),ON
CONFIG STOR(E=2),ON
CONFIG CPU(0,2),ON
CONFIG CHP(ALL,0),ON
```

*MVS/XA Planning: Recovery and Reconfiguration* contains the detailed information - including software and hardware commands - you need to partition the 3084 processor complex. The book also contains examples of reconfiguration scenarios. *IBM 3084 Operators' Guide for the System Console* contains descriptions of the procedures you must follow at the hardware console.

**The procedure for partitioning the 3090 model 400 processor complex** is very similar to that for the 3084 with some important exceptions:

- The 3090 model 400 might have extended storage and Vector Facilities that can be configured.
- The 3090 model 400 does not have I/O engines.
- The numbering of the CPUs and storage elements is different.

Figure 1-3 describes the identification numbers you might need to know to partition the 3090 model 400.

| SIDE 0 (or Side A)       |     | SIDE 1 (or Side B)       |     |
|--------------------------|-----|--------------------------|-----|
| CPU                      | 1   | CPU                      | 2   |
| Vector Facility          | 1 * | Vector Facility          | 2 * |
| Real storage element     | 0   | Real storage element     | 1   |
| Extended storage element | 0   | Extended storage element | 1   |

|                          |     |                          |     |
|--------------------------|-----|--------------------------|-----|
| CPU                      | 3   | CPU                      | 4   |
| Vector Facility          | 3 * | Vector Facility          | 4 * |
| Real storage element     | 2   | Real storage element     | 3   |
| Extended storage element | 2   | Extended storage element | 3   |

\* The Vector Facility has the same number as the processor to which it is attached. You can have up to four Vector Facilities on a 3090 model 400.

**Figure 1-3. Identifying the Resources in a 3090 Model 400 Processor Complex**

Before you issue the commands to partition the 3090 model 400 processor complex, you might want to use the DISPLAY M=SIDE command to check the status of its resources. Figure 1-4 shows an example of the output of the DISPLAY M=SIDE command:

```

D M=SIDE
IEE174I 09.16.18 DISPLAY M 225
SIDE STATUS
SIDE:          0                      1
STATUS:        ONLINE                 ONLINE
CPU:           1 2                    3 4
VF:           1 2                    3 4
CHP:          0-F*  10-1F  20-2F*    40-4F*  50-6F
STOR(E=X):    0-1                      2-3
ESTOR(E=X):   0-1                      2-3

TOTAL STOR:   128M          UNASSIGNED: 0M
TOTAL ESTOR:  512M
*=OFFLINE

```

**Figure 1-4. Output of the DISPLAY M=SIDE Command**

The procedure for partitioning the 3090 model 400 is similar to partitioning the 3084 processor complex. For example, to take side 0 offline, you would issue the following commands in this order:

```

CONFIG CHP(ALL,0),OFFLINE,UNCOND
CONFIG CPU(1,2),OFFLINE
CONFIG ESTOR(E=0),OFFLINE
CONFIG ESTOR(E=1),OFFLINE
CONFIG STOR(E=0),OFFLINE
CONFIG STOR(E=1),OFFLINE

```

*Note:* The CONFIG CPU(1,2),OFFLINE command also takes offline the Vector Facilities attached to CPU 1 and CPU 2.

When no CPUs, real and extended storage elements, and channel paths remain on side 0, use the side 1 hardware console to place the 3090 model 400 processor complex in physically-partitioned mode. To use side 0, perform a power-on-reset from the side 0 hardware console and IPL side 0.

*MVS/XA Planning: Recovery and Reconfiguration* contains the detailed information - including software and hardware commands - you need to partition the 3090 model 400 processor complex. The book also contains examples of reconfiguration scenarios.



## Rebuilding a Global Resource Serialization Ring

Whenever a system or a CTC adapter failure disrupts a global resource serialization ring, effective recovery involves several factors:

- Whether the global resource serialization complex is completely or only partially connected. A completely-connected complex offers more options for recovering the ring when a failure occurs.
- The availability of alternate CTC adapters. If an alternate CTC adapter is available, your installation can recover more completely from the failure of a CTC adapter. Global resource serialization can continue with the alternate while the problem is resolved.
- The communication and coordination of actions among operators of the systems in the global resource serialization complex. Operators must know what to do in a particular recovery situation before the problem occurs. Because the major goal during recovery is to protect the integrity of resources, operators must know, for example, about the resource integrity exposures that they might create by issuing VARY GRS,PURGE. The following descriptions of system and CTC adapter failures clearly show the need for communication and coordination among operators.

### After a System Failure

If a system in a global resource serialization ring fails and the failure is not in global resource serialization processing, message ISG023E notifies the operators of all systems in the ring that the ring is disrupted. The operator response when a system failure disrupts the global resource serialization ring depends upon whether or not one or more systems of the ring are authorized to rebuild the ring automatically.

If one or more systems of the ring are authorized to rebuild the ring automatically and it is possible for a system to do so, one system in the ring usually issues message ISG024I. That system attempts to rebuild the original ring. The other systems of the disrupted ring usually issue message ISG025E, which explains why each system is not attempting to rebuild the ring. A system cannot attempt to rebuild a disrupted ring if:

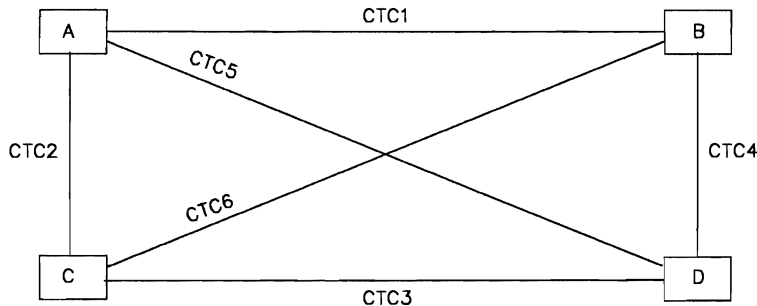
- The system's GRSCNFxx PARMLIB member does not authorize it to rebuild the ring.
- The system has already granted permission to another system to rebuild the ring, or another system is already attempting to rebuild the disrupted ring.
- The system can not communicate with at least half of the systems in the disrupted ring.
- A system error occurred while processing the VARY GRS(ALL),RESTART command the system issued to rebuild the disrupted ring.

If all systems in the disrupted ring issue message ISG025E (no system is attempting to rebuild the ring automatically), you have to talk to the operators on the other systems and rebuild the ring manually, as described below. If a system that attempts to rebuild the ring does not succeed in rebuilding the entire ring, have your system programmer analyze each system that did not rejoin the ring and follow the procedures described below to bring that system back into the ring.

If none of your global resource serialization systems are authorized to rebuild the disrupted ring automatically, you must always work closely with the operators of all the other ring systems to rebuild the ring manually. First, try to rebuild the ring and recover as many of the original systems as possible by issuing `VARY GRS(ALL),RESTART` on one of the systems that did not fail. (Issue the command on your most important system or on a system that can communicate with most of the other global resource serialization systems.) **Remember that indiscriminate use of the `VARY GRS,RESTART` command can cause resource integrity problems.**

If `VARY GRS(ALL),RESTART` does not rebuild the entire ring, have your system programmer analyze each system that did not rejoin the ring. (The system that failed and caused the original ring to be disrupted is probably not part of the rebuilt ring.) If any system that did not rejoin the ring does not require a re-IPL, issue `VARY GRS(sysname),RESTART` on any one of the active ring systems that can communicate with the excluded system to bring the excluded system into the ring. Note: It is recommended that you do not issue the `VARY GRS(*),RESTART` command on the excluded system to bring the excluded system back into the active global resource serialization ring. Issue the `VARY GRS(sysname),RESTART` command only after you fix the problem that caused the system to disrupt or not rejoin the ring. If any system that did not rejoin the ring does require a re-IPL, issue `VARY GRS(sysname),QUIESCE` (if the system is not already quiesced) and `VARY GRS(sysname),PURGE` for the excluded system on any one of the active ring systems. Then, re-IPL the system that did not rejoin the ring with `GRS=JOIN`.

For example, assume you have the following completely-connected global resource serialization ring, consisting of systems A, B, C, and D.



Suppose that system A disrupts the global resource serialization ring and none of the systems are authorized to rebuild the ring automatically. You would try to rebuild the ring by issuing `VARY GRS(ALL),RESTART` on system B, C, or D. If system A does not rejoin the ring after the `VARY GRS(ALL),RESTART` command is issued, determine if system A has to re-IPL. If you do not have to re-IPL, issue `VARY GRS(*),RESTART` on system A or issue `VARY GRS(A),RESTART` on any one of the systems in the ring. If you do have to re-IPL system A, issue `VARY GRS(A),PURGE` on system B, C, or D and then re-IPL system A with `GRS=JOIN`.

The fact that CTC6 exists means that a three-system ring (consisting of systems B, C, and D) can continue to actively serialize access to global resources until the problem with system A is fixed.

## After a CTC Adapter Failure

If a CTC adapter fails within the global resource serialization ring, messages ISG046E and ISG022E appear on at least one of the two systems connected by the CTC adapter that failed. Message ISG046E indicates the CTC adapter that is disabled and whether the failure is a hardware or a system software problem. Message ISG022E notifies the two connected systems that global resource serialization has been disrupted. Message ISG023E notifies all other systems in the ring that global resource serialization has been disrupted. The operator response when a CTC adapter failure disrupts the global resource serialization ring depends upon whether the failure is due to a hardware or system software problem and, once again, upon whether or not one or more systems of the ring are authorized to rebuild the ring automatically.

### Automatic Rebuilding of the Ring

If one or more systems of the ring are authorized to rebuild the ring automatically and it is possible for a system to do so, one system in the ring usually issues message ISG024I. That system attempts to rebuild the original ring. The other systems of the disrupted ring usually issue message ISG025E, which explains why each system is not attempting to rebuild the ring. A system cannot attempt to rebuild a disrupted ring if:

- The system's GRSCNFxx PARMLIB member does not authorize it to rebuild the ring.
- The system has already granted permission to another system to rebuild the ring, or another system is already attempting to rebuild the disrupted ring.
- The system cannot communicate with at least half of the systems in the disrupted ring.
- A system error occurred while processing the VARY GRS(ALL),RESTART command the system issued to rebuild the disrupted ring.

If there is an alternate for the CTC adapter that failed, the system attempting to rebuild the ring should succeed in rebuilding the entire ring. If there is no alternate, one or more of the global resource serialization systems will be left out of the rebuilt ring. To bring these inactive or quiesced systems back into the ring (when the CTC adapter failed because of a hardware problem), you must first either supply an alternate for the failed CTC adapter or repair the failed CTC adapter. To bring the alternate or repaired CTC adapter online, use the VARY device ONLINE command. Then, to bring the inactive or quiesced systems into the rebuilt ring, issue either VARY GRS(\*),RESTART on each inactive or quiesced system or VARY GRS(sysname),RESTART for each inactive or quiesced system on any system already in the ring.

*Note:* If message ISG046E indicates that a CTC adapter failed because of a system software problem, you must first take the CTC adapter offline with a VARY device OFFLINE command before you can bring it online with a VARY device ONLINE command. You can then attempt to bring its inactive or quiesced systems into the rebuilt ring. If, after you restore all CTC adapters and systems to the ring, a system software problem disables the same CTC adapter again, issue VARY GRS,QUIESCE (if the system is not already quiesced) and VARY GRS,PURGE for the system with the software problem. Then, re-IPL that system with GRS=JOIN to rebuild the original ring.

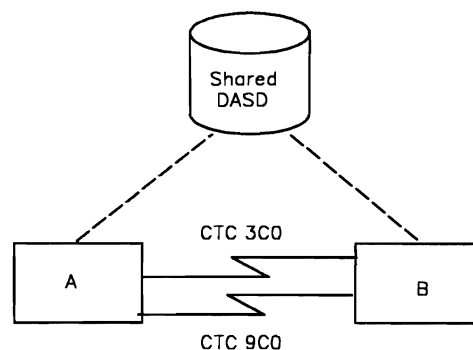
## Manual Rebuilding of the Ring

If all the systems in the disrupted ring issue message ISG025E (no system is attempting to rebuild the ring automatically) or if none of your global resource serialization systems are authorized to rebuild the ring automatically, work closely with the operators of all the other systems to rebuild the ring. First, issue `VARY GRS(ALL),RESTART` on a system with working CTC adapters. (Of course, if the ring consists of only two systems, you would have to issue the command on a system that has a non-working CTC adapter.) As with a system failure, make sure that another operator does not also try to rebuild the global resource serialization ring by issuing `VARY GRS(ALL),RESTART` on one of the other systems. Indiscriminate use of the `VARY GRS,RESTART` command can cause resource integrity problems.

If there is an alternate for the failed CTC adapter, issuing `VARY GRS(ALL),RESTART` should rebuild the original ring. If there is no alternate, one or more of the global resource serialization systems will be left out of the rebuilt ring. To bring these inactive systems back into the ring (when the CTC adapter failed from a hardware problem), you must first either supply an alternate for the failed CTC adapter or repair the failed CTC adapter. To bring the alternate or repaired CTC adapter online, use the `VARY device ONLINE` command. Then, to bring the inactive or quiesced systems into the rebuilt ring, issue either `VARY GRS(*),RESTART` on each inactive or quiesced system or `VARY GRS(sysname),RESTART` for each inactive or quiesced system on any system already in the ring.

*Note:* If message ISG046E indicates that a CTC adapter failed because of a system software problem, you must first take the CTC adapter offline with a `VARY device OFFLINE` command before you can bring it online with a `VARY device ONLINE` command. You can then attempt to bring its inactive or quiesced systems into the rebuilt ring. If, after you restore all CTC adapters and systems to the ring, a system software problem disables the same CTC adapter again, issue `VARY GRS,QUIESCE` (if the system is not already quiesced) and `VARY GRS,PURGE` for the system with the software problem. Then, re-IPL that system with `GRS=JOIN` to rebuild the original ring.

For an example of rebuilding a disrupted ring manually, assume that you have the following ring consisting of systems A and B; CTC adapter 9C0 is the active communication link, and 3C0 is the alternate:



When you issue DISPLAY GRS from system A, you get the following information:

```
ISG020I  hh.mm.ss  GRS  STATUS  [id]
SYSTEM   STATE    COMM  SYSTEM   STATE    COMM
A        ACTIVE   B     B        ACTIVE   YES
LINK     STATUS   TARGET LINK    STATUS   TARGET
9C0      IN-USE   B     3C0     ALTERNATE B
```

If CTC adapter 9C0 fails and you again issue DISPLAY GRS from system A, you get the following:

```
ISG020I  hh.mm.ss  GRS  STATUS  [id]
SYSTEM   STATE    COMM  SYSTEM   STATE    COMM
A        INACTIVE B     B        INACTIVE YES
LINK     STATUS   TARGET LINK    STATUS   TARGET
9C0      DISABLED B     3C0     ALTERNATE B
```

Because there is an alternate CTC adapter (CTC 3C0), issuing the VARY GRS(ALL),RESTART command on either system A or B can rebuild the original ring. The messages issued from systems A and B when VARY GRS(ALL),RESTART is entered on system A are as follows:

```

                SYSTEM A                                SYSTEM B
VARY GRS(ALL),RESTART
ISG013I SYSTEM A - RESTARTED
                GLOBAL RESOURCE SERIALIZATION
ISG011I SYSTEM B - RESTARTING
                GLOBAL RESOURCE SERIALIZATION
ISG011I SYSTEM B - RESTARTING
                GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM B - RESTARTED
                GLOBAL RESOURCE SERIALIZATION
ISG013I SYSTEM B - RESTARTED
                GLOBAL RESOURCE SERIALIZATION
```

When you issue DISPLAY GRS from system A again, the response is:

```
ISG020I  hh.mm.ss  GRS  STATUS  [id]
SYSTEM   STATE    COMM  SYSTEM   STATE    COMM
A        ACTIVE   B     B        ACTIVE   YES
LINK     STATUS   TARGET LINK    STATUS   TARGET
9C0      DISABLED B     3C0     IN-USE   B
```

Note that, in this example, the original ring could not have been rebuilt after the failure if an alternate CTC adapter had not been available. When a CTC adapter fails in a two-system ring where there is no alternate, issuing the VARY GRS(ALL),RESTART command gives only one of the systems access to resources protected by global resource serialization. To bring the other system into the ring, you must either supply an alternate CTC adapter for the one that failed or fix the failed CTC adapter. Bring the alternate or fixed CTC adapter online with a VARY device ONLINE command. Then, to bring the inactive system into the ring, issue either VARY GRS(\*),RESTART on the inactive system or VARY GRS(sysname),RESTART for the inactive system on the active system.

## 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). 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/Extended Architecture (MVS/XA) system involve:

1. Operating the system itself -- that is, controlling the MVS/XA system software and most installation hardware (including processors, channel paths, and I/O devices)
2. Operating the MCS (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 and SYS1.PARMLIB parameters 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/XA 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 MCS console if it is specified as a device to the MVS configuration program (MVSCP) and defined as a console on a CONSOLE statement in the CONSOLxx member of SYS1.PARMLIB. 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 in MVSCP. If the console is allocated to a subsystem - CONSOLE DEVNUM(SUBSYSTEM) - there is no IODEVICE statement.

## Subsystems' Use of MCS Consoles

Many different devices can function as consoles in an MVS/XA system if they are specified as consoles in a CONSOLxx member of SYS1.PARMLIB. Some subsystems, such as JES3 and MVS/Operator Communication Control Facility (OCCF), can reserve consoles for their own use. You should familiarize yourself with subsystem consoles if your configuration includes them; some of them can affect MVS operations in important ways. For example, the pseudo-master console that OCCF uses has system command authority similar to that of a master console in the multiple-console configuration.

JES3 operators have an option of controlling a JES3 complex through a JES3 console or through MCS consoles that are attached to the global processor. The MCS consoles that control JES3 accept most JES3 commands and can receive all JES3 messages. JES3 directs message traffic to these MCS consoles by converting its messages into data that MCS recognizes. To learn how to initialize the JES3 consoles that use MCS, see *JES3 Initialization and Tuning*.

This chapter describes the various consoles that MVS, using MCS, allows in a multiple-console configuration. Details about the functions and command capabilities of subsystem consoles can be found in the subsystem books listed in the preface.

## Devices MVS/XA Supports as Consoles

Most of the consoles you use are probably display consoles, that is, they have a screen to display messages and a keyboard to enter commands. However, MVS supports some consoles that are capable of output-only or input-only functions. There are four categories of devices:

1. Display consoles (a 3278 display station, for example)  
  
The display console can be a combination input (operator-to system) and output (system-to-operator) device. It cannot, however, provide a permanent record of system communications.
2. Output-only consoles, which can be used for the hardcopy log (a 1403 or 3284 printer, for example)
3. Composite consoles made up of separate input and output devices (3505/3525 card read punch and the 3211 printer, for example)
4. Console printer-keyboards, namely, a 2740-1 communications terminal, which is a combination input and output device.

A **composite console** is made up of two devices, each having a unique device number. To assign a composite console, you must define both devices to MVSCP and specify both device numbers on the DEVNUM parameter in CONSOLxx member of SYS1.PARMLIB. Console operation is the same as with a combination input and output device. If the NIP console is a composite console, both devices must remain online throughout the IPL.



Figure 2-1 lists the devices that MVS supports as operator's consoles.

| Device                         | Input/Output Device | Composite Input Device | Composite Output Device | Output-Only Device | Description             |
|--------------------------------|---------------------|------------------------|-------------------------|--------------------|-------------------------|
| 1403                           |                     |                        | x                       | x                  | Printer                 |
| 2501-B1, B2                    |                     | x                      |                         |                    | Reader                  |
| 2540-1                         |                     | x                      |                         |                    | Reader                  |
| 2740-1                         | x                   |                        |                         |                    | Communications terminal |
| *3180-1                        | x                   |                        |                         | x                  | Display                 |
| 3203-5                         |                     |                        | x                       | x                  | Printer                 |
| 3211                           |                     |                        | x                       | x                  | Printer                 |
| 3270-x                         | x                   |                        |                         | x                  | Display                 |
| 3277-2                         | x                   |                        |                         | x                  | Display                 |
| *3278-2, 3, 4, 5               | x                   |                        |                         | x                  | Display                 |
| 3278-2A                        | x                   |                        |                         | x                  | Display                 |
| 3279-2A, 2C, 3A                | x                   |                        |                         | x                  | Display                 |
| *3279-2X, 2B, S2B, 3B, 3X, S3G | x                   |                        |                         | x                  | Display                 |
| 3284-1, 2                      |                     |                        | x                       | x                  | Printer                 |
| 3286-1, 2                      |                     |                        | x                       | x                  | Printer                 |
| *3290                          | x                   |                        |                         | x                  | Display                 |
| 3505-B1, B2                    |                     | x                      |                         |                    | Reader                  |
| 3525-5                         |                     | x                      |                         |                    | Punch                   |

Figure 2-1. Devices MVS/XA Supports as Consoles

\* Indicates that you can specify the device as a 3270-x console

If the devices are attached to a controller that supports the Read Partition Query Feature and the device also supports the feature, specify the 3278 and 3279 display units as 3270-x.

In this book, references to devices often do not mention model numbers. When you see a device referenced without a model number, assume the reference applies to all models of the device.

### Multiple-Console Configuration

Because it is difficult to control MVS/XA through a single console, 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 consists of up to 99 consoles; one console is the **master console**, and the rest are **secondary consoles**. The master console is always active. The secondary consoles can be active (being used) or inactive. A composite console represents two logical consoles out of the secondary set; however, MCS manages a composite console as if it were one console.

The master console is your principal means of communicating with the system. It is the only console that you can use to:

- Enter all operator commands
- Change the status of the hardcopy log
- Switch to a new master console

A secondary console is used for specific types of operator-system communication when it is more convenient to have a console located away from the processor. A secondary 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. A secondary console cannot enter all commands (see “System Command Groups”), 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 mode** that prevents the console from accepting certain 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. The purpose for having the different modes of consoles is to limit the number of consoles that operators can use to issue commands, and yet provide all operators the information that they need to do their work.

At IPL, the system looks to the CONSOLxx member of SYS1.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 consoles — the backup consoles assigned in case of a malfunction
- Hardcopy log — the console that produces the hardcopy log
- PFK definitions — the commands that console’s PFKs issue

Chapter 3 describes the parameters in CONSOLxx that define these and other characteristics and the commands that you can use to make changes to them.

## Features on Display Consoles

Most MVS/XA consoles are display consoles. A display console is an operator’s console that displays messages on a screen instead of printing them on paper. It lets you control the system more efficiently than a communications terminal or composite console because:

- Messages appear faster on the display console, letting you respond more quickly to the system’s needs.
- Special indicators on the display console highlight different types of messages. For example, the system highlights messages requiring action and keeps them on the display screen until you respond to them.

- The selector pen and the program function keyboard on the display console let you enter many commands at one time.

All display consoles can operate in full-capability, status display, or message stream mode. Each one has a keyboard to type 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.

*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.

Figure 2-2 summarizes the available features for the display consoles that IBM supports.

| Capability<br>Device                 | Selector<br>Pen | Audible<br>Alarm | Number of<br>Program<br>Function<br>Keys | Extended<br>High-<br>lighting | Color | Intensity |
|--------------------------------------|-----------------|------------------|--|-------------------------------|-------|-----------|
| 3180-1                               |                 | NO               | YES                                      | 24                            | YES   | NO YES    |
| 3277-2                               | YES*            | YES*             | 12*                                      | NO                            | NO    | YES       |
| 3278-2,3,4,5                         | YES*            | YES*             | 24*                                      | YES                           | NO    | YES       |
| 3278-2A                              | NO              | YES              | 12                                       | NO                            | NO    | YES       |
| 3279-2A,3A                           | YES*            | YES*             | 24*                                      | NO                            | YES   | NO        |
| 3279-2B,3B,3X,<br>S3G,2X,<br>and S2B | YES*            | YES*             | 24*                                      | YES                           | YES   | NO        |
| 3279-2C                              | NO              | YES              | 12                                       | NO                            | YES   | NO        |
| 3290                                 | NO              | YES              | 24                                       | YES                           | NO    | YES**     |

\*Optional feature

\*\*Intensity is simulated using an underscore

**Figure 2-2. Available Features of Display Consoles**

## Display Screen Areas

The operating mode of the console controls the appearance of a display screen. Figure 2-3 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, MCS uses the display area 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.

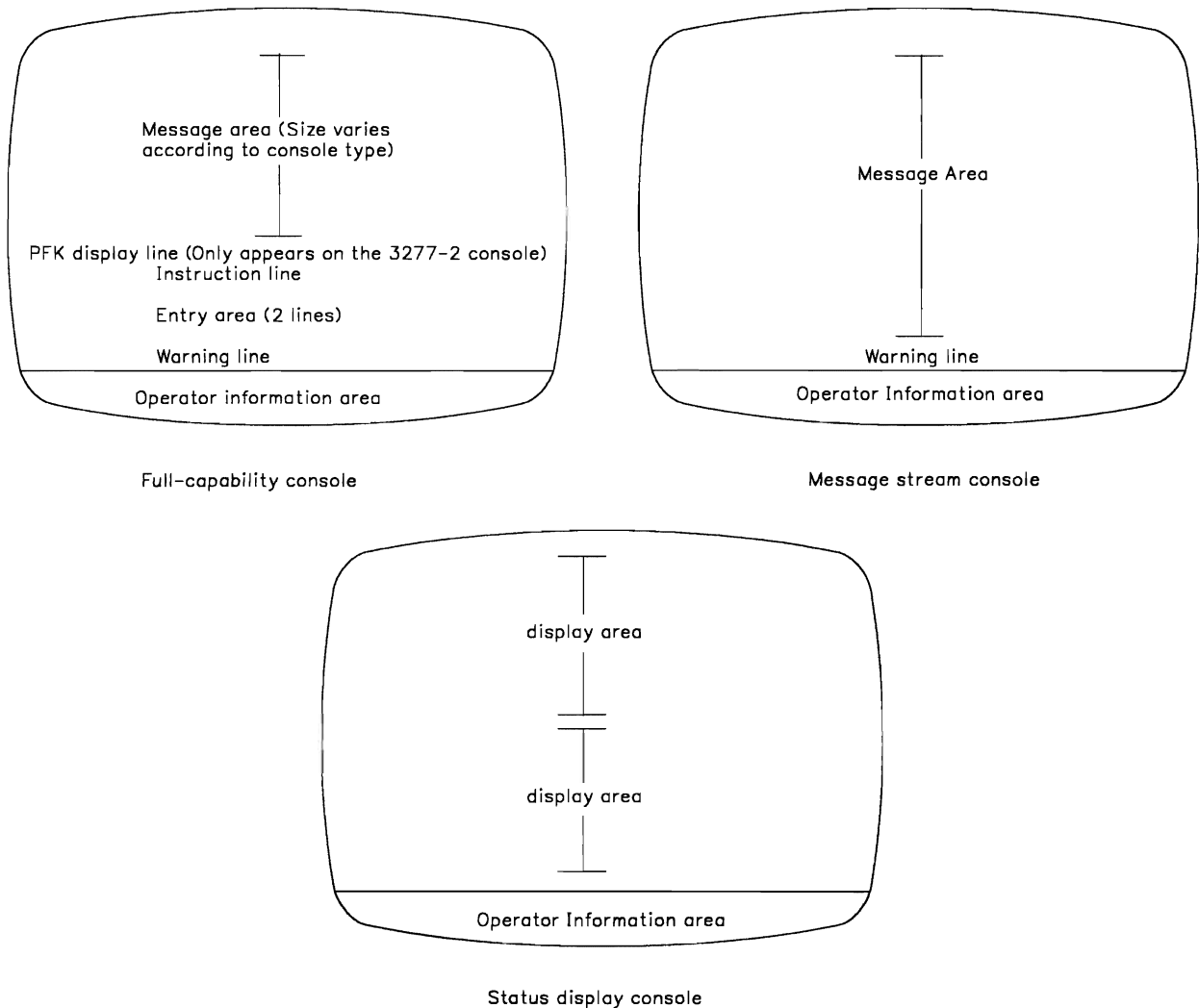
*Instruction Line:* This line contains console control messages. For example, if you make an error entering a CONTROL command, an error message appears in the instruction line.

*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, contains messages and symbols that keep you informed of the operational status of the terminal. It is not available on some terminals.

Figure 2-3 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 2-3. Comparison of the Display Screens of Full-Capability and Output-Only Display Consoles**

## 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/XA system and any program running under MVS/XA 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 appear in the format:

|   |
|---|
| <p><b>f message</b></p> <p>or</p> <p><b>hh.mm.ss sysname jobident f message</b></p> |
|---|

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 the next section, "Special Screen Characters."

**message**    Message identifier and text

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

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

**sysname** Name of system that issued the message

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

To add any combination of job identification, time stamp, and system name to all console messages, see “Controlling the Format of Messages” in Chapter 3.

## Specific Characteristics of Display Consoles

The following display consoles are described in this section:

- 3277 Model 2 display console
- 3278 and 3279 display consoles
- 3090 information panel
- 3180 model 1 display station

### 3277 Model 2 Display Console

The 3277 model 2 display console can be used as an operator console. It works in full-capability, message stream, or status display operating mode. Other MVS/XA consoles can be in the same system with it, and you can use it to control output-only consoles.

#### Console Characteristics

The 3277 model 2 console has a display screen and usually a keyboard. If a console does not have a keyboard, it can function only as an output-only console.

#### Screen Format

Figure 2-4 shows the display screen format. The screen has five functional areas:

- **Message Area:** The message area is 19 lines long; it contains system messages, operator commands, and status displays. Each message area line contains 78 visible character positions. (There are two system indicator positions preceding the first visible character position; these always appear blank on the screen). The first two visible positions contain the message line number (if message numbering is in effect). Positions three, four, and five contain the message-type indicator (\*,|,-,+,@). Positions 5 through 78 contain the message text. Position 5 can contain either a message type indicator or text. Messages longer than 78 characters continue on the next screen line, beginning in position 6. The following is a sample message area line:

```
01 IEE334I HALT EOD SUCCESSFUL
```

Messages appearing in this area are intensified (made brighter than the other messages on the screen) if they require action or a reply.

*Note:* If you request that the console display a time stamp or job id, the system truncates multi-line messages at screen position 78; they do not continue on the next screen line. To prevent truncation, you can choose not to display the time stamp and job id.

- **PFK Display Line:** This line contains PFK numbers (for selection with the selector pen). When you request the display, the line appears as follows:

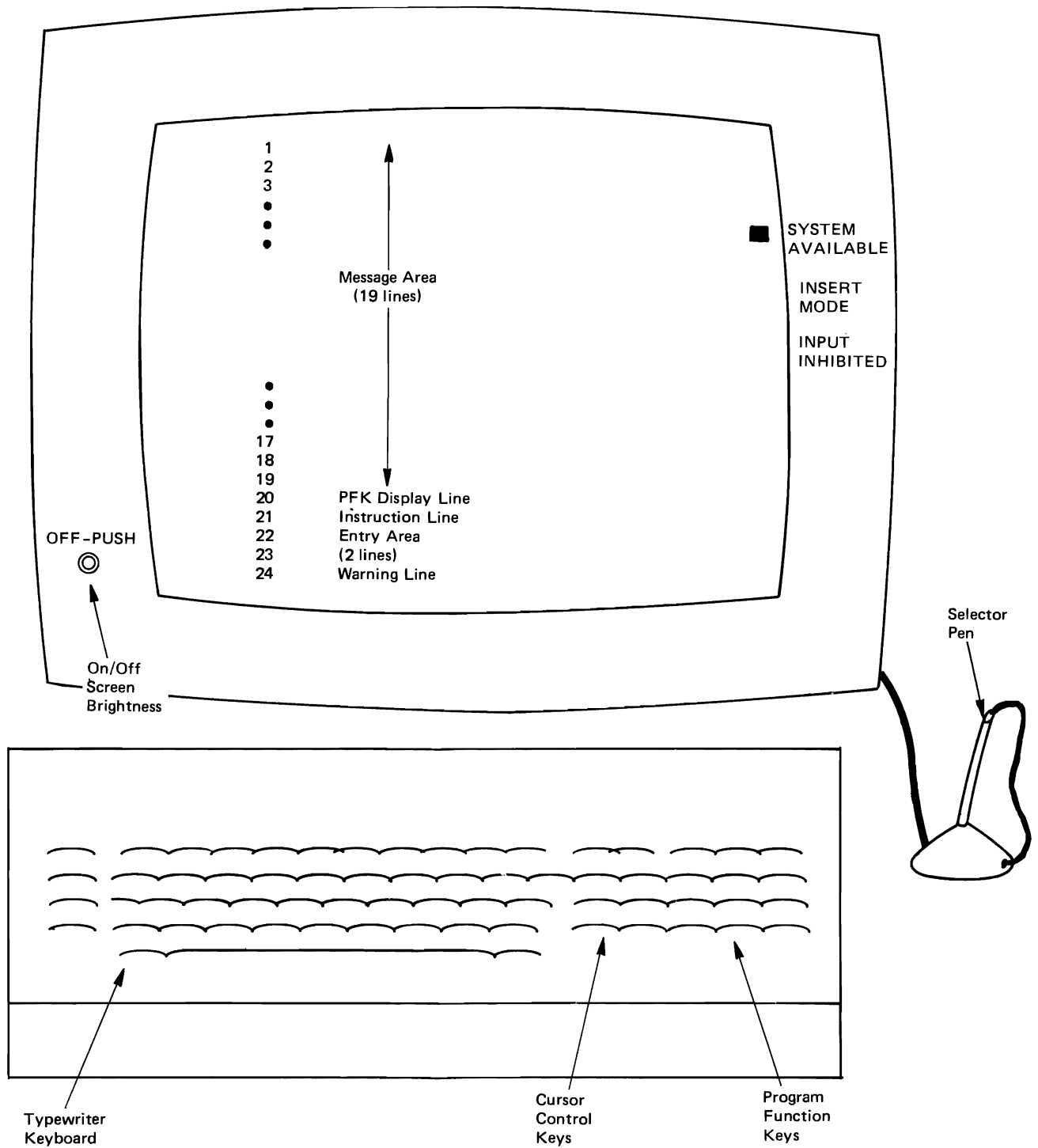
1 2 3 4 5 6 7 8 9 10 11 12

- **Instruction Line:** This line contains system messages that relate to console control. Messages that appear in this line are intensified, with the following exception:

IEE152I ENTER CANCEL D C,K

- **Entry Area:** These two screen lines, together with the cursor and the keyboard, you use to enter commands and reply to system messages. The first line of the entry area has 78 character positions (the first two positions contain system indicators and always appear blank on the screen). The second line has 80 character positions. The maximum length of a command is 126 characters.
- **Warning Line:** This line displays messages that warn you of conditions that could require action. Messages appearing in this line are intensified (made brighter than the other messages on the screen).





**Figure 2-4. 3277 Model 2 Display Console with Optional Features Attached**

## Special Screen Characters

The 3277 model 2 uses special screen characters to indicate the console's input mode and the status of certain screen messages.

*Input Mode:* The mode marker ( ■ ) appears on the right side of the screen next to one of the printed mode indicators: INSERT MODE, INPUT INHIBITED, or SYSTEM AVAILABLE. Insert mode, which you request with the INS MODE key, lets you correct typing errors by inserting extra letters within a line of text. After correcting the error, restore normal input mode by pressing the RESET key. INPUT INHIBITED indicates that the system is temporarily unable to receive input. Sometimes, you can restore normal system operation by pressing the RESET key. The system restores normal input mode after most processes that result in inhibited input. SYSTEM AVAILABLE indicates that the console is communicating with the system and can send or receive messages.

*Message Status:* The system uses five special screen characters (\*, |, -, +, and @) 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.

## The Cursor

The cursor appears on the screen as an underscore. You can move the cursor anywhere on the screen, but its position is meaningful only in the following locations:

- Any position on a nonaction message line, for deleting action messages
- The asterisk on an action message line, for deleting action messages
- The special indicators in a status display control line, for framing, holding, updating, or erasing the display
- The special indicators in the instruction line, for entering or canceling commands
- Any position in the entry area, for entering commands

*Cursor Control Keys:* Four basic cursor control keys are on the right side of the keyboard. Pressing one of these keys moves the cursor continuously in the indicated direction: up, down, left, or right. Cursor movement stops when you release the key. Other cursor control keys move the cursor as follows:

- The backspace key, on the right side of the top line of keys, moves the cursor to the left.
- The tab key moves the cursor forward to the next unprotected field (an unprotected field is one in which you can type data). Because the only unprotected field on the display console screen is the entry area, pressing this key moves the cursor to the first position in the entry area.
- The back-tab key moves the cursor back to the first position of an unprotected field. Pressing this key returns the cursor to the first data position in the entry area.
- The new line key moves the cursor to the first unprotected field in any line below the line the cursor is on. Pressing this key can move the cursor to the first position in the entry area.

## The Selector Pen

The selector pen is an optional feature of the 3277 model 2 display console. The selector pen is a light-sensitive device that identifies an area of the screen for system action. To use the pen, place it against the screen over the area that you are identifying and push on the pen slightly to activate the light-sensing mechanism.

You can place the selector pen anywhere on the screen, but its position is meaningful only in the following locations:

- On a nonaction message line, for deleting all messages above that line
- On an indicator in the control line of a status display, for framing, holding, updating or erasing the display
- On ENTER, CANCEL, or D C,K in the instruction line, for entering or canceling commands, or for requesting a display of CONTROL command operands
- On a key number in the PFK display line for entering commands

Placing the selector pen or cursor over screen locations not defined for detect actions causes the following message:

```
IEE164I ILLEGAL LIGHT PEN-CURSOR DETECT
```

In response to this message, move the selector pen to a valid location.

## The Program Function Keyboard

The program function keyboard, an optional feature of the 3277 model 2, consists of a group of 12 keys called PFKs. The keys are numbered PF1 through PF12. Use the PFKs as a shortcut for entering commands. Some of the PFKs on your console probably have the commands that IBM defined; others have commands that your system programmer defined. When you bring up the console, the PFK definitions are in place. Use the CONTROL command to change the definitions for the duration of the IPL. To see how your PFKs are defined, issue the DISPLAY PFK command. For more information on PFKs, see “Defining Commands in PFK Tables” and “Defining PFKs Using the CONTROL Command” in Chapter 3.

## Special Keys

**CLEAR Key:** If you press the CLEAR key, the screen is cleared and refreshed, except for the following areas:

- **Status Displays:** The CLEAR key removes status displays from the screen but does not restore display areas for use by other messages. To restore a display area, either request another display for the area or issue a CONTROL E,D,L =cca command, specifying the area in the L =cca operand. Dynamic status displays resume normal operation when the time interval in progress elapses. (See “Controlling System Status Displays” on page 3-42.)
- **PFK Display Line:** To restore this line, first issue a CONTROL E,PFK command and then issue a CONTROL D,PFK command (see “Entering Commands with the Selector Pen” in the next section.)

**TEST REQ Key:** This key tests the console; you do not use it for normal MVS/XA console functions.

*PA1 Key:* This key makes the system display, in the entry area, the last command entered on this console. You can then modify the command, if necessary, and re-enter it.

*PA2 Key:* This key performs the CANCEL action; it restores the screen and clears the entry area.

### The Audible Alarm

The audible alarm is an optional feature of the 3277 model 2. This alarm sounds for one second when:

- The IPL procedure or a VARY command initializes the console
- You make an error entering a CONTROL or MSGRT command
- You put the console in roll mode to delete messages
- The screen is full and another message is waiting to appear
- An action message appears on the screen

*Response:* If an alarm sounds while you are entering a CONTROL or MSGRT command, check the instruction line to see if the system issued an error message. If so, follow the procedures in the message explanation in *Message Library: System Messages*.

If an alarm sounds and the following message appears in the warning line:

```
IEE159E MESSAGE WAITING
```

delete some screen messages using the procedures described under “Deleting Messages from the Console Screen” in Chapter 3.

### Message Presentation

The 3277 presents messages with two degrees of intensity. You can control the contrast between the two message intensities with the brightness and contrast controls on the on/off switch.

The system intensifies the following message types:

- Error messages in the instruction line
- Roll and roll-deletable mode messages in the warning line
- Action messages

### 3278 and 3279 Display Consoles

The 3278 and 3279 display consoles can be used as operator consoles. The 3278 models 2, 2A, 3, 4 and 5 work in either full-capability or output-only operating mode. The 3279 models 2A, 2B, 2C, 3A, 3B, 3X, S3G, 2X, and S2B work in either full-capability or output-only mode. Other MVS/XA consoles can be in the same system with 3278 and 3279 consoles; you can use 3278 and 3279 consoles to control output-only consoles.

## Console Characteristics

The 3278 and 3279 consoles consist of a display screen and, usually, a keyboard. If a console does not have a keyboard, it functions as an output-only console.

### Screen Format

Figure 2-5 shows the display screen format. The screen has five functional areas:

- **Message Area:** The message area contains system messages, operator commands, and status displays. The visible portion of each message area line contains 78 character positions (there are two system indicator positions preceding the first visible character position; these always appear blank on the screen). The first two visible positions contain the message line number (if message numbering is in effect). Positions three, four, and five contain the message-type indicator (\*,|,-, +,@). Positions 5 through 78 contain the message text; messages longer than 78 characters continue on the next screen line, beginning in position 6. The following is a sample message area line:

```
01 IEE334I HALT EOD SUCCESSFUL
```

The message area size depends on the model and the console's mode. See Figure 3-12 on page 3-44 for message area sizes for the 3278 and 3279 display screens.

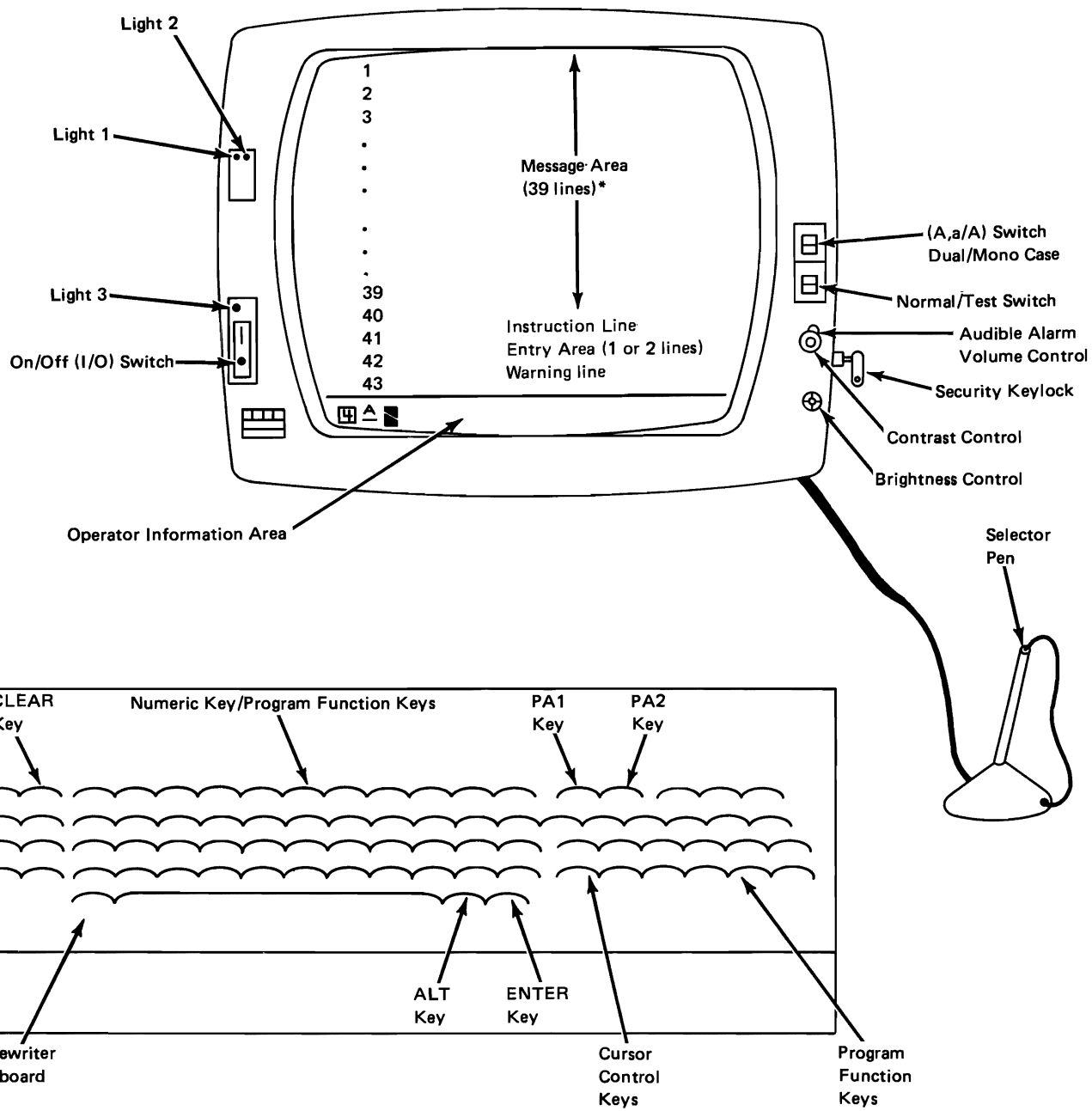


Figure 2-5. 3278 Model 4 Display Console with Optional Features Attached

- Instruction Line:** This line contains system messages that relate to console control. Messages that appear in this line are intensified, with the following exception:

```
IEE152I ENTER CANCEL D C,K
```

- **Entry Area:** The one or two screen lines, together with the cursor and the keyboard, you use to enter commands and to reply to system messages. Because the 3278 model 5 has a screen size of 132 characters, it has a one-line entry area. On other models, the first line of the entry area has 78 character positions. The second line has 80 character positions. The first two positions on the first line contain system indicators and always appear blank on the screen. The following example shows a DISPLAY UNIT command in the entry area:

D U,L=12A

- **Warning Line:** This line contains warning messages that usually require operator action. Messages appearing in this line might be highlighted, depending on MPF specifications. The following is an example of a warning line containing message IEE163I:
- IEE163I MODE=RD
- **Operator Information Area:** This line, the bottom-most line on the screen, is separated from the rest of the display area by a horizontal line. The information area contains operating and status messages about the operation of the console. The messages, made up of symbols, words, and numerals, fall into five categories and appear in five different places in the area, as follows:

|                                 |                                |           |                  |                |
|---------------------------------|--------------------------------|-----------|------------------|----------------|
| Readiness and System Connection | Do Not Enter (Input Inhibited) | Reminders | Shifts and Modes | Printer Status |
|---------------------------------|--------------------------------|-----------|------------------|----------------|

### Special Screen Characters

The 3278 and 3279 use many special screen characters in various combinations in the operator information area. All of these characters and their meanings are described in *IBM 3270 Information Display System 3278 Display Station Operator's Guide*. The most common character combinations are:

- X SYSTEM The system is temporarily unable to receive input. Sometimes, you can restore normal operation by pressing the RESET key. Usually, however, the system restores normal input mode.
- X ⊙ The system is busy processing your request.
- ^ The console is in insert mode. (You pressed the insert mode key.) You can correct any errors in the last command you entered and then restore normal operation by pressing the RESET key.

### The Cursor

The cursor appears on the screen as an underscore or a rectangular symbol. You can move the cursor anywhere on the screen, but its position is meaningful only in the following locations:

- Any position on a nonaction message line, for deleting nonaction messages
- The asterisk on an action message, for deleting action messages
- The special indicators in a status display control line, for holding, framing, updating, or erasing the display

- The special indicators in the instruction line, for entering or canceling commands
- Any position in the entry area, for entering commands

*Cursor Control Keys:* Four basic cursor control keys are on the right side of the keyboard. Pressing one of these keys moves the cursor continuously in the indicated direction: up, down, left, or right. Cursor movement stops when you release the key. Other cursor control keys move the cursor as follows:

- The backspace key, on the right side of the top line of keys, moves the cursor to the left.
- The tab key moves the cursor to the next unprotected field (an unprotected field is one in which you can type data). Because the only unprotected field on the display console screen is the entry area, pressing this key moves the cursor to the first position in the entry area.
- The back-tab key moves the cursor back to the first position of an unprotected field. Pressing this key returns the cursor to the first data position in the entry area.
- The new-line key moves the cursor to the first unprotected field in any line below the line the cursor is on. Pressing this key can move the cursor to the first position in the entry area.

*Notes:*

1. *On a 3278, the home key repositions the cursor to the input area. On a 3279, this key repositions the cursor to the first unprotected area.*
2. *If you press the tab, back-tab, or new-line keys on a 3279 models 2A, 2C, or 3A, the cursor might move to a field (other than the entry area) that contains data in red or green.*

## **The Selector Pen**

The selector pen is an optional feature of the 3278 models 2, 3, 4, and 5 and all models of the 3279. The selector pen is a light-sensitive device that identifies an area of the screen for system action. To use the pen, place it against the screen over the area that you are identifying and push on the pen slightly to activate the light-sensing mechanism.

You can place the selector pen anywhere on the screen, but its position is meaningful only in the following locations:

- On a nonaction message line, for deleting all messages above that line
- On an indicator in the control line of a status display, for framing, holding, updating, or erasing the display
- On ENTER, CANCEL, or D C,K in the instruction line, for entering or canceling commands, or for requesting a display of CONTROL command operands

Placing the selector pen over screen locations not defined for detect actions causes the following message in the instruction line:

```
IEE164I ILLEGAL LIGHT PEN-CURSOR DETECT
```

In response to this message, move the selector pen to a valid location.



## The Program Function Keyboard

The program function keyboard is an optional group of 24 keys (called PFKs) on the console keyboard. The keys are numbered PF1 through PF24. You must hold down the ALT key when pressing PF1 - PF12. Use the PFKs as a shortcut for entering commands. Some of the PFKs on your console probably have the commands that IBM defined; others have commands that your system programmer defined. When you bring up the console, the PFK definitions are in place. You can use the CONTROL command to change the definitions for the duration of the IPL. To see how your PFKs are defined, issue the DISPLAY PFK command. For more information on PFKs, see "Defining Commands in PFK Tables" and "Defining PFKs Using the CONTROL Command" in Chapter 3.

## Special Keys

*CLEAR/SYS REQUEST Keys:* If you press either of these keys, the system performs a cancel action to restore the screen. These keys remove status displays from the screen but do not restore display areas for use by other messages. To restore a display area, either request another display for the area or issue a CONTROL E,D,L=cca command, specifying the area in the L=cca operand. Dynamic status displays resume normal operation when the time interval in progress elapses. (See "Controlling System Status Displays" on page 3-42) On the 3278 models 3, 4, and 5, and the 3279 models 3A, 3B, 3X, and S3G, you must press the RESET key after pressing the CLEAR key to restore normal system operation. In addition, on these devices, pressing the CLEAR key makes the system write a SYS1.LOGREC record indicating that a unit check has occurred.

*TEST Key:* This key tests the console; you do not use it for normal console functions. If you press it accidentally, immediately press it again and then press the CLEAR key. If this action does not restore the console, a console switch has taken place. Using an active console, enter a VARY command to put the inoperative device back in console state. If the lost console was the only active full-capability console, follow the procedure for responding to a no-consoles condition described in the preceding section.

*PA1 Key:* This key makes the system display, in the entry area, the last command entered on this console. You can then modify the command, if necessary, and re-enter it.

*PA2 Key:* This key performs the CANCEL action; it restores the screen and clears the entry area.

*ALT Key:* Hold down this key to activate alternate functions (those functions printed on the front of the keys) when pressing keys that have alternate functions. The alternate functions are: CLEAR, ERASE INPUT, PF1-PF12, PA1, and PA2.

*ERASE INPUT Key:* For a 3278, this key is used to erase the entry area. For the 3279 models 2A, 2C, and 3A, when the base color switch is in the four-color position, pressing this key causes all data displayed in green or red to be deleted. Note that you must also reposition the cursor; the cursor repositions itself to the first unprotected area. If the erase input key is pressed, you can refresh the screen by pressing the PA2 (CANCEL) key. If the console is in roll or roll-deletable mode, the screen is refreshed when the rolling action is performed.

## The Audible Alarm

The audible alarm is an optional feature of the 3278 and 3279. This alarm sounds for one second when:

- The IPL procedure or a VARY command initializes the console
- You enter data on the last line in the next to last character position (79 on an 80 character line)
- You make an error entering a CONTROL or MSGRT command
- You put the console in roll mode to delete messages
- The screen is full and another message is waiting to appear
- An action message appears on the screen

*Response:* If an alarm sounds while you are entering a CONTROL or MSGRT command, check the instruction line to see if the system issued an error message. If so, follow the procedures in the message explanation in *Message Library: System Messages*.

If an alarm sounds and the following message appears in the warning line:

```
IEE159E MESSAGE WAITING
```

delete some screen messages using the procedures described under “Deleting Messages from the Console Screen” in Chapter 3.

## Message Presentation

The 3278 and 3279 display different types of messages and console fields in different colors and with different forms of highlighting.

*On the 3278:* The system has defaults for highlighting different types of messages on the 3278. On this console:

- Urgent attention messages appear in high intensity. Urgent attention messages include:
  - Messages with a descriptor code of 1 or 11
  - Message IEE159E (“MESSAGE WAITING”), which appears on the right side of the warning line
  - The system termination message, which appears in the warning line
- Immediate action messages appear in high intensity. Immediate action messages include:
  - Messages with a descriptor code of 2
  - Messages requiring a reply (WTOR messages)
  - Error messages appearing in the instruction line

- Console operation messages and fields appear in normal intensity. These messages and fields include:
  - Selector pen detectable fields such as:
    - ENTER, CANCEL, and D C,K in the instruction line
    - F or E in the control line of an out-of-line display
    - U, PT, or H in the control line of a TRACK display
    - C in the control line of an inline status display
  - Console mode messages appearing on the left side of the warning line:
    - IEE163I MODE=R
    - IEE163I MODE=RD
    - IEE161I WARNING - CON=N,DEL=Y
  - Message IEE160I (“UNVIEWABLE MESSAGE”), which appears on the right side of the warning line
- The following messages and fields appear in normal intensity:
  - Eventual action messages (messages with a descriptor code of 3)
  - General system messages (with descriptor codes other than 1, 2, 3, or 11)
  - Problem program messages (with descriptor codes other than 1, 2, 3, or 11)
  - Entry area
  - Out-of-line data lines
  - Message IEE152I id in the instruction line
  - Selector pen detectable fields. These fields include:
    - ENTER, CANCEL, and D C,K in the instruction line
    - F or E in the control line of an out-of-line display
    - U, PT, or H in the control line of a TRACK display
    - C in the control line of an inline status display
  - General messages in the warning line
  - Out-of-line control lines
  - Out-of-line label lines

The 3278 uses several additional forms of highlighting to display different message types and console fields. The forms of message highlighting are:

- Normal display (messages appear in green on a dark screen)
- Reverse video (messages appear in black on a green screen)
- Blinking display (messages blink on the screen)
- Underscoring (messages appear underscored)

Normally on the 3278, the following messages appear in high intensity in a blinking display:

- Urgent attention messages on the right side of the warning line, such as message IEE159E (“MESSAGE WAITING”)

Although these are the system defaults for highlighting different types of messages on the 3278, different types of highlighting for different types of messages can be defined in an MPFLSTxx member of SYS1.PARMLIB. Your installation can, therefore, set up its own highlighting display defaults for the 3278 and, during IPL, activate the MPFLSTxx member. To activate an MPFLSTxx member, code the MPF parameter on the INIT statement in the CONSOLxx member of SYS1.PARMLIB, or issue the SET MPF command. For information on defining an MPFLSTxx member, see *Initialization and Tuning*.

*On the 3279 models 2A, 2C, and 3A:* The 3279 models 2A, 2C, and 3A use four colors to display different message types and console fields. MPF color specifications will not affect these four-color models. On these consoles:

- Urgent attention messages appear in red.
- Immediate action messages appear in white.
- Console operation messages and fields appear in blue.
- General message traffic appears in green.

*On the 3279 models 2B, 3B, 2X, S3G, 3X, and S2B:* The 3279 models 2B, 3B, 2X, S3G, 3X, and S2B use seven colors and several forms of highlighting to display different message types and console fields. The seven message colors are red, white, green, blue, pink, yellow, and turquoise. The forms of message highlighting are:

- Normal display (messages appear in color on a black screen)
- Reverse video (messages appear in black on a colored screen)
- Blinking display (messages blink on the screen)
- Underscoring (messages appear underscored)

Normally, on these 3279 models:

- The following messages appear in red in a blinking display:
  - Urgent attention messages on the right side of the warning line, such as message IEE159E (“MESSAGE WAITING”)
- The following messages appear in red in a normal display:
  - Urgent attention messages that are not on the right side of the warning line.
  - Messages with a descriptor code of 1 or 11
  - The system termination message
  - Messages IEE170E and IEE171E
- The following messages appear in white in a normal display:
  - Immediate action messages.
  - Error messages in the instruction line

- The following messages and fields appear in green in a normal display:
  - Eventual action messages (messages with a descriptor code of 3)
  - General system messages (with descriptor codes other than 1, 2, 3, or 11)
  - Problem program messages (with descriptor codes other than 1, 2, 3, or 11)
  - Entry area
  - Out-of-line data lines
  - Message IEE152I
- The following messages appear in blue in a normal display:
  - Selector pen detectable fields.
  - General messages in the warning line
- The following messages appear in turquoise in a normal display:
  - Out-of-line control lines
  - Out-of-line label lines

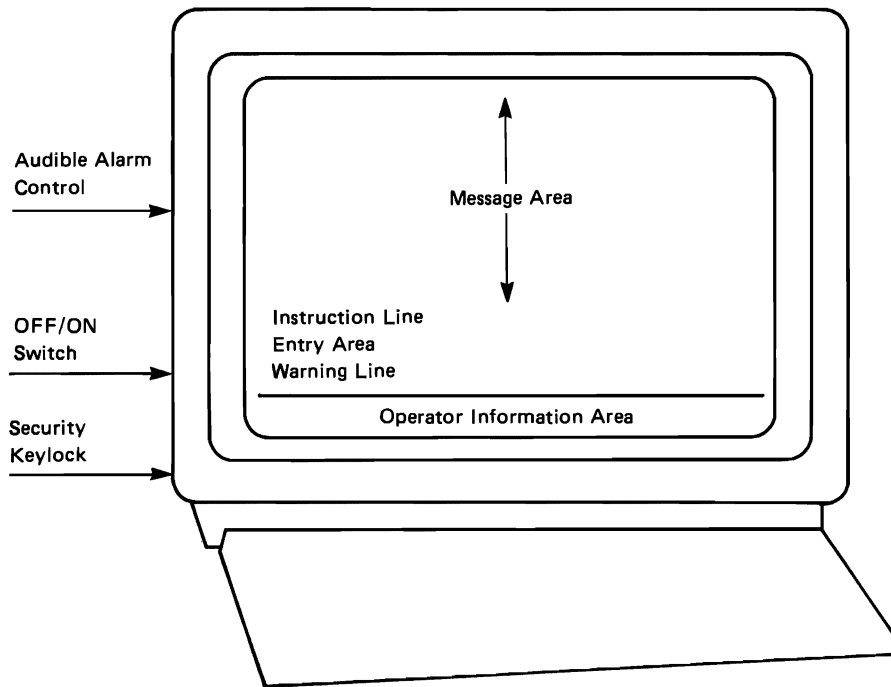
Although these are the system defaults for coloring and highlighting different types of messages on the 3279 models 2B, 3B, 2X, S2B, 3X, and S3G, different colors and types of highlighting for different types of messages can be defined in an MPFLSTxx member of SYS1.PARMLIB. To define an MPFLSTxx member, code the MPF parameter on the INIT statement in the CONSOLxx member of SYS1.PARMLIB, or issue the SET MPF command. For information on defining an MPFLSTxx member, see *Initialization and Tuning*.

## 3290 Information Panel

The 3290 information panel can serve as one large-screen logical console or can be two or four logical consoles. (A logical console has been defined to MCS at SYSGEN as a separate console.) In this one device, you have these options:

- One console, capable of displaying 50 messages in large characters or 62 messages in smaller characters, each message on a single line.
- Two consoles, with the 3290 screen divided either vertically to display 62 lines of messages, or horizontally to display 31 messages on each logical console.
- Four consoles, each able to display 31 lines of messages. If you divide the message traffic among them, you have a console cluster on one physical device.

When the 3290 serves as two consoles or four consoles, it is in **multi-console mode**. All logical consoles in multi-console mode operate independently of each other (and are defined separately to MCS) but depend on the same IBM 3274 control unit. Each logical console can work in full-capability, status display, or message stream operating mode. Other MVS/XA consoles can be used in the same system as the 3290. Figure 2-6 shows the 3290 as a single console.



**Figure 2-6. 3290 Information Panel**

### Console Characteristics

The 3290 display screen is a flat gas panel, which permits the vivid display of orange characters against a black background or, in reverse video, black characters on an orange background.

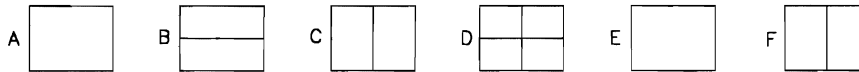
The keyboard has 104 keys, including 24 program function keys. An optional keypad can be customized to contain 24 program function keys that you can use instead of the 24 on the keyboard.

Before you can use the information panel, the system programmer must customize it. This customizing determines:

- The console configuration options (up to six)
- The console configuration option that appears when you power on
- The highlighting capabilities

*Note:* To prevent message loss, the system programmer must turn on bit 7 in response to the 3290 option (question 173) during customization of the 3274. *3270 Information Display System - 3274 Control Unit Planning, Setup, and Customizing Guide* helps the system programmer initialize the 3290.

The six options for your 3290 might be:



where:

- A is a single console with screen size 50 x 106.
- B is two consoles with screen sizes 31 x 160.
- C is two consoles with screens sizes 62 x 80.
- D is four consoles with screen sizes 31 x 80.
- E is a single console with screen size 43 x 80 (to be used to support some applications).
- F is two consoles with one screen size 62 x 80 (to be used as an MCS console) and another screen size 43 x 80 (to be used as a JES3 console).

You can make changes to these options by entering setup mode, as explained later in this section.

### Using the Two-Part or Four-Part Console

When the 3290 is in multi-console mode, it looks different from the consoles that you are used to, but you relate to each logical console the same as you relate to the 3278. Figure 2-7 shows the 3290 as four consoles. Because the upper right console screen contains the cursor, it is the only logical console currently able to accept your commands. (Messages, of course, continue to appear on all console screens.) The keyboard, including the PF keys, and the indicators and symbols in the operator information area relate to this screen. You can use this logical console to perform all the tasks of running an MVS/XA system.

If you want to expand this console screen to fill the entire 3290 screen, press the ZOOM key. The screen enlarges and the messages appear in the largest allowable character size and with optimum spacing. In zoom mode, only one logical screen is visible; other screens are erased but continue to interact with the system. If you stay in zoom mode too long, messages on other logical console screens roll off unnoticed. To leave zoom mode, press the ZOOM key again.

You can make another console able to receive commands by pressing the JUMP SCREEN key. The cursor moves from the current cursor position to the cursor position on the next clockwise screen.

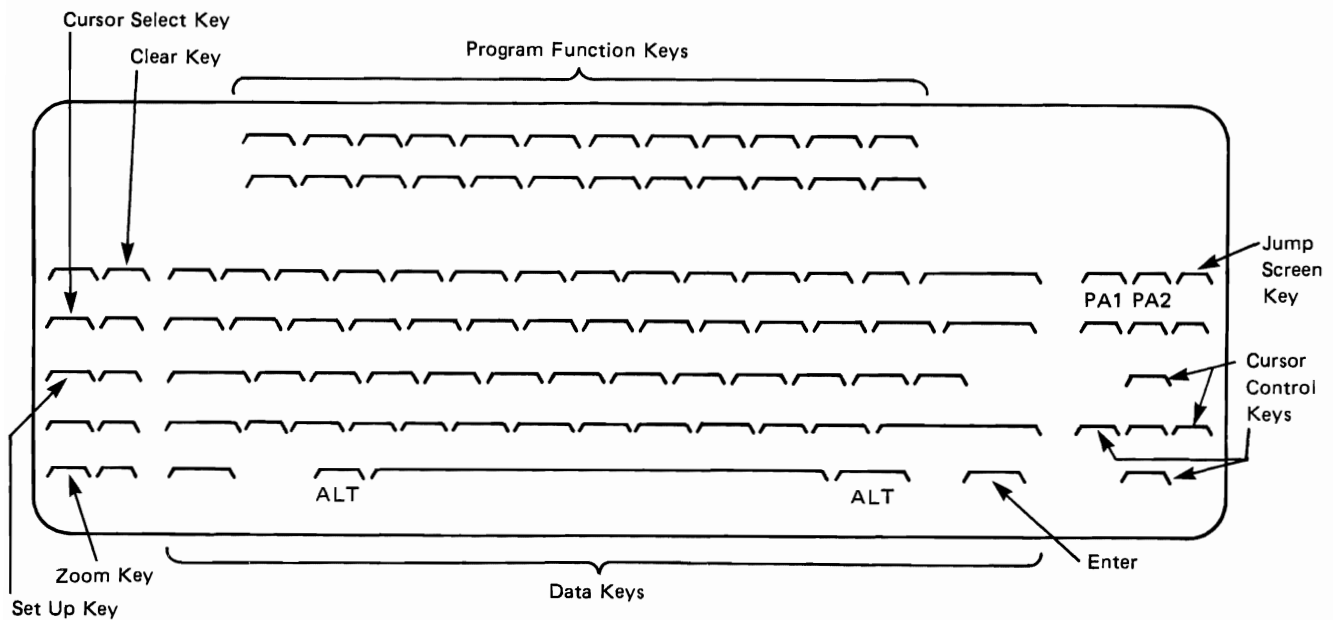
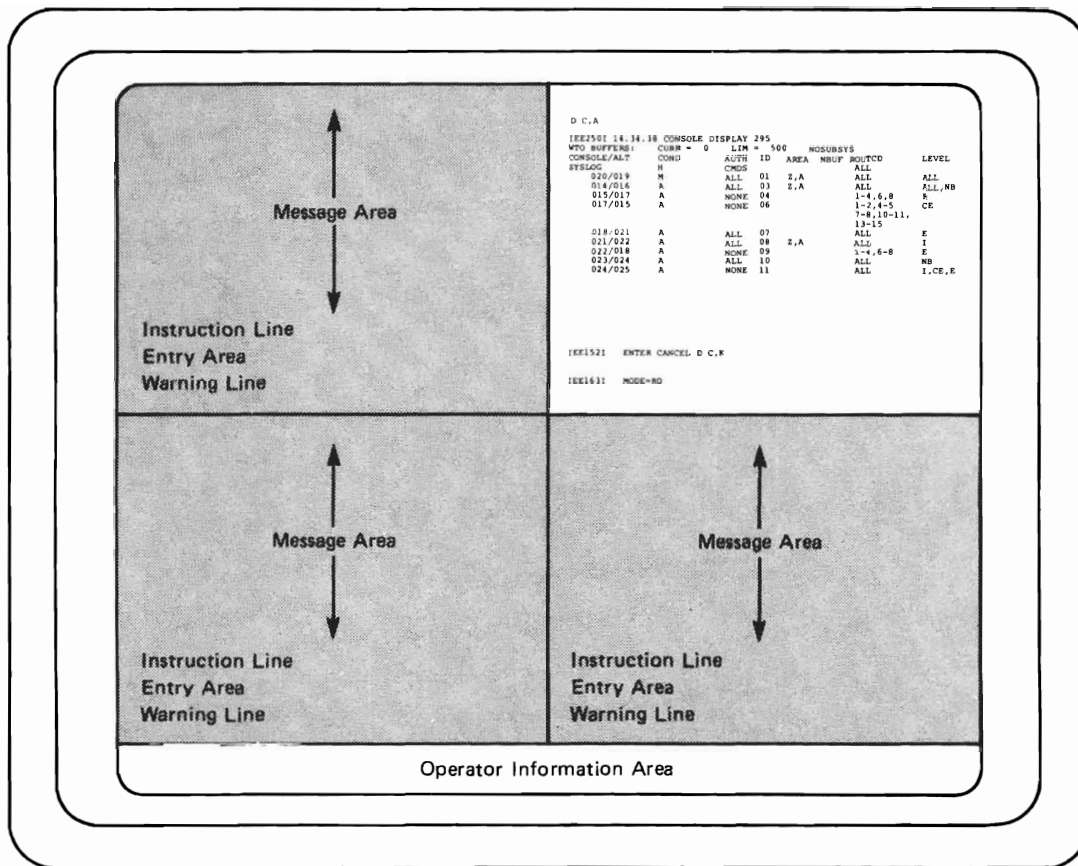


Figure 2-7. 3290 Four-Part Console with Keyboard



## Screen Format

There are five functional areas on the 3290 display screen; where they are located depends on how your 3290 is set up. To understand the different display screen formats, compare the single console screen in Figure 2-6 with the four-part console screen in Figure 2-7. Each logical console screen has a message area, an instruction line, an entry area, and a warning line. The operator information area relates to the logical console that contains the cursor.

*Message Area:* The message area contains system messages, operator commands, and status displays.

*Instruction Line:* This line contains syntax error messages in response to the commands you issue.

*Entry Area:* The entry area contains commands and replies to system messages. Any screen size that has a width of over 126 characters has a one-line entry area. Otherwise, the entry area has two lines.

*Warning Line:* This line contains informational messages and warning messages that might require operator action.

*Operator Information Area:* The operator information area is the bottom line of your display panel. A horizontal line separates this area from the display space. The operator information area displays operation and status indicators that are associated with your 3290 operation. To understand the meaning of the symbols in this area, see *IBM 3290 Information Panel Description and Reference*.

## The Program Function Keyboard

On the 3290 information panel, the program function keyboard is a group of 24 keys at the top of the keyboard or on a separate keypad. In MVS/XA console mode, you can define up to 24 PFKs for each logical console. You initialize these keys the same as you would the PFKs on a 3278 or 3279.

## Special Keys

Many keys on the 3290 are similar to keys on the 3278:

*CLEAR Key:* If you press the CLEAR key, the console screen that contains the cursor is cleared. To restore this screen and restore normal system operation, press the RESET key. Pressing the CLEAR key makes the system write a SYS1.LOGREC record indicating that a unit check has occurred.

*TEST Key:* This key tests the console; you do not use it for normal console functions. If you press it accidentally, wait until the word "TEST" appears in the operator information area, press the TEST key again, and then press the CLEAR key. If the console screen is still not displaying messages, use another active MCS console to vary the device back to console state.

*PA1 Key:* This key makes the system display, in the entry area, the last command entered on this logical console. You can then modify the command, if necessary, and re-enter it.

*PA2 Key:* This key performs the CANCEL action; restores the screen and clears the entry area of the logical console.

**CURSOR SELECT Key:** You can use the key labeled 'Cursr Sel' to select items in selection fields that appear in the operator information area. You can also use it in selector pen fields.

Because of the multi-screen capability, there are additional keys, such as:

**JUMP SCREEN Key:** When you press the key labeled 'JmpSc' while holding down either ALT key, the cursor jumps to the current cursor position of the next logical console screen, moving in a clockwise direction. This key works only when you have multiple logical consoles.

**ZOOM Key:** When you press  $a \rightarrow \bar{a}$ , the 3290 enters or leaves zoom mode. Only the logical console screen that contains the cursor is displayed; the remaining logical console screens of the screen division disappear. When the 3290 leaves zoom mode, all logical consoles reappear in their original size and spacing. This key works only when you are in mult-console mode.

### The Audible Alarm

The audible alarm for the 3290 information panel sounds for the same reasons as the audible alarm on the 3278 and 3279.

### Message Presentation

The 3290 is a monochrome device that has several forms of highlighting to display different message types. The message processing facility (MPF) regulates these options:

- Reverse video (messages appear in black on an orange background)
- Blinking display (messages blink on the screen)
- Underscoring (messages appear with each character separately underscored)

Because it can display characters in only one intensity, the 3290 underscores with a bold line the same messages that the 3278 intensifies. (See "Message Presentation" for the 3278 and 3279 display consoles earlier in this Chapter.)

Although these are the system defaults for highlighting different types of messages on the 3290, different types of highlighting for different types of messages can be defined in an MPFLSTxx member of SYS1.PARMLIB. To define an MPFLSTxx member, code the MPF parameter on the INIT statement in the CONSOLxx member of SYS1.PARMLIB, or issue the SET MPF command. For information on defining an MPFLSTxx member, see *Initialization and Tuning*.

**Reverse Video:** Because there is a limit to the number of light points that can be on at one time, if MPF requests that a message appear in reverse video, the 3290 displays it in this way:

- For system failure (descriptor code 1), immediate action (descriptor codes 2), and critical eventual action (descriptor code 11) messages, only the action character (@ or \*) that precedes the message is in reverse video.
- The text and characters that appear in the instruction line, the entry area, and the warning line appear in reverse video.
- Selector pen fields appear in reverse video.

For other types of messages and fields, the reverse video request is ignored.

## Changing to Another Console Option

When you power-on the console, the default console option appears. If you want to change the screen division to another option, you must vary all the 3290 logical consoles offline and enter setup mode.

- Enter setup level 1 by pressing the SETUP key while holding down the ALT key.
- Enter setup level 2 by moving the cursor under the diamond at the far right in the operator information area and pressing the CURSOR SELECT key. In setup level 2, a selection field appears in the operator information field. Letters identify the screen division options that were initialized by the system programmer for your 3290.

A  B  C  D  E  F

- Move the cursor under the option you select in the selection field and press the CURSOR SELECT key.
- Exit setup level 2 by pressing the SETUP key while holding the ALT key.
- Vary the logical consoles online.

The 3290 screen division you selected appears and will remain until you turn the 3290 off or repeat the preceding process to change the console option.

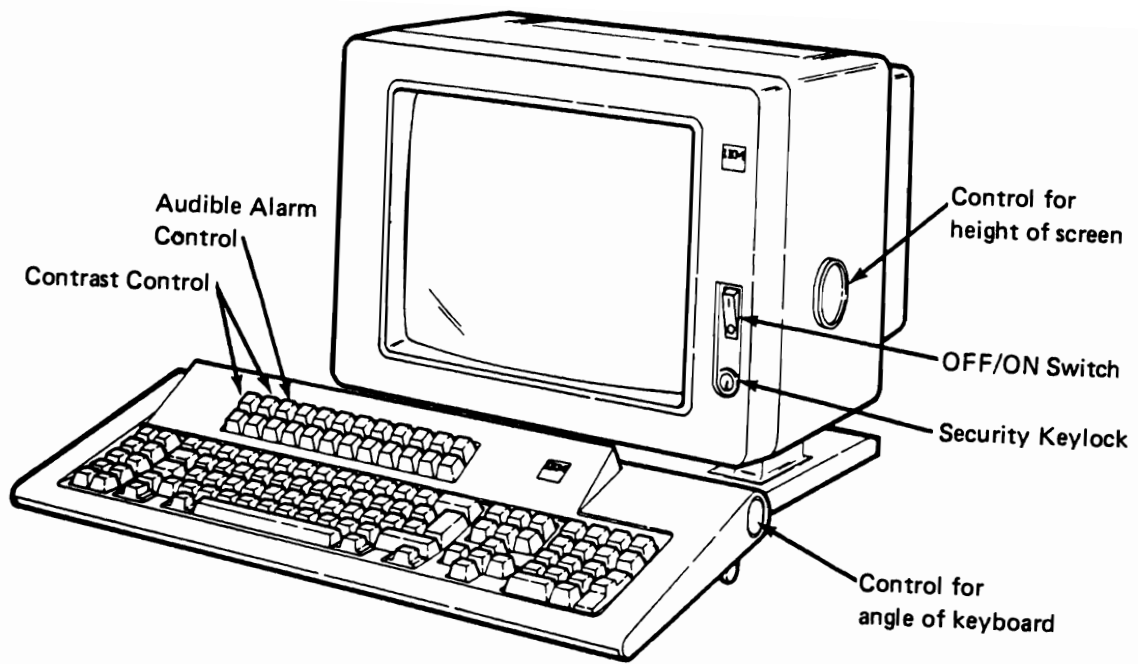
## 3180 Model 1 Display Station

The 3180 model 1 display station can be used as an operator console. It works in either full-capability or output-only operating mode. Other MVS/XA consoles can be in the same system with 3180 model 1 consoles.

### Console Characteristics

The 3180 display station, as shown in Figure 2-8, consists of a monochrome display screen and a movable keyboard. You can raise or lower the display screen and tilt it forward or backward. You can also adjust the angle of the keyboard. The 3180 model 1 display station is similar to the 3278 display consoles, with some exceptions:

- A security keylock in the locked position prevents input from the keyboard and blanks the screen.
- You can define any one of four screen formats on the 3180. See Figure 3-12 on page 3-44 for message area sizes for the 3180 display screen.
- The display screen automatically blanks, except for the cursor, and operator information area, after 10 minutes of keyboard and display station inactivity. You can restore the screen by pressing any key on the keyboard.



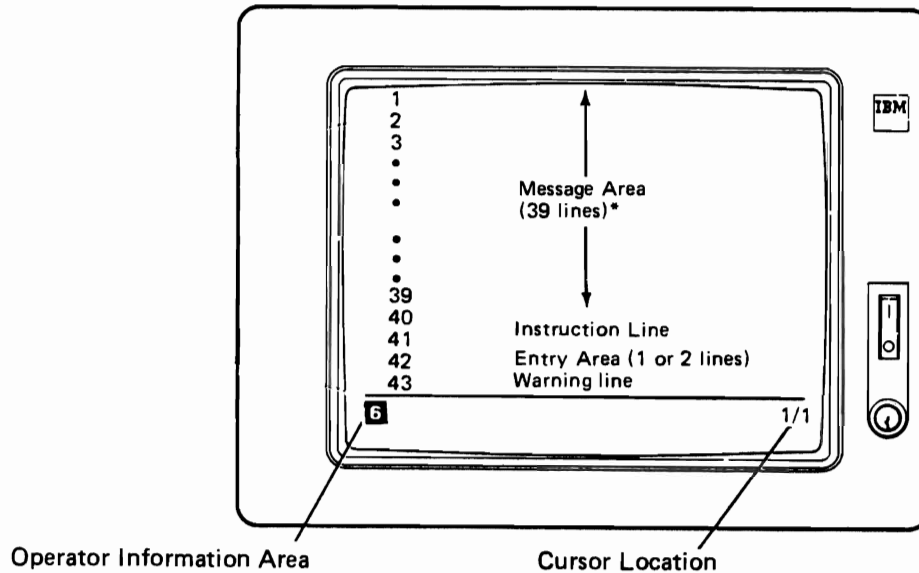
**Figure 2-8. 3180 Model 1 Display Station**

### **Screen Format**

The screen on the 3180 model 1 has the same five functional areas as the 3278 display console:

- Message area
- Instruction line
- Entry area
- Warning line
- Operator information area

The following illustration shows the screen on the 3180 model 1 defined to have 43 lines of 80 characters each.



### Special Screen Characters

The 3180 model 1 uses many special screen characters in various combinations in the operator information area. All of these characters and their meanings are described in *IBM 3180 Display Station User's Guide*. The most common character combinations are:

- |          |   |   |
|----------|---|---|
| X SYSTEM |   | The system is temporarily unable to receive input. Sometimes, you can restore normal operation by pressing the RESET key. Usually, however, the system restores normal input mode.        |
| X        | Ⓢ | The system is busy processing your request.   |
| ^        |   | The console is in insert mode. (You pressed the insert mode key.) You can correct any errors in the last command you entered and then restore normal operation by pressing the RESET key. |

### The Cursor

The cursor appears on the screen as an underscore or a rectangular symbol, either blinking or not blinking. You use PFKs (with the Set Up key) to control the shape of the cursor on the screen, as well as its blinking and the intensity of its light.

Functions of the cursor and the cursor control keys on the 3180 are the same as those on the 3278.

## The Program Function Keyboard

The program function keyboard is a group of 24 keys at the top of the keyboard. You can define these PFKs just as you would the PFKs on a 3278. You can use these PFKs to regulate some of the features on the display station, such as the alarm and the cursor.

You can use the RECORD key and a PFK to save a command that you use often. When you want to use the command, use the PLAY key and the PFK to write the command in the entry area and, optionally, enter it.

## Special Keys

Figure 2-9 shows the keyboard of the 3180 and identifies the location of some of the keys.

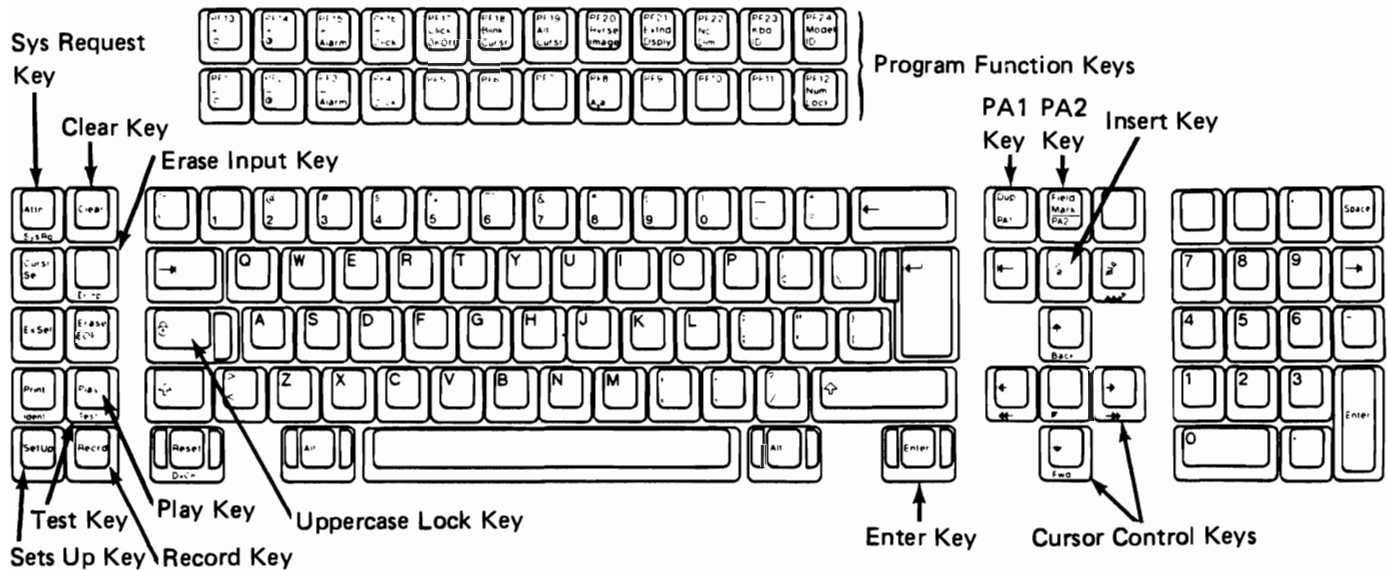


Figure 2-9. Keyboard of the 3180 Model 1

**CLEAR/SYS REQUEST Keys:** If you press either of these keys, the system performs a cancel action to restore the screen. These keys remove status displays from the screen but do not restore display areas for use by other messages. To restore a display area for use by other messages, either request another display for the area or issue a CONTROL E,D,L = cca command, specifying the area in the L = cca operand. Dynamic status displays resume normal operation when the time interval in progress elapses. (See “Controlling System Status Displays” on page 3-42.)

**TEST Key:** This key tests the console; you do not use it for normal console functions. If you press it accidentally, immediately press it again and then press the Clear key. If this action does not restore the console, a console switch has taken place. Using an active console, enter a VARY command to put the inoperative console back in console state. If the lost console was the only active full-capability console, follow the procedure described in “Responding to a No-Consoles Condition” later in this section.

*PA1 Key:* This key makes the system display, in the entry area, the last command entered on this console. You can then modify the command, if necessary, and re-enter it.

*PA2 Key:* This key performs the cancel action; it restores the screen and clears the entry area.

*ALT Key:* Hold down this key to activate alternate functions (those functions printed on the front of the keys) when pressing keys that have alternate functions.

*ERASE INPUT Key:* Use this key to erase the entry area. If the erase input key is pressed, you can refresh the screen by pressing the PA2 key. If the console is in roll or roll-deletable mode, the screen is refreshed when the rolling action is performed.

For console operations, you will not need the keys at the right side of the keyboard. These keys exist for terminal applications other than MCS console operations.

### **The Audible Alarm**

The audible alarm for the 3180 sounds for the same reasons as the audible alarm on the 3278.

### **Message Presentation**

The 3180 is a monochrome device that has several forms of highlighting to display different message types. The message processing facility (MPF) regulates these options: The forms of message highlighting are:

- Normal display (messages appear in green on a dark screen)
- Reverse video (messages appear in black on a green screen)
- Blinking display (messages blink on the screen)
- Underscoring (messages appear underscored)

The system has defaults for highlighting different types of messages on the 3180:

- Urgent attention messages appear in high intensity. Urgent attention messages include:
  - Messages with a descriptor code of 1 or 11
  - Message IEE159E (“MESSAGE WAITING”), which appears on the right side of the warning line
  - The system termination message, which appears in the warning line
- Immediate action messages appear in high intensity. Immediate action messages include:
  - Messages with a descriptor code of 2
  - Messages requiring a reply (WTOR messages)
  - Error messages appearing in the instruction line

- Console operation messages and fields appear in normal intensity. These messages and fields include:
  - Cursor-detectable fields such as:
    - ENTER, CANCEL, and D C,K in the instruction line
    - F or E in the control line of an out-of-line display
    - U, PT, or H in the control line of a TRACK display
    - C in the control line of an inline status display
  - Console mode messages appearing on the left side of the warning line:
    - IEE163I MODE = R
    - IEE163I MODE = RD
    - IEE161I WARNING - CON = N,DEL = Y
  - Message IEE160I (“UNVIEWABLE MESSAGE”), which appears on the right side of the warning line
- The following messages and fields appear in normal intensity:
  - Eventual action messages (messages with a descriptor code of 3)
  - General system messages (with descriptor codes other than 1, 2, 3, or 11)
  - Problem program messages (with descriptor codes other than 1, 2, 3, or 11)
  - Entry area
  - Out-of-line data lines
  - Message IEE152I id in the instruction line
  - General messages in the warning line
  - Out-of-line control lines
  - Out-of-line label lines

Urgent attention messages on the right side of the warning line, such as message IEE159E (“MESSAGE WAITING”), appear in high intensity in a blinking display.

Although these are the system defaults for highlighting different types of messages on the 3180, your installation can define different types of highlighting for different types of messages in an MPFLSTxx member of SYS1.PARMLIB. This action sets up your own highlighting display defaults for the 3180, and the system can activate the MPFLSTxx member during IPL. To activate an MPFLSTxx member, code the MPF parameter on the INIT statement in the CONSOLxx member of SYS1.PARMLIB or issue the SET MPF command. For information on defining an MPFLSTxx member for the duration of the IPL, see *Initialization and Tuning*.



## Characteristics of 2740 Communications Terminal

The IBM 2740 Communications Terminal can function as an MCS console when it connects to an IBM 2701 Data Adapter Unit. This section describes some of the operations for the 2740, which are very different from the operations for a display console. To use this console with MVS/XA, the system must include the BTAM/SP program product.

For complete information on the physical characteristics and operation of the 2740, see the *IBM 2740/2741 Communications Terminal Operator's Guide*.

### Entering Commands or Responding to Messages on the 2740

To enter commands or to respond to messages from the system:

1. Set the Local/Communicate switch to COM (Communicate mode). The Standby light (S light) must come on before you proceed.
2. Press the Bid key. Once your bid enters the communications line, the Transmit light (T light) turns on, the S light turns off, and the typewriter motor turns on.
3. Enter your command or reply and press the Return key. If your console does not have the automatic EOB feature on the Return key, press the EOB (end-of-block) key. Your console momentarily switches to receive status (R light on) so it can accept the checking reply from the system (Receive and Restart lights on). If the system receives the command or reply correctly, your console automatically switches back to standby status (Standby light on and Restart light off).

### Responding to a Negative System Response on the 2740

If the system receives a command or reply incorrectly, you get a negative response that causes a dash or underscore symbol to be printed as the first character in the line after your entry. A negative response also causes the keyboard to lock, the Restart light to remain on, the Transmit light to turn on, and the alarm to sound at your terminal. Pressing the Restart key unlocks the keyboard, turns off the alarm, and turns off the Restart light. You can then either enter the command or reply again or press EOB to put the 2740 back in standby status.

### Changing Information on the 2740

If you make an error when entering a command or reply, correct the error by backspacing to the characters in error and typing over them with the correct characters. Start at or before the first incorrect character and continue to the end of the command or reply. To cancel a command or reply, backspace once and press the Return key.

## Receiving Messages on the 2740

When the system sends messages to a 2740, it first readies the console for the messages by entering a call. For your console to receive a call:

1. The Local/Communicate switch must be set to COM. If it is not, an alarm sounds when the console receives the call.
2. Paper must be inserted. If it is not, an alarm sounds when the console receives the call.
3. The Standby light must be on.

When your terminal receives the system's call, the Standby light turns off, the Receive light turns on, and the typewriter motor turns on. Your keyboard then locks, and the system prints its message. After the message finishes, the Standby light turns on, the keyboard remains locked except for the Bid key and EOT key, and the typewriter motor turns off.

If the system prints a dash or underscore as the first character of the line following the message, your 2740 might have received a part of the message incorrectly. The system tries to send the message again. If, however, after five retries, your console is still receiving the message incorrectly, the system switches automatically to the alternate console.

If your console runs out of paper while receiving messages, it switches to a no-status standby condition. In this condition, all status lights turn off, the keyboard locks, and the message you are receiving when the paper runs out is lost.

## 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. Alternatively, select the ENTER indicator on the screen with the selector pen or cursor.

**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.

## 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 *one* of the following methods:
  - Perform the ENTER action.
  - Place the selector pen over the ENTER indicator in the instruction line.
  - Move the cursor under the ENTER indicator in the instruction line and press the ENTER key.

## 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  $\rightarrow|$ , the back-tab key  $\leftarrow|$ , or the new line key  $\leftarrow|$ .
- 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 a 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.

## 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 2-43.

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 *Message Library: System Messages*.

## 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 Commands in PFK Tables" and "Defining PFKs Using the CONTROL Command" in Chapter 3.

This page left blank.

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.

The maximum number of PFKs provided for each display console device appears earlier in Figure 2-2 on page 2-6.

If your system programmer does not define and activate a PFK table for your PFKs, IBM supplies the following definitions (in nonconversational mode):

|           | PFK                | Command Comment                               |
|-----------|--------------------|---|
| 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                        |

**Figure 2-10. IBM Defaults for PFKs**

If you are controlling JES3 through an MCS console, your PFK definitions can reside in a JES3 PFK table. These definitions are in effect only for JES3. To define PFKs for JES3, see *JES3 Initialization and Tuning*.

### 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 2-43.

## 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.

| 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 nn TABLE - MASTCMDS IN PFKTAB02
KEY# CON -----DEFINITION-----
```

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

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

```
IEE235I hh:mm:ss PFK DISPLAY
NO PFK DEFINITIONS FOR CONSOLE 04
```

*Example 2:* To determine the definitions in effect for the PFKs on console 04, enter:

```
DISPLAY PFK, CN=4
```

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 CONSOLE 4 TABLE - MASTCMDS IN PFKTABJC
```

where the PFK table in effect for console 4 is MASTCMDS in the PFKTABJC member of SYS1.PARMLIB.



The definition for each key appears under the headings. If, however, no PFKs are defined for 4, 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 the ENTER action or by selecting the ENTER indicator with the selector pen. The next command associated with the PFK (if any) then appears in the entry area.
  - Change the command from the keyboard before entering it (see “Changing Information in the Entry Area” on page 2-43.)
  - Cancel the command that appears in the entry area by performing a CANCEL action or by selecting the CANCEL indicator with the selector pen. 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.

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.”*
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 “Entering Commands with the Selector Pen in Conversational Mode.”

### **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 or the ENTER indicator with the selector pen. 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.”
3. Cancel the command in the entry area by performing a CANCEL action or by positioning the selector pen over the CANCEL indicator in the instruction line. 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 might want to change information in the entry area to correct a typing error or to change a command during conversational command entry or message deletion. (Both conversational command entry and message deletion are described in this section.) You might also want to blank out the entry area without entering a command to the system.

### **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, 'DIPSLAY 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:** Don't 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.

## Using the Hardcopy and System Logs

The hardcopy log and the system log (SYSLOG) provide a permanent record of system activity. The hardcopy log can be printed at once by the hardcopy output device or directed to the system log; the system log is printed periodically. In fact, at some installations, the system log and the hardcopy log are the same, and the terms are used interchangeably. Either way, you have a permanent record that your installation can use to obtain information about operating conditions and maintenance.

### Using the Hardcopy Log

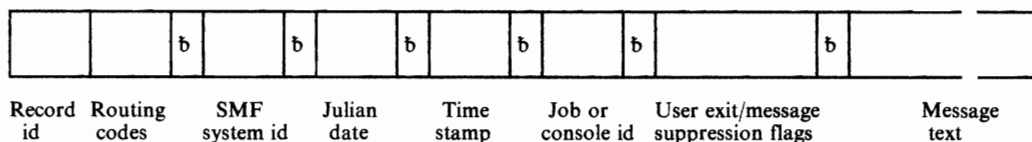
The hardcopy log is a record of:

- Messages with specified routing codes
- Broadcast messages
- Commands
- In-line command responses (command responses routed to specified display areas on consoles)
- Out-of-line command responses (command responses routed to general display areas on consoles)

You have some control over what messages, commands, and command responses the hardcopy log records. "Controlling the Hardcopy Log" in Chapter 3 describes how you can assign the hardcopy log device and specify what the hardcopy log is to record.

With each message, the system includes information that you can use to direct messages to the consoles at your installation. Specifically, it tells you which consoles the message is routed to.

Figure 2-11 shows the first line of a single line and a multi-line message:



**Figure 2-11. First Line of a Hardcopy Log Message**

The first character on each message line, as part of the record id, identifies the *record type*. The second character identifies the *request type*

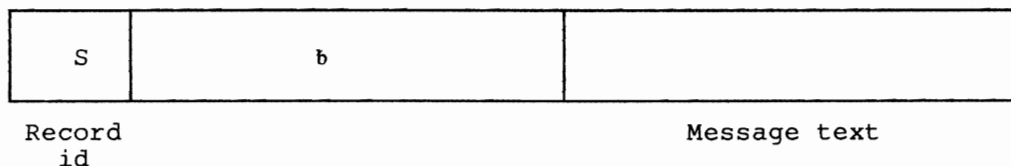
On first lines of single and multi-line messages, the record type can be one of the following letters:

- N - first line of a single-line message that does not need a reply
- W - first line of a single-line message that does need a reply
- M - first line of a multi-line message
- O - text of an operator LOG command
- X - message other than a system message or the text of a LOG command

The request type for *any* line can be one of the following:

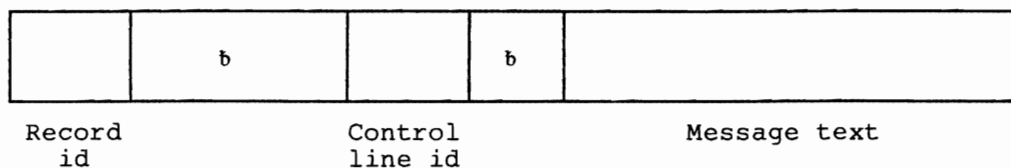
- C - command input
- R - command response
- I - command issued internally (The job id field contains the name of the internal issuer, for example, TSO userid.)
- blank - other

The single line message text might be split, depending on the output device. The format is:



The record type is identified by the letter 'S'.

Subsequent lines of multi-line messages have this format:



The record type for the subsequent lines of multi-line messages can be one of the following letters:

- L - label line of a multi-line message
- D - data line
- E - last line of a multi-line message (Not every last line of a multi-line message is identified with this letter.)

All forms of the CONTROL command are recorded on the hardcopy log except CONTROL with no operands, CONTROL D, and CONTROL E.

The hardcopy log is required in systems with one active display console or more than one active console because it provides a permanent, consolidated record of console message traffic. The hardcopy log is also required when an installation uses the message processing facility (MPF) to suppress nonessential messages and thus decrease message volume. Suppressed messages do not appear on the console display; they appear only on the hardcopy log.

If the console producing this log malfunctions, MVS/XA assigns the log to another console. To find another console, MVS/XA searches the alternate console chain of the failing console. If the search is not successful, MVS/XA searches the master console and its alternate console chain. If this search is not successful, MVS/XA selects any active non-display console by checking all consoles in the order they were specified in the CONSOLxx member of SYS1.PARMLIB. MVS/XA assigns the hardcopy function to the first active non-display console it finds. If the entire search is unsuccessful, MVS/XA suspends the hardcopy log and notifies the master console.

If hardcopy is not present and you issue a VARY ddd, CONSOLE command to add a second active console to the system, MVS/XA assigns a hardcopy log. The type of hardcopy log is indicated in Figure 2-12.

| If the type of console you are assigning is:               | And the master console is a PRINTER console, MVS/XA assigns: | And the master console is a DISPLAY console, MVS/XA assigns: |
|--|--|--|
| Nondisplay Secondary                                       | Hardcopy = Master Console                                    | Hardcopy = Secondary Console                                 |
| Display Secondary without SYSLOG                           | Hardcopy = Master Console                                    | Hardcopy = *suspended  |
| Display Secondary with SYSLOG                              | Hardcopy = Master Console                                    | Hardcopy = SYSLOG  |
| *The master console is notified and hardcopy is suspended. |  |  |

**Figure 2-12. Hardcopy Devices that MVS/XA Assigns**

To display information about the hardcopy log, enter

DISPLAY CONSOLES ,HARDCOPY

## Using the System Log

The system log (SYSLOG) is a data set residing 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 problem programs and system functions. You can use the LOG command to add an entry to the system log.

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.

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.

The system log often includes the hardcopy log. If it does, the system attempts to switch hardcopy to an appropriate printing console if the SYSLOG device fails. If a suitable console is not active at the time of failure, hardcopy is suspended and the master console is notified.

## 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" 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.

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 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 hardcopy 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 “Assigning Alternate Consoles” in Chapter 3 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:

```
IEE143I CONSOLE SWITCH, OLD = console NEW = console REASON CODE = rc
```

Appropriate values and 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.”

### 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 hardcopy device. 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 hardcopy log could be the only place they will subsequently appear.

The VARY command provides an alternate method of recovering lost messages on a failing console. However, this method does not recover multiple-line messages in progress. If the

failing console is a full-capability console, do the following to avoid losing the waiting messages:

1. Use the VARY command to make the current master console the alternate of the failing console.
2. Use the VARY command to make the failing console the master console.
3. Press the interrupt key on the processor. This action causes the failing console's functions and waiting messages to be automatically switched to the alternate console.

## Responding to Console Message Backups

The MVS/XA 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 member of SYS1.PARMLIB. If these parameters are not coded, system defaults are in effect:

- The default for WTO buffers is 1500.
- The default for WTOR buffers is 10.

When messages fill up the buffers, non-privileged jobs go into wait states. To avoid WTO message buffer shortages, you can raise your WTO buffer limit and adjust message deletion specifications on your consoles. To avoid WTOR message buffer shortage, raise your WTOR buffer limit and reply to WTORs more frequently. Procedures for responding to WTO and WTOR buffers shortages follow in this section. Procedures for adjusting the message deletion specifications appear in "Finding Appropriate Message Deletion Specifications" in Chapter 3.

## 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
```

If the problem continues and WTO buffer use reaches its limit, the system issues the following action message:

```
IEA404A WTO BUFFER SHORTAGE CRITICAL - 100% FULL
```

In addition, the system puts all non-privileged tasks requesting a WTO buffer into a wait state.

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.

- A console is not in roll mode, and messages are filling the console message buffers.
- The 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 is not sufficient.)

To determine the extent of the problem and the responsible console or consoles, issue the DISPLAY CONSOLES,BACKLOG command. Figure 2-13 shows an example of the DISPLAY CONSOLES,BACKLOG output. Information about all consoles that have any outstanding WTO messages appears. The output in the figure includes the following line:

```
MSG: CURR=1356 LIM=1500  RPLY: CURR=1 LIM=10  NOSUBSYS  PFK=02
```

In this line, MSG: CURR = 1356 LIM = 1500 tells you the current use of WTO buffers and the specified limit. RPLY: CURR = 1 LIM = 10 tells you the number of WTOR messages that have 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 2-13 indicates, through NBUF, the number of buffers queued to each console. It is clear that 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.

```

D C,B
IEE249I 09.21.31 CONSOLE DISPLAY 214
MSG: CURR=1356 LIM=1500  RPLY: CURR=0 LIM=10  NOSUBSYS  PFK=NONE
CONSOLE/ALT  ID -----
SYSLOG      COND=H  AUTH=CMDS  NBUF=0
            ROUTCDE=ALL
            3D0/3D4  12  COND=A  AUTH=ALL  NBUF=1217
            AREA=Z,A  MFORM=M
            DEL=RD  RTME=2  RNUM=5  SEG=10  CON=N
            USE=FC  LEVEL=ALL  PFKTAB=*DEFAULT
            ROUTCDE=ALL
            014/3D4  08  COND=A  AUTH=ALL  NBUF=0
            AREA=Z,A  MFORM=M
            DEL=RD  RTME=2  RNUM=5  SEG=14  CON=N
            USE=FC  LEVEL=ALL,NB  PFKTAB=*DEFAULT
            ROUTCDE=ALL

IEE152I  ENTER CANCEL D C,K

IEE163I  MODE=RD

```

**Figure 2-13. Output of DISPLAY CONSOLES,BACKLOG Command**

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 state until the buffer count goes below the limit.*
- 3. The system does not use the MLIM and RLIM parameter values specified in the CONSOLxx member of SYS1.PARMLIB until either the hardcopy facility (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.

## Responding to a Master Console Failure

Normally, MVS/XA 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 an attempt is made to switch the master console, all active secondary consoles receive a message asking that the VARY MSTCONS command be entered. The first VARY MSTCONS command entered with the correct syntax from any secondary console is accepted and processed. A message is issued that informs the operator of the new master console of the change and gives the device number of the previous master console.

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

## 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 are put in a wait state and MVS/XA enters an enabled wait. See “Responding to Console Message Backups” earlier in this section.

*Note:* Output-only consoles and JES3 consoles can continue to function during a no-consoles condition.

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

- Issuing message IEA420A to any active output-only console or JES3 console.

```
IEA420A NO FULL CAPABILITY CONSOLES,REASON=xxxx
      TO RESTORE MASTER CONSOLE:
      1) PRESS ENTER, REQUEST, OR END ON ANY AVAILABLE CONSOLE
      2) PRESS THE EXTERNAL INTERRUPT KEY
```

- Sounding an alarm on a device. The system can sound a device alarm only if:
  - The device was specified as a console at system generation
  - The device has an alarm feature
  - The device is online or in console status
  - The device is unallocated
  - The device is not a JES3 console
- 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 IEA420A can appear and might not find any devices on which to sound an alarm.
- Rejecting, with the following message, any VARY CONSOLE commands issued while the system is in a no-consoles condition:

```
IEE744I VARY CONSOLE REJECTED - NO FULL CAPABILITY CONSOLES
      TO RESTORE MASTER CONSOLE:
      1) PRESS ENTER, REQUEST, OR END, ON ANY AVAILABLE CONSOLE
      2) PRESS THE EXTERNAL INTERRUPT KEY
```

*Note:* The VARY CONSOLE command could have been entered from a JES3 console.

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

- You cannot attempt recovery until all alarms have sounded and message IEA420A has been issued.
- Recovering the master console should be your first priority.

To recover the master console:

1. Select a device other than a 2740, preferably the master console, to be the master console.
2. Power on the device if it is not already powered on.
3. Press the ENTER, REQUEST, or END key. This action causes an attention interrupt to be generated on the device.
4. Press the external interrupt key to cause the system to attempt to recover the master console.

If the recovery attempt is successful, the device that generated the attention interrupt becomes the master console and message IEE143I appears on it.

```
IEE143I CONSOLE SWITCH, OLD = console NEW = console REASON CODE = rc
```

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.

*Notes:*

1. *The first device on which an attention interrupt is generated is selected to be the new master console. Any attention interrupts generated for any device after the first are ignored.*
2. *If message lines on the screen of the new master console are not fully recovered, press the PA2 (CANCEL) key.*

If the recovery attempt is unsuccessful, the device that generated the attention interrupt does not become active and the alarm sounds. Repeat the recovery procedure once more. If the device does not become active after the second attempt, there could be an error on the device and it cannot be used. Select another device and repeat the recovery procedure.

If you cannot make a device active by using the general recovery procedure, you might take away from a subsystem, such as JES3, one of its full-function consoles. For example, if JES3 is active and has two full-capability JES3 consoles (CN1 and CN2), try the following:

1. Using JES3 commands, direct JES3 to give up control of a full-capability console:

```
*SWITCH,CN2,CN1
*DISABLE,CN2
```

The console is now available to MCS.

2. Press the external interrupt key on the processor. If the old master console becomes active, the no-consoles condition has been resolved and the former JES3 console is no longer needed. If the old master console does not become active, proceed to the next step.
3. Press ENTER, REQUEST, or END on the former JES3 console.
4. Press the external interrupt key. The former JES3 console becomes the new master console.

If you have only one JES3 console, do not enter the \*SWITCH,CN2,CN1 command, but follow all other procedures. If JES3 is not active and the general recovery procedure does not work, you must re-IPL.

## Processing MVS Messages at the System Console

During system recovery, MVS might try to communicate with you through the disabled console communications facility (DCCF). It tries first to use the MCS console or its alternate. If it cannot process messages through either of these consoles, it tries to issue a message to the system (hardware) console. When this action is successful, the alarm on the processor 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 MODIFY commands on the system console:

```
F SYSMSG (on a 308x processor)
      or
F SCPMSF (on a 3090 processor)
```

A response line appears below the MVS message. Unlike the response line on an MCS console, the response line on the system console does not contain the characters "R 00." Do not enter these characters. Instead, enter the response requested by the MVS message.

The MVS message on the system console does not time out; the message remains on the screen until you enter a reply.



## Chapter 3. Defining and Changing Console Characteristics

The IODEVICE statement in the MVSCP makes a device available for MVS to use. A CONSOLE statement in the CONSOLxx member of SYS1.PARMLIB establishes the device as an MCS console and defines certain values for its characteristics. These values are specified by system programmers or are IBM defaults. After IPL, operators can use certain commands to change the characteristics. The effects of these commands last only for the duration of the IPL; at the next IPL, the values will be those in the SYS1.PARMLIB members or the IBM defaults.

This chapter describes:

- How system programmers can establish the consoles and their characteristics through SYS1.PARMLIB members
- How operators can use CONTROL, SET, MSGRT, MONITOR, or VARY commands to make changes to the console characteristics
- The IBM defaults that exist if console characteristics are not defined by SYS1.PARMLIB members
- How to display the current console characteristics
- Some of the operations that are affected by how the consoles are defined

It begins with a general description of the SYS1.PARMLIB members that affect the operations and characteristics of the consoles you use to control the system. It ends with “Managing Messages in a Console Cluster,” an extended description of how to set up four consoles to share the functions of a single console. The description includes many practical examples of defining and changing console characteristics.

## Using SYS1.PARMLIB Members to Define Your Consoles

Three members of SYS1.PARMLIB define your consoles and control how the system uses your consoles to communicate with you:

- CONSOLxx defines console characteristics.
- MPFLSTxx defines message retention, suppression, and presentation.
- PFKTABxx contains the PFK tables that are available for your consoles.

When your system comes up, the definitions in these 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.

Do not use these pages to code the SYS1.PARMLIB members; for that purpose, turn to *Initialization and Tuning*.

### Using CONSOLxx to Define Console Characteristics

The CONSOLxx member of SYS1.PARMLIB defines the characteristics for your consoles. This member determines such important factors as how the system routes your message traffic, how your PFKs are defined, and how you delete messages.

At IPL, the system requires that a CONSOLxx member be identified. If the CON parameter in IEASYSxx member of SYS1.PARMLIB is not coded, the operator must enter CON=xx (where xx is the suffix for CONSOLxx) in response to the message:

```
IEA101A SPECIFY SYSTEM PARAMETERS FOR RELEASE xx.yy.zz
```

The name of the parameter in CONSOLxx usually corresponds to the name of the operand in the MVS command. For example, the USE(FC) parameter on the CONSOLE statement in CONSOLxx defines a console as a full-capability console. The USE=FC operand on the CONTROL V command performs the same function; however, the effect of the command lasts only for the duration of the IPL. Figure 3-1, Figure 3-2, and Figure 3-3 compare the operands on system commands with the parameters in CONSOLxx.

CONSOLxx contains four statements:

- CONSOLE
- INIT
- HARDCOPY
- DEFAULT

The following four sections describe the parameters on these four statements. Tables in these sections identify the commands an operator or system programmer can use to make changes to console characteristics and tell you what the system defines for your console if a parameter is not coded. Programmers can use the tables to help plan the console definitions; operators can use the tables to understand how to temporarily change some of the definitions for the rest of the IPL.

You may have statements in the COMMNDxx member of SYS1.PARMLIB that are similar to the statements in the CONSOLxx member. To avoid having the same definition in two SYS1.PARMLIB members, use the CONSOLxx member to define the characteristics on your consoles.

### Using the CONSOLE Statement in the CONSOLxx Member

Each console statement defines the characteristics for one console. The first parameter is DEVNUM, which identifies its device number. The rest of the parameters on the CONSOLE statement:

- Control message displays
- Identify a PFK table
- Define the console operating mode
- Define how to delete messages that appear
- Define the messages that can appear at the console
- Define the commands that an operator can issue from the console
- Establish an alternate console
- Establish system command groups

Figure 3-1 describes each command that has a corresponding CONSOLE parameter in CONSOLxx, the parameter in CONSOLxx, and the console characteristic that the parameter or command affects. The value in parentheses indicates the default.

| MVS/XA Commands             | Parameters on the CONSOLE Statement   | Characteristic that the Parameter Affects  |
|-----------------------------|---------------------------------------|--|
| CONTROL A,nn                | AREA (See Figure 3-12 on page 3-44)   | Size of the out-of-line display areas  |
| CONTROL N,PFK               | PFKTAB (See Figure 2-10 on page 2-39) | PFK table  |
| CONTROL S,CON               | CON(N)                                | Conversational or nonconversational mode   |
| CONTROL S,DEL               | DEL(RD)                               | Message deletion mode  |
| CONTROL S,MFORM             | MFORM(M)                              | Format in which the messages appear  |
| CONTROL S,RNUM              | RNUM(5)                               | Number of message lines included in one message roll                               |
| CONTROL S,RTME              | RTME(2)                               | Number of seconds between message rolls  |
| CONTROL S,SEG               | SEG (See Figure 3-10 on page 3-35)    | 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,LEVEL             | LEVEL(ALL)                            | 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 devnum,CONSOLE,ALTCONS | ALTERNATE                             | Alternate console  |
| VARY devnum,CONSOLE, AUTH   | AUTH(INFO)                            | Command groups   |
| VARY devnum,CONSOLE, ROUT   | ROUTCODE(NONE)                        | Routing codes for the console  |

Figure 3-1. Comparison of System Commands and CONSOLE Statements in CONSOLxx

### Using 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. Use the INIT statement to:

- Specify the limits for WTO and WTOR buffers
- Activate MPFLSTxx and PFKTABxx members of SYS1.PARMLIB
- Activate the action message retention facility
- Activate the WTO user exit IEAVMXIT
- Specify the MVS command delimiter
- Specify how the MONITOR command is to display mount and demount messages

Figure 3-2 describes each MVS/XA 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.

| MVS/XA Commands | Parameters on INIT Statement | Characteristic that the Parameter Affects  |
|-----------------|------------------------------|--|
| CONTROL M,AMRF  | AMRF(Y)                      | Establishes whether the action message retention facility is to be active                                    |
| CONTROL M,MLIM  | MLIM(1500)                   | Limits the number of buffers for WTO messages that the system has not yet displayed                          |
| 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 user exit IEAVMXIT is to be active   |
| MONITOR         | MONITOR                      | Establishes how the system displays mount and demount messages in response to the MONITOR command            |
| SET MPF         | MPF(NO)                      | Specifies the MPFLSTxx member of SYS1.PARMLIB that the system is to use                                      |
| SET PFK         | PFK(NONE)                    | Specifies the PFKTABxx member of SYS1.PARMLIB that holds the PFK tables that are available for your consoles |

Figure 3-2. Comparison of System Commands and INIT Statements in CONSOLxx

### Using the HARDCOPY Statement in the CONSOLxx Member

Use the HARDCOPY statement in the CONSOLxx member of SYS1.PARMLIB to assign the hardcopy log to a device or to SYSLOG, and to establish what kinds of messages the system is to record on the hardcopy log. Figure 3-3 describes each VARY ,HARDCPY command operand, the corresponding parameter in CONSOLxx, and the task the command and parameter performs. The value in parentheses indicates the default.

| MVS/XA VARY HARDCPY Command Parameters | Parameters on HARDCOPY Statement | Description   |
|--|----------------------------------|---|
| devnum or SYSLOG                       | DEVNUM(SYSLOG)                   | Establishes whether the hardcopy log is a device or SYSLOG  |
| devnum,ROUT                            | ROUTCODE(ALL)                    | Establishes the routing codes for messages the hardcopy log is to receive                           |
| NOCMDS, INCMDS, STCMDS, or CMDS        | CMDLEVEL(CMDS)                   | Establishes whether the hardcopy log is to receive operator commands, responses, or status displays |

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

The HARDCOPY statement is optional; you may not have more than one. If you don't code this statement, the system records all messages on SYSLOG.

## Using the DEFAULT Statement in the CONSOLxx Member

Use parameters on the DEFAULT statement in CONSOLxx member of SYS1.PARMLIB to assign default routing codes to all WTO and WTOR messages that do not have routing codes specifically assigned. You cannot assign these default values through an MVS/XA command, not even on a temporary basis. If you don't code this statement, the system assigns routing codes 1 through 16 to the messages.

## Using MPFLSTxx to Control Messages

The MPFLSTxx member of SYS1.PARMLIB contains information that the message processing facility (MPF) uses to control:

- Message suppression

A suppressed message does not appear at a console but is written to the hardcopy log.

- Message retention

The action message retention facility keeps action messages in a buffer area, allowing the operator to request that any action messages not acted upon be recalled to the screen. In MPFLSTxx member, you can identify certain action messages that the action message retention facility is not to retain.

- Message presentation

On certain devices, messages can appear with highlighting, in color, or with added intensity.

- User exits

These user exits can gain control when a message is issued, at which point they can perform additional processing.

The MPF parameter on the INIT statement in CONSOLxx identifies the MPFLSTxx member that you want the system to use. When an MPFLSTxx member is active, the system checks each time a message is issued to see if MPF has to process the message.

For more information on MPFLSTxx, see "Defining the Message Processing Facility (MPF)" later in this chapter.

## Using PFKTABxx to Define your PFks

The PFKTABxx members of SYS1.PARMLIB 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.

By using CONSOLxx and PFKTABxx members, you ensure the PFK definitions at IPL. There are also ways to make changes to the PFKs that last only for the duration of the IPL:

- SET PFK = xx changes the PFKTABxx member in effect for the console. (You make the tables within the member available for assignment through the CONTROL command. See the next item in the list.) Other consoles using the former PFKTABxx member are not affected by your issuing the SET command for your console.
- CONTROL N,PFK = nnnnnnnn assigns the PFK table that contains the PFK definitions for the console.
- CONTROL N,PFK = (nn,CMD = ) changes a PFK definition for the console.

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:

```
IEA180 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 Figure 2-10 on page 2-39.

To define PFKs for your consoles, see “Defining PFKs in PFK Tables” on page 3-28.

## Controlling the Hardcopy Log

The hardcopy log provides a permanent record of messages with specified routing codes, broadcast messages, and, optionally, commands and command responses. The format of the hardcopy log message appears in Chapter 2 under “Using the Hardcopy Log.” The following section describes how you can control the type of device that the hardcopy log uses and some types of messages, commands, and command responses that appear in the hardcopy log.

### Assigning the Hardcopy Log to a Device

You can assign the hardcopy log to a device, such as a printer, or to the system log (SYSLOG). If you use SYSLOG as the hardcopy log, start a writer that includes the system log message class (A for MVS/XA, unless otherwise specified in your installation).

To display information about the hardcopy log, issue the DISPLAY CONSOLES,HARDCOPY command. The resulting display tells you whether the hardcopy log is SYSLOG or a device, what kinds of messages are directed to the log, and the number of messages waiting to be placed on it.

You assign the hardcopy log by:

- **Using the CONSOLxx Member of SYS1.PARMLIB**

The DEVNUM parameter on the HARDCOPY statement in the CONSOLxx member controls the assignment of the hardcopy log. If you do not code a HARDCOPY statement, the system sends all hardcopy log messages to the system log (SYSLOG).

*Example:* To assign the hardcopy log to an output-only device, such as a 3203 printer with device number 0E3, code the following statement in the CONSOLxx member:

```
HARDCOPY DEVNUM(0E3)
```

- **Using the VARY Command:**

Use the HARDCPY operand on the VARY command to assign the hardcopy log to a device. Issue this command only from the master console.

*Example:* To assign the hardcopy log to 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.

## Routing Messages to the Hardcopy Log

Unless you specify otherwise, the hardcopy log records all messages, except those that are explicitly omitted from the log through the WTO macro or user exits. You can request that the hardcopy log not record 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 sends messages with these routing codes to the log anyway.

To see information about the kinds of messages that the system routes to the hardcopy log, but not to any console, issue the DISPLAY CONSOLES,HCONLY command.

You control the routing of some messages to the hardcopy log by:

- **Using the CONSOLxx Member of SYS1.PARMLIB:**

The ROUTCODE parameter on the HARDCOPY statement in the CONSOLxx member controls the routing of messages to the hardcopy log.

*Example:* To assign the minimum set of routing codes (1, 2, 3, 4, 7, 8, 10, and 42) to the hardcopy log, code the following statement in the CONSOLxx member:

```
HARDCOPY ROUTCODE(1-4,7,8,10,42)
```

If you asked for fewer routing codes than the minimum set, the system gives you the minimum set anyway.

- **Using the VARY Command:**

Use the VARY ,HARDCPY command to change the routing codes for the hardcopy log. You can add to the existing set (AROUT operand), subtract from the existing set (DROUT), or redefine the set (ROUT).

*Example:* To remove all routing codes except the minimum set of routing codes, enter:

```
VARY ,HARDCPY,DROUT(5,6,9,11-41,43-128)
```

Actually, 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.

### **Sending Commands and Command Responses to the Hardcopy Log**

Unless you specify otherwise, the system records all operator and system commands, responses, and status displays (including static and time-interval updated status displays). To request that some commands and command responses not be recorded on the hardcopy log, the system gives you the following choices:

**NOCMDS** The system records no commands or responses.

**INCMDS** The system records all operator commands and responses except status displays.

**STCMDS** The system records all operator commands and responses except dynamic status displays.

**CMDS** The system records all operator commands and responses, including status displays and dynamic status displays.

To see what commands and command responses the system records on the hardcopy log, issue the DISPLAY CONSOLES command. Figure 3-6 on page 3-14 shows the display that appears in response to this command.

**You control the routing of commands and command responses to the hardcopy log by:**

- **Using the CONSOLxx Member of SYS1.PARMLIB:**

The CMDLEVEL parameter on the HARDCOPY statement in the CONSOLxx member controls the recording of operator commands and command responses. It allows you to specify what kinds of commands and command responses you want to have recorded on the log.

*Example:* To request that the hardcopy log keep a record of all operator commands and responses except status displays, code the following statement in the CONSOLxx member:

```
HARDCOPY CMDLEVEL(INCMDS)
```



- **Using the VARY Command:**

Use the VARY ,HARDCPY command to change the commands or the command responses that the hardcopy log records.

*Example:* To request that the hardcopy log keep a record of 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.

## Making Console Assignments

Every system has one master console, with, optionally, secondary consoles to share the responsibilities of monitoring and controlling the system. In dividing responsibilities among different consoles, an installation can limit some consoles in their ability to accept commands. For example, the master console needs the authority to issue **all** commands, including those that affect other consoles. On the other hand, a console used only to accept I/O commands, such as PURGE, MOUNT, UNLOAD, needs the authority to accept only those commands. For this reason, the system commands are grouped into system command groups.

### System Command Groups

MVS commands are assigned to one of five command groups according to command function. The command groups are:

- Informational commands (INFO)
- System control commands (SYS)
- I/O control commands (IO)
- Console control commands (CONS)
- Master console-only commands (MSTCONS, MC, or MASTER)

The commands in each group are shown in Figure 3-4. (For information about JES commands, see *Operations: JES2 Commands* or *Operations: 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/XA rejects the command and sends an error message to the issuing console. At the master console, you can enter all operator commands. In addition, only at the master console can you enter master console-only commands and the special operands of the VARY command listed in Figure 3-4.

| Command Group   | Commands  |
|---|---|
| INFO  | CONTROL (See Note 6)<br>DEVSERV<br>DISPLAY (See Notes 1 and 4)<br>LOG<br>MONITOR<br>MSGRT (See Note 1)<br>REPLY<br>SEND<br>STOPMN<br>STOPTR (See Note 1)<br>TRACK (See Note 1)  |
| SYS<br>(System Control)   | CANCEL<br>CHNGDUMP<br>DUMPDS<br>HALT (See Note 5)<br>HOLD (See Note 2)<br>MODE<br>PAGEADD<br>RELEASE (See Note 2)<br>RESET<br>SET<br>SETDMN<br>SETSMF<br>SLIP<br>START<br>STOP<br>SWITCH<br>TRACE (with ST or STATUS)<br>WRITELOG |
| IO<br>(I/O Control)   | ASSIGN (See Note 4)<br>MOUNT<br>PURGE (See Note 4)<br>SWAP<br>UNLOAD<br>VARY {<br>OFFLINE<br>ONLINE<br>PATH<br>devnum<br>NET }<br>(See Note 3)<br>(See Note 5)  |
| CONS<br>(Console Control)   | CONTROL (See Note 6)<br>VARY {<br>ALTCONS<br>CONSOLE }<br>VARY {<br>OFFLINE<br>ONLINE<br>PATH<br>devnum } (See Note 3)  |
| MASTER, MC or<br>MASTCONS<br>(Master Console only)  | VARY {<br>AUTH<br>CN<br>GRS<br>HARDCPY<br>MSTCONS<br>CONFIG }<br>CONTROL (See Note 6)<br>DUMP<br>FORCE<br>OFFLINE, FORCE<br>QUIESCE<br>TRACE (with MT)  |
| <p><i>Notes:</i></p> <ol style="list-style-type: none"> <li><i>CONS command group when message routing is specified.</i></li> <li><i>HOLD and RELEASE are explained in TCAM Operation.</i></li> <li><i>IO command group when specifying a non-console device; CONS when specifying a console.</i></li> <li><i>ASSIGN, PURGE, and DISPLAY 3850 are explained in Mass Storage System Extensions Operation.</i></li> <li><i>HALT NET and VARY NET are explained in VTAM Operation.</i></li> <li><i>CONTROL is in the INFO command group except when</i> <ul style="list-style-type: none"> <li><i>Rerouting the message queues of any other full-capability MCS console - MASTER.</i></li> <li><i>Message routing is specified - CONS.</i></li> <li><i>Changing or displaying the status of the action message retention facility.</i></li> <li><i>Changing or displaying the number of allowed message buffers - MASTER.</i></li> <li><i>Changing or displaying the status of WTO user exit IEAVMXIT - MASTER.</i></li> </ul> </li> </ol> |   |

**Figure 3-4. System Command Groups**

## Assigning a Master Console or Secondary Console

By assigning the system command groups for a console, you establish the master console and the secondary consoles.

You assign the master and secondary consoles by:

- **Using the CONSOLxx Member of SYS1.PARMLIB:**

The AUTH parameter on the CONSOLE statement in CONSOLxx establishes the kinds of commands that a console can accept.

*Example:* To assign the console with device number 031 as the master console, code the following statement in CONSOLxx.

```
CONSOLE DEVNUM(031) AUTH(MASTER)
```

If no master console is defined in CONSOLxx member, the system chooses the first full-capability console defined in the CONSOLxx member.

- **Using the VARY Command:**

The VARY device,MSTCONS command defines the master console by establishing the commands that a console can receive. If the master console is operating properly, you can switch to another master without disrupting normal operations.

*Example:* To reassign the console with device number 31E as the master console, enter:

```
VARY 31E,MSTCONS
```

Enter this command through the current master console. If you try to enter this command from a secondary console when the master console is operating normally, the command is rejected and a message appears at the secondary console to indicate that the switch did not take place.

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

## Placing a Console in Offline Status

When a secondary console or the hardcopy log device must be bypassed for any reason, you must enter a VARY command to place the secondary 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 are now routed to the master console or to the hardcopy log.

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 log device, you must change the hardcopy log function 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 on the IODEVICE statement in the input stream to the MVSCP, you can replace it with another device that supports the 3270 data stream architecture without either performing an IPL or running the MVSCP. These are the steps:

- Vary the device offline.
- Detach the device from the control unit.
- Attach the new device to the control unit.
- Customize the control unit for the new device, if necessary.
- Turn the device on.
- Vary the device into console state. MCS will query the device to determine its characteristics.

## Assigning Alternate Consoles

Each console, whether secondary or master, is assigned an alternate console, or **alternate**, to back it up if it malfunctions. The alternate is usually another functioning console in the system configuration, but it can be any console 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 one of the consoles is failing, it automatically switches to an active console in the failing console's alternate console chain. If an active alternate cannot be found, the master console is assigned. Figure 3-5 shows one possible arrangement of alternate consoles. An arrow points to the console that assumes the functions of a console, should the first console fail.

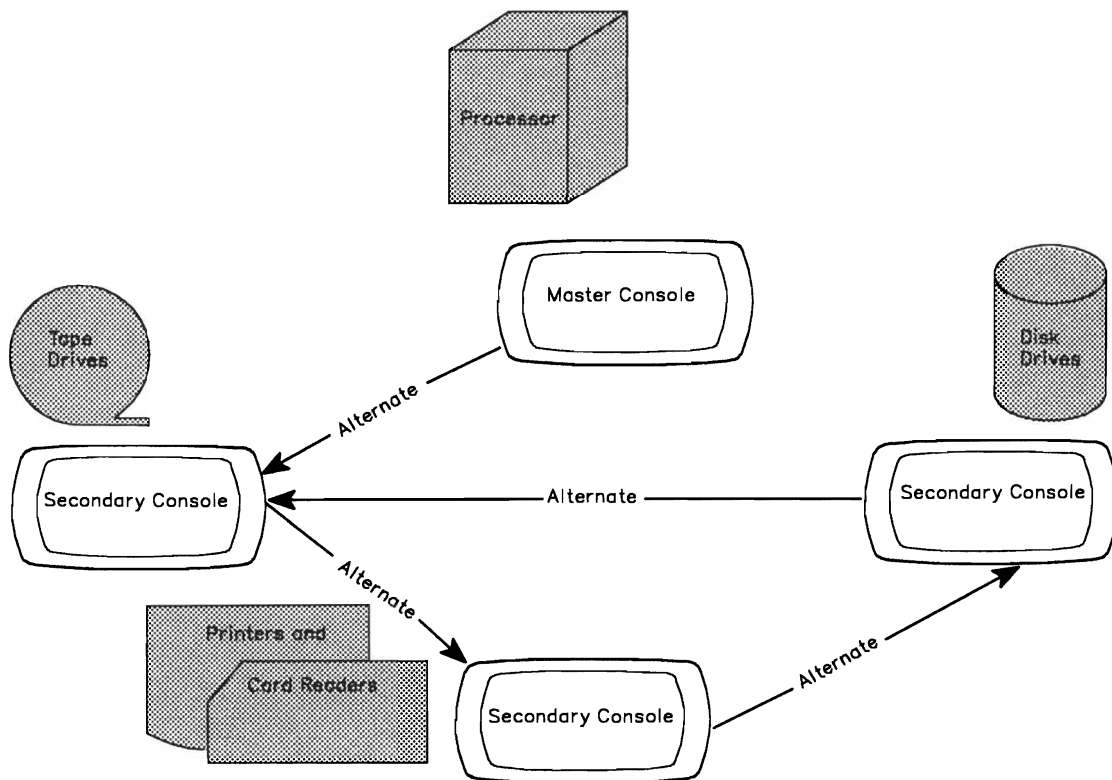


Figure 3-5. Sample Alternate Console Chain

When a console fails, the communication routines in MVS/XA attempt to switch to the failing console's alternate. The first active alternate console MVS/XA finds is assigned the functions of the failing console. If there is no active alternate console, the master console assumes the failing console's function.

MVS/XA transfers to the alternate console all unanswered and unissued messages from the failing console. However, 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.

To determine the alternate console chain for your consoles, use the DISPLAY CONSOLES command. Figure 3-6 shows the display that appears in response to this command.

You assign alternate consoles by:

- **Using the CONSOLxx Member of SYS1.PARMLIB:**

The ALTERNATE parameter on the CONSOLE statement in SYS1.PARMLIB establishes the alternate console for each console.

*Example:* To assign the device with device number 2E1 as the alternate for the console with device number 1E1, code the following statement in CONSOLxx:

```
CONSOLE DEVNUM(1E1) ALTERNATE(2E1)
```

- **Using the VARY Command:**

Use the ALTCONS operand on the VARY command to assign alternate consoles to your consoles.

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

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

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

## Displaying Information About Console Characteristics

To learn the current characteristics of the console, use the DISPLAY CONSOLES,A command. The display contains information about the system's use of consoles as well as information about each console's characteristics. Figure 3-6 shows the output of the command. The information in the boxes describes the message headings and general information about the system. Information about each console is described under the console screen in the figure.

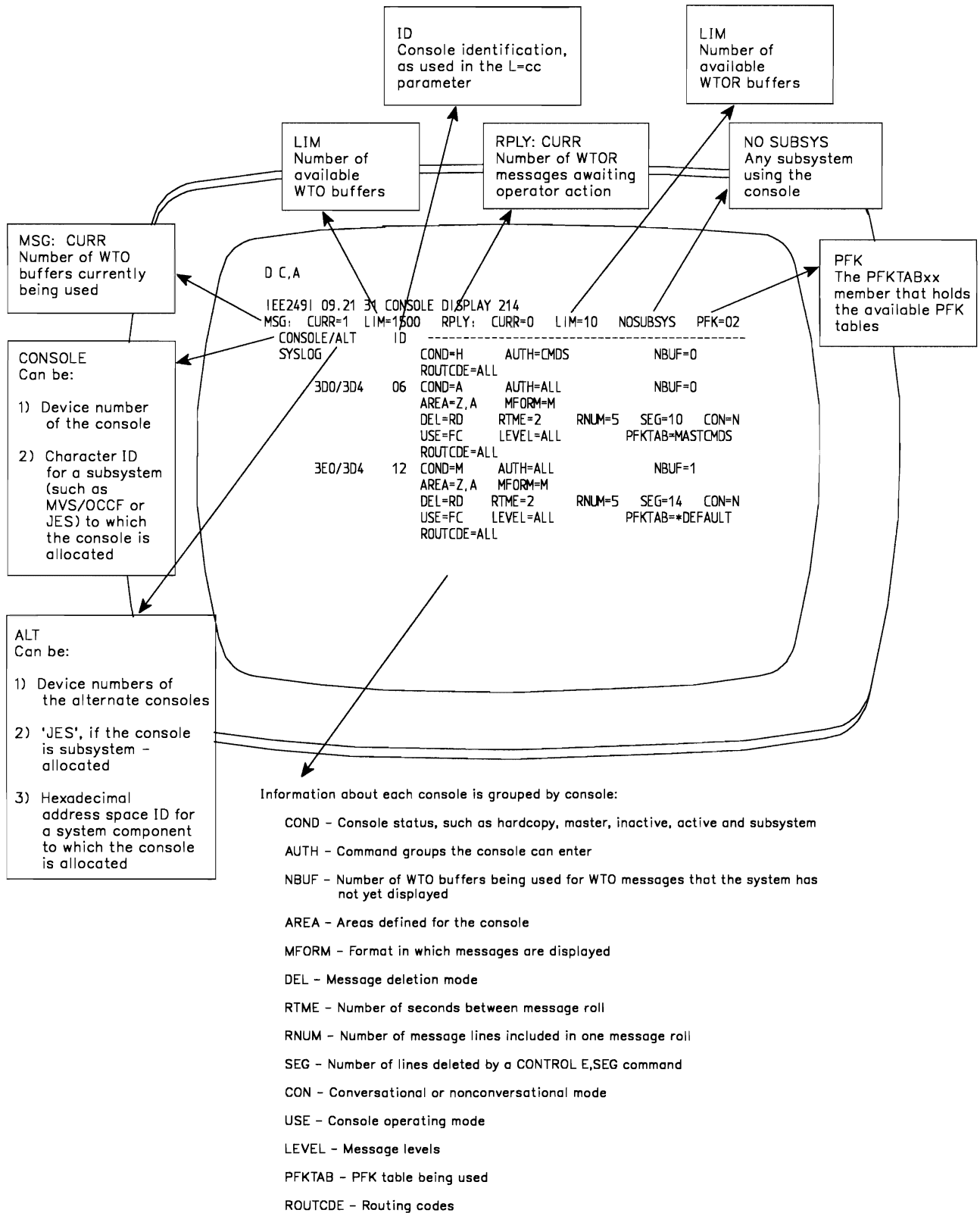


Figure 3-6. Output of the DISPLAY CONSOLES,A Command

## Defining the Console Operating Mode

MCS consoles can operate in one of three operating modes:

- Full-capability Mode

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.

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.

- Status Display Mode

A status display console has output capability only; it cannot be used to enter commands. The system uses the screen to receive status displays.

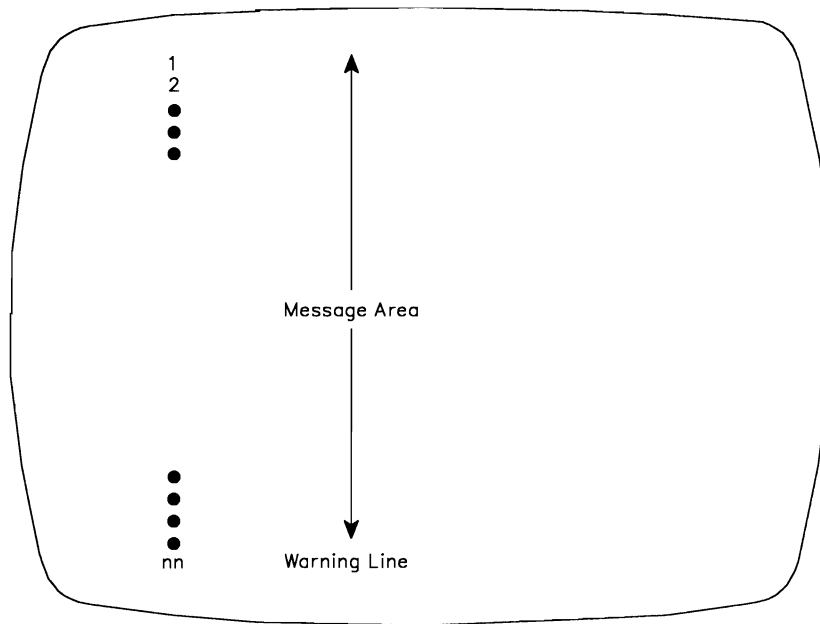
- Message Stream Mode

A message stream console also has output capability only; it cannot be used to enter commands. The system uses the screen to present general messages.

*Note:* In this book, the term output-only mode refers to status display mode and message stream mode.

The screens of 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 3-7 shows the 3277 model 2, in message stream mode. On all devices except the 3277 model 2, the warning line goes into the message area. 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 3-7. Format of the 3277 Model 2 Screen in Message Stream Mode**

The system gives you the following choices for operating mode:

|           |                 |
|-----------|-----------------|
| <b>FC</b> | Full-capability |
| <b>MS</b> | Message stream  |
| <b>SD</b> | Status display  |

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 A,REF command. In response to this command, the specifications appear in the entry area. You can change the specifications using the procedures described on page 2-43 under “Changing Information in the Entry Area.”

**You define the operating mode of a console by:**

- **Using the CONSOLxx Member of SYS1.PARMLIB:**

The USE parameter on the CONSOLE statement in SYS1.PARMLIB controls operating mode of a console.

*Example:* To define a console with a device number of 008 as a full-capability console, code the following statement in the CONSOLxx member:

```
CONSOLE DEVNUM(008) USE(FC)
```



- **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 id of 8 as a full-capability console, enter:

```
CONTROL V,USE=FC,L=8
```

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=8
```

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 member of SYS1.PARMLIB. 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 3-7.

*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 “Assigning and Controlling MCS Consoles” under the VARY command in Chapter 4). Otherwise, the VARY command is rejected.

*Example 3:* To return console 8 to full-capability mode, enter the following command from a full-capability console:

```
CONTROL V,USE=FC,L=8
```

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 following message appears in the instruction line:

```
IEE152I ENTER CANCEL D C,K
```

The display area specifications also return to the specifications established the last time the console was in full-capability mode.

## Using a Status Display Console

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. Figure 3-12 on page 3-44 shows the maximum display area sizes for the display consoles and the defaults.

### Controlling Displays on Status Display Consoles

Procedures for requesting, framing, holding, and terminating status displays (both static and dynamic) are described on page 3-42 under “Controlling System Status Displays.” 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 Command Groups” earlier in this chapter.

## Using a Message Stream Console

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 on page 2-43 under “Changing Information in the Entry Area.”

### 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, therefore, automatically removed from the screen.

*Note:* In unusual circumstances, the screen could become full of action messages for which no action can be taken. If this happens, you might need to take the console out of message stream mode to remove the unwanted action messages. Use the VARY command, described in Chapter 4, to take the console out of operator console mode, or use the CONTROL V,USE = command to change the console operating mode from MS (message stream) to SD (status display) and then back to MS.

## Controlling System Messages

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, if a number of such requests go unanswered for any length of time, system performance might suffer.

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.
- Using a master console cluster. With a master console cluster, you can divide the message traffic to the master console among several screens. For more information on setting up and using a master console cluster, see "Managing Messages with a Console Cluster" later in this chapter.
- Activating the action message retention facility so you can put the console in roll mode without losing action messages.

### Defining the Message Processing Facility (MPF)

The message processing facility (MPF) controls message processing. It controls the suppression and retention of messages, the user exits that gain control when certain messages are issued, and message presentation (that is, the color, intensity and highlighting of messages) at certain consoles.

Message suppression, retention, and presentation options and user exit information reside in MPFLSTxx members of SYS1.PARMLIB. At IPL, the system uses the MPFLSTxx member that the MPF(xx) parameter on the INIT statement in CONSOLxx indicates.

If you do not have an active MPFLSTxx member:

- MPF does not suppress messages.
- The action message retention facility retains all action messages, if it is active.
- No user exit except IEAVMXIT and IECCVXIT can gain control to process messages.
- Default options for message presentation are in effect.

*MPF Message Suppression:* MPF can suppress all messages except the following:

- Action messages (those with descriptor codes 1, 2, 3, or 11)
- Messages that require a reply (WTORs)
- Command responses with descriptor code 5 (except those generated by the MONITOR command)
- Command responses where MCSFLAG=RESP was specified on the WTO macro

For MPF to suppress messages, the hardcopy log must be active. The suppressed messages do not appear on any console; they do appear on the hardcopy log.

*Message Presentation:* Message presentation refers to the way the system uses color, intensity, and highlighting (including blinking, reverse-video, underscoring) to identify messages that require action. The presentation depends on the type of device you are using. Chapter 2 describes each display device that IBM supports as a console, including the default message presentation for the console.

The MPFLSTxx member in effect determines the color, intensity, and highlighting options of messages displayed on devices generated as 3270 model X and the 3279 console (models 2B, 3B, 2X, S2B, 3X, and S3G).

**You select the MPFLSTxx member that contains the message suppression, retention, and presentation options by:**

- **Using the CONSOLxx Member of SYS1.PARMLIB:**

The MPF parameter on the INIT statement in SYS1.PARMLIB specifies the MPFLSTxx member of SYS1.PARMLIB that is in effect for the consoles. If the parameter is not coded, the default is MPF(NO) and MPF is not in effect.

*Example:* To specify MPFLST03 as the MPF member for the system to use, code the following statement in CONSOLxx:

```
INIT MPF(03)
```

- **Using the SET Command:**

Issue the SET MPF command to change the MPFLSTxx member that the system is to use.

*Example:* To specify MPFLST03 as the MPF member for the system to use, enter:

```
SET MPF=03
```

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

## Controlling the Action Message Retention Facility

During its initialization, the system starts the action message retention facility. 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 MPFLSTxx member in effect.

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, then retrieve them in their entirety later by using the DISPLAY R command. About 1000 messages can be retained. 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 a MPFLSTxx member that contains the message retention options you want. See *Initialization and Tuning* for details of how to create MPFLSTxx members. 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 CONSOLxx Member of SYS1.PARMLIB:**

The AMRF parameter on the INIT statement in the CONSOLxx member controls whether the action message retention facility is on or off.

*Example:* To have the action message retention facility inactive at IPL, code the following statement in the CONSOLxx member:

```
INIT AMRF(N)
```

- **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
```

### 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.)

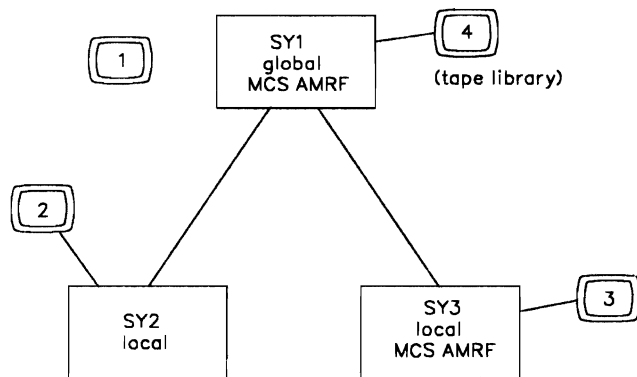
The DISPLAY R command allows you to display all outstanding action messages or a subset of these messages. For example, to display all the outstanding action messages at your console, enter DISPLAY R,M; to display all the outstanding critical eventual action messages (descriptor code 11), enter DISPLAY R,CE.

JES3 uses the dynamic support program (DSP) names as keynames to group messages by function. To help you keep track of messages, your application programmer can also group and name messages by function. The WTO macro in MVS/XA and the MESSAGE macro in JES3 allow your system programmer or application programmer to associate a 1 to 8 alphanumeric character or "keyname" with certain messages. Use the KEY operand on the DISPLAY R command to display all the outstanding action messages by keynames. For example, if your application programmer assigned the characters "PAYROLL" to all payroll application messages, you could list all the outstanding messages for payroll messages by entering DISPLAY R,M,KEY=PAYROLL or D R,KEY=PAYROLL.

The best way to describe how to use the DISPLAY R command at a JES3 complex is through an example. Assume a JES3 complex has the following identifiers:

- SY1        JES3 global
- SY2        JES3 local
- SY3        JES3 local
- 1         MCS console running MVS/XA and JES3 at SY1
- 2         MCS console running MVS/XA at SY2.
- 3         MCS console,running MVS/XA at SY3.
- 4         MCS console running MVS/XA at SY1. The console is controlling the tape library.

The example assumes that the MCS action message retention facility is active at two processors. The MCS action message retention facility at the global processor has a complete set of JES3 and MVS messages for the complex. To receive messages about a local application, the MCS action message retention facility is active at SY3.



- To see the texts and identification numbers of all outstanding action messages and WTORs destined for console 1, enter the following command at console 1:

```
DISPLAY R,M
```

- To learn the number of outstanding action messages with routing codes assigned to console 1, enter the following command at console 1:

```
DISPLAY R,ROUT=ALL
```

The message includes the total of outstanding action messages at the global processor and the two locals that are routed to console 1.

- To see all outstanding action messages in the JES3 complex, enter the following command at console 1 or 4:

```
DISPLAY R,M,CN=(ALL)
```

The message includes the total of outstanding action messages at the global processor and the two locals.

- To see all outstanding action messages for the local application running on SY3, enter the following command on console 3:

```
DISPLAY R,M
```

- To see a list of all the keynames of outstanding action messages that are destined for console 1, enter the following command at that console:

```
DISPLAY R,KEY
```

- If your installation has established the keyname TL for all tape library messages and you want to see a list of all outstanding action messages for the tape library console, enter the following command at console 4:

```
DISPLAY R,KEY=TL
```

## Activating WTO and WTOR User Exit Routines

The system programmer at your installation codes user exit routines that gain control when the system issues certain messages. A WTO user exit can change routing codes, descriptor codes, and message texts and perform other message processing; it can override MPF processing. Information on coding these user exits appears in *User Exits*.

The most effective message control involves coding and installing the user 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 REF command. The system displays in the entry area the status of the action message retention facility and the user exit IEAVMXIT and the limit of the number of WTO and WTOR buffers.

Your installation might have other exit routines to process messages. MPFLSTxx members of SYS1.PARMLIB contain the IDs of messages and the user exits that process these messages. To activate processing by these user exits, see “Defining the Message Processing Facility (MPF)” earlier in this chapter.

You can activate the user exit IEAVMXIT by:

- **Using the CONSOLxx Member of SYS1.PARMLIB:**

The UEXIT parameter on the INIT statement in SYS1.PARMLIB controls whether the WTO user exit IEAVMXIT is in effect. If you do not code this parameter, IEAVMXIT is active.

*Example:* To have the user exit IEAVMXIT inactive at IPL, code the following parameter on the INIT statement:

```
UEXIT(N)
```

- **Using the CONTROL Command:**

From the master console, use the UEXIT operand on the CONTROL command to control whether the user 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 command to see:

1. Which messages at your installation are being suppressed by MPF
2. Which action messages at your installation are not being retained by the action message retention facility
3. Which WTO user exits receive control for selected messages
4. The status of the general WTO user exit IEAVMXIT
5. What color, intensity, and highlighting capabilities are in effect

If you enter DISPLAY MPF,MSG, you receive the first four items on the above list for all messages that are defined in the MPFLSTxx member. If you enter DISPLAY MPF,COLOR, you receive only information about color, intensity, and highlighting for those 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. Figure 3-8 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 hardcopy log. To determine the routing codes each console receives, use the DISPLAY CONSOLES,A command. Figure 3-6 on page 3-14 shows the display that appears in response to this command.



| Message Routing 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                             |
| 29-41                | Reserved for IBM                                       |
| 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            |

**Figure 3-8. Message Routing Codes**

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, which 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 3-6 on page 3-14 shows the display that appears in response to this command.

**You define routing codes for a console by:**

- **Using the CONSOLxx Member of SYS1.PARMLIB:**

The `ROUTCODE` parameter on the `CONSOLE` statement in the `CONSOLxx` member controls the routing codes for a console. If you do not code this parameter, the system sends messages to the console without regard to the routing codes.

*Example:* To route all messages with routing codes 1, 2, 9, and 10 to the master console with device number 81D, code the following `CONSOLE` statement in the `CONSOLxx` member:

```
CONSOLE DEVNUM(81D) AUTH(MASTER) ROUTCODE(1,2,9,10)
```

*Note:* Operator's consoles do not need to receive routing code 11 (programmer information).

If you code `ROUTCODE(NONE)` for a master console, the system sends messages with routing codes 1 and 2. For all other consoles, `NONE` means the console receives no messages by routing codes alone.

- **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 with device number 81D, enter:

```
VARY 81D,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 might 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 messages 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:

|            |  |
|------------|--|
| <b>R</b>   | Write to operator (WTOR) messages are to appear                      |
| <b>I</b>   | Immediate action messages (descriptor codes 1 and 2) are to appear   |
| <b>CE</b>  | Critical eventual action messages (descriptor code 11) are to appear |
| <b>E</b>   | Eventual action messages (descriptor code 3) are to appear           |
| <b>IN</b>  | Informational messages are to appear                                 |
| <b>NB</b>  | Broadcast messages are <b>not</b> to appear                          |
| <b>ALL</b> | 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 3-6 on page 3-14 shows the display that appears in response to this command.

To display the routing codes and message levels that appear only on hardcopy log **and not on any console**, issue the DISPLAY CONSOLES,HCONLY command.

**You define the level of messages for a console by:**

- **Using the CONSOLxx Member of SYS1.PARMLIB:**

The LEVEL parameter on the CONSOLE statement in the CONSOLxx member controls the message levels for a console.

*Example 1:* To direct only WTOR messages and immediate action messages to console with device number 0C6, code this statement in CONSOLxx:

```
CONSOLE DEVNUM(0C6) LEVEL(R,I)
```

To understand the relationship between the routing codes and the message levels assigned to a console, study the following example:

*Example 2:* To assign to the console with device number 81D, the informational messages directed to the tape libraries (routing code 5) and disk libraries (routing code 6), code this statement in the CONSOLxx member:

```
CONSOLE DEVNUM(81D) ROUTCODE(5,6) LEVEL(R,IN)
```

- **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 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 in the hardcopy log. 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=10
```

## 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” in Chapter 2 describes the format of messages and describes the optional information that the system can include with each message:

|   |  |
|---|--|
| T | A time stamp                                   |
| S | The name of the system that issued the message |
| J | The jobname/job id of its issuer               |
| M | Only the message text displays                 |

You request that additional information precede each message the system sends a console by:

- **Using the CONSOLxx Member of SYS1.PARMLIB:**

The MFORM parameter on the CONSOLE statement in the CONSOLxx member controls the format of messages the system sends to a console.

*Example:* To request that the system add a time stamp, the name of the system that issued the message, and the jobname or id of its issuer, code the following parameter on the CONSOLE statement:

```
MFORM(J,S,T)
```

The format of a message that includes the above information is:

```
Time stamp System name Jobname/id Message text
```

- **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.

## 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 member of SYS1.PARMLIB. 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
```

Selector pens also use the definitions in PFK tables.

If you are controlling JES3 through an MCS console, your PFK definitions can reside in a JES3 PFK table. These definitions are in effect only for JES3. To define PFKs for JES3, see *JES3 Initialization and Tuning*.

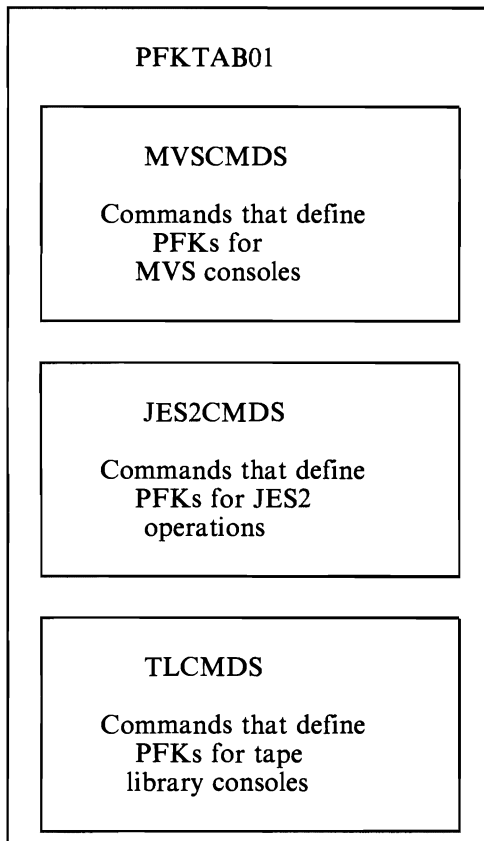
## An Example of Defining a PFK Table

The best way to explain the steps you take to define and activate a PFK table is through an example. Assume that the CON parameter in the IEASYSxx member of SYS1.PARMLIB identifies CONSOL01 as the appropriate CONSOLxx member for the system to use for this IPL. Assume, further, that your installation has not defined any PFK tables; you have been using the system defaults for PFKs 1 through 9. In the example, the PFK table that is created is named MVSCMDS; it resides in member PFKTAB01 of SYS1.PARMLIB. The procedure for creating MVSCMDS and activating it for your console is as follows:

| Procedure  | Coding of SYS1.PARMLIB Member  |
|--|--|
| Create the PFK table named MVSCMDS   | Assign commands to PFK(nn) definitions in PFKTAB01, where nn is the PFK number.      |
| Associate MVSCMDS with a console   | Specify PFKTAB(MVSCMDS) on the CONSOLE statement in CONSOL01 member of SYS1.PARMLIB. |
| Activate the PFKTAB01 member of SYS1.PARMLIB that contains the PFK table named MVSCMDS | Specify PFK(01) on the INIT statement in the CONSOL01 member of SYS1.PARMLIB         |

When you IPL the MVS/XA system, the system uses MVSCMDS to define the PFKs on your console.

Use the same PFKTAB01 member to hold the PFK tables for your JES2 and tape library operators. Figure 3-9 shows PFKTAB01 member of SYS1.PARMLIB. It contains three tables: MVSCMDS, JES2CMDS, and TLCMDS.



**Figure 3-9. PFKTAB01 Member of SYS1.PARMLIB.**

You can use some MVS/XA 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 issue 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='')
```

## Deleting Messages from the Console Screen

As programs execute during system operation, the message area of the screen gradually fills with messages. To make room for more messages, delete nonaction messages and messages for which action has been taken.

You can delete messages from the screen *manually* using the CONTROL E command, the cursor, or the selector pen, or you can request that the system delete messages *automatically* in automatic mode, roll mode, or roll-deletable mode. **Note:** Procedures for deleting system status displays are described under “Erasing Status Displays.” on page 3-48.

Manual message deletion by selector pen or cursor is useful for providing screen space quickly; manual message deletion using the CONTROL E command is more flexible for deleting groups of messages selectively. Automatic message deletion is most useful when messages appear frequently, and when it is important that messages not back up on system queues while waiting for screen space.

## Defining Conversational or Non-conversational Message Deletion Mode

Message deletion, like command entry, can be either *conversational* or *nonconversational*. In conversational mode, the system allows you to verify your deletion request before it removes the messages from the screen. In nonconversational mode, your deletion request causes the designated messages to be deleted immediately.

To display the current values for message deletion, enter the CONTROL S,REF command.

**You establish conversational or nonconversational message deletion by:**

- **Using the CONSOLxx Member in SYS1.PARMLIB**

The CON parameter on the CONSOLE statement in the CONSOLxx member of SYS1.PARMLIB controls the mode of message deletion. The default is CON(N), which indicates nonconversational message deletion mode.



*Example:* To request that designated messages appearing on console 2 be deleted without your verification, code the following parameter on the CONSOLE statement for the console:

```
CON(N)
```

- **Using the CONTROL Command**

Use the CONTROL S,CON command to change the message deletion mode.

*Example:* To request that designated messages appearing on console 2 be deleted without verification, enter:

```
CONTROL S,CON=N,L=2
```

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

## Deleting Messages Manually in Conversational Mode

Using conversational message deletion, you can verify all messages that you have selected for deletion by cursor, selector pen, or CONTROL command. To delete messages in conversational mode:

1. Follow the procedures described for nonconversational message deletion in “Deleting Messages Manually in Nonconversational Mode” or use the CONTROL E command.
2. After you enter the deletion request, a vertical line appears in position three of each message to be deleted (all other vertical lines are temporarily removed from the screen), and the following message appears in the instruction line:

```
IEE157E DELETION REQUESTED
```

Message line numbers are written for all messages on the screen, and the deletion request appears in the entry area in CONTROL command form.

If your deletion request is made by selector pen or cursor, or is a CONTROL E,nn,nn or CONTROL E,SEG command, the CONTROL E,nn,nn, command form appears in the entry area. For example, if SEG has been defined as 10, and you enter CONTROL E,SEG, the following appears:

```
CONTROL E,1,10
```

If you made your deletion request by positioning the selector pen or the cursor on the fifth message line, that message and all nonaction messages above it are marked with vertical bars, and the following appears in the entry area:

```
CONTROL E,1,5
```

If your deletion request was a CONTROL E,F command, the following appears in the entry area:

```
CONTROL E,F
```

## Verifying the Deletion Request

You must verify whatever deletion request appears in the entry area. To verify the request:

1. Check the command in the entry area and the messages marked with vertical bars. Make certain that the indicated messages are the ones that you want to delete. If you want to make any changes, use the procedures described in “Changing Information in the Entry Area” on page 2-43. If you want to cancel the deletion request, perform a CANCEL action.
2. When the command is in the proper form, perform an ENTER action. If you originally used the selector pen to designate the messages to be deleted, you can enter the command by positioning the selector pen over the same line again.

The messages selected for deletion are removed from the message area, and any remaining messages move up toward the top of the screen. Whether you enter the command in the entry area or cancel it, message line numbers are removed, and any flags that existed before the request are restored. Also, the entry area is blanked out, and the cursor is repositioned to the first data entry position.

## Deleting Messages Manually in Nonconversational Mode

When you delete messages, all messages below the deleted messages move toward the top of the screen to fill the lines that the deleted messages occupied. As message lines become available at the bottom of the message area, any messages waiting for screen space move into them. In addition to deleting the messages, you also blank out the entry area and reposition the cursor to the first data entry position.

To delete a nonaction message *and all nonaction messages above it* with the selector pen, position the pen anywhere within the nonaction message to be deleted. To delete a nonaction message *and all nonaction messages above it* with the cursor, move the cursor to any position within the nonaction message to be deleted. Perform an ENTER action.

You can also use the CONTROL E command to delete action and nonaction messages. You can delete one message, a segment of messages, or flagged messages. (The “flag” is a vertical or horizontal bar in position 3.) Flags are used in conversational message deletion to mark messages slated for deletion so that you can double check the messages before they disappear.

*Example 1:* To delete the message at line 10, which appears on the screen as follows:

```
10 IEE334I HALT EOD SUCCESSFUL
```

enter:

```
CONTROL E,10
```

*Example 2:* To delete the nonaction messages from a segment of messages, enter:

```
CONTROL E,SEG or CONTROL E
```

A CONTROL S command defines a segment as described later under “Defining the Scope of Message Deletion.”

*Example 3:* To delete the nonaction messages in message lines 4-10, enter:

```
CONTROL E,4,10
```

*Example 4:* To delete all flagged messages (messages marked with a vertical or horizontal bar in position 3), enter:

```
CONTROL E,F
```

*Note:* Message numbers can help you determine which messages you want to delete; see “Numbering Messages to Aid Manual Message Deletion” later in this section.

## Defining the Scope of Message Deletion

You can delete groups, or “segments,” of messages by issuing the CONTROL E,SEG command. The number of messages that the system deletes at one time depends on one of the following:

- The value you set on the CONTROL S,SEG command
- The value established with the SEG parameter on the CONSOLE statement in the CONSOLxx member of SYS1.PARMLIB
- The IBM default

The IBM default depends on the type of console. The default values for the consoles that IBM supports appear in Figure 3-10.

| Screen size | Devices  | SEG |
|-------------|--|-----|
| 20 x 80     | 3278-2A, 3279-2C   | 8   |
| 24 x 80     | 3180-1, 3278-2, 3279-2A, 3279-2B,<br>3279-2X, 3279-S2B, 3290 | 10  |
|             | 3277-2   | 9   |
| 27 x 132    | 3180-1, 3278-5, 3290   | 12  |
| 31 x 80     | 3290   | 13  |
| 31 x 160    | 3290   | 14  |
| 32 x 80     | 3180-1, 3278-3, 3279-3A, 3279-3B,<br>3279-3X, 3279-S3G, 3290 | 14  |
| 43 x 80     | 3180-1, 3278-4, 3290   | 19  |
| 50 x 106    | 3290   | 23  |
| 62 x 80     | 3290   | 29  |
| 62 x 160    | 3290   | 28  |

**Figure 3-10. Default Values for SEG Parameter in CONSOLxx**

To display the current values for message deletion, issue the CONTROL S,REF command.

You can establish the scope of message deletion by:

- **Using the CONSOLxx Member of SYS1.PARMLIB**

The SEG parameter on the CONSOLE statement in the CONSOLxx member controls the number of lines that will be deleted when you issue the CONTROL E,SEG command.

*Example:* To allow the CONTROL E,SEG command to delete 10 lines, code the following parameter on the CONSOLE statement for the console:

```
SEG(10)
```

- **Using the CONTROL Command**

Use the CONTROL S,SEG command to change the number of lines (*nn*) that will be deleted when you issue the CONTROL E,SEG command.

*Example:* To allow the CONTROL E,SEG command to delete 10 lines for the console on which you enter the CONTROL command, enter:

```
CONTROL S,SEG=10
```

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

## Deleting Retained Action Messages with the CONTROL Command

Use the CONTROL C command to delete individual outstanding action messages or a range of messages that have been retained by the action message retention facility. You will want to delete messages when you have already performed the action but the messages are still marked as outstanding. Make sure you have performed the requested action because, once you delete these messages, you cannot display them again.

Delete retained action messages by **type** and **message id**. To determine the identification numbers of the messages, issue the DISPLAY R command. Then, use the ids on the CONTROL C command.

|           |   |
|-----------|---|
| <b>A</b>  | Deletes immediate action messages, eventual action messages, or critical eventual action messages |
| <b>I</b>  | Deletes immediate action messages   |
| <b>E</b>  | Deletes eventual action messages  |
| <b>CE</b> | Deletes critical eventual action messages   |

*Example 1:* 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 ids of the messages and then use CONTROL C,E to delete the three messages.

```
DISPLAY R,E  
CONTROL C,E,id,id,id
```

*Example 2:* To delete all immediate action and eventual action messages that have been retained with ids from 0 to 99999, enter:

```
CONTROL C,A,0-99999
```

## Numbering Messages to Aid Manual Message Deletion

Message numbers are useful in determining the range of messages to delete with the CONTROL E,nn command and the CONTROL E,SEG command. You can request that message numbers appear in positions one and two of the message area lines. Consecutive numbers appear for each message line, including continuation lines, for all message area messages except status displays. A numbered message appears as follows:

```
12 IEE041I THE SYSTEM LOG IS NOW ACTIVE
```

Use the CONTROL D,N and CONTROL E,N commands to display and erase message numbers.

Specify the CONTROL D,N to display consecutive numbers in character positions one and two of each message area line containing a message; the numbers are removed from the screen after the operator deletes a message or performs a cancel action. Add the HOLD operand to this command if you want the remaining messages renumbered as you delete messages.

Specify CONTROL E,N to remove from the screen the message numbers requested by the CONTROL D,N,HOLD command.

*Notes:*

1. *Automatic message deletion (in automatic mode, roll mode, or roll-deletable mode) stops message numbering requested by the CONTROL D,N,HOLD command.*
2. *Because a display console screen can be "burned" by the number images, it is recommended that you do not have the messages numbered all of the time. When you are in conversational mode and delete messages by the CONTROL command, all messages are temporarily numbered so that you can verify that you have entered the correct delete command.*

*Example 1:* To request continual message numbering, enter:

```
CONTROL D,N,HOLD
```

*Example 2:* To cancel continual message numbering, enter:

```
CONTROL E,N
```

## Temporarily Suspending the Screen Roll

You might want to suspend the message roll on your screen. For example, you might want to copy some information from the screen or consult a messages reference book. Press PFK 5 to immediately stop the messages from rolling. (At IPL, PFK 5 is assigned the command CONTROL S,DEL=N.) Then, press PFK 6 to place your screen in roll-deletable mode and prevent message backup. (PFK 6 is defined at IPL as CONTROL S,DEL=RD.) Note that in this example, the IBM defaults are in effect.

## Defining Automatic Message Deletion

Automatic message deletion is a way to remove messages from the screen without your intervention. There are three modes of automatic message deletion: automatic mode, roll mode, and roll-deletable mode. In all cases, deletion occurs only when the message area is full and messages are waiting to be displayed.

In *automatic mode*, all flagged messages are removed whenever the message area becomes full, or when a status display is overlaying messages in the bottom portion of the message area. Flagged messages are the only messages removed under automatic mode. These messages include:

- Action messages for which the action has been taken
- System or problem program messages that are marked deletable by the issuer
- Messages that are indicated as deletable at job step end
- WTOR messages that have been answered
- WTOR messages that have not been answered but that are associated with a job step that has ended

In *roll mode*, a specified number of flagged messages are removed (or “rolled off”) when a specified time interval elapses. Roll mode is particularly useful for monitoring message traffic.

*Roll-deletable mode* is the same as roll mode except that action messages are not removed; they accumulate at the top of your screen. You can then delete the action messages one at a time, either by using the CONTROL E command or placing the cursor under the “\*” or “@” that precedes the message and pressing ENTER.

When a console is operating in roll mode or roll-deletable mode, messages are *not* numbered. Instead, there is a two-digit number in the first new message line on the screen after each roll. This number indicates the number of messages waiting to be displayed (including any messages that are hidden by a status display). If the number of messages waiting to be displayed exceeds 99, AA is displayed.

The system gives you the following choices for automatic message deletion:

**DEL(Y)** Automatic message deletion mode is in effect.  
**DEL(N)** Only manual message deletion can be used.  
**DEL(RD)** Roll-deletable mode of message deletion is in effect.  
**DEL(R)** Roll mode of message deletion is in effect.  
**RNUM** The number of lines in a message roll  
**RTME** The number of seconds between message rolls

If your system programmer does not define automatic message deletion in the CONSOLxx member, IBM supplies the following values:

- Roll deletable mode of deletion - **DEL(RD)**
- 5 lines in a message roll - **RNUM(5)**
- 2 seconds between message rolls - **RTME(2)**

To see what message deletion specifications are in effect, enter the CONTROL S,REF command. The system responds with the current console specifications in CONTROL command form in the entry area. You can then change the specifications by following the procedures described earlier under “Changing Information in the Entry Area” on page 2-43.

**You control automatic deletion specifications by:**

- **Using the CONSOLxx Member of SYS1.PARMLIB**

The DEL, RNUM, and RTME parameters on the CONSOLE statement in the CONSOLxx member control automatic message deletion:

*Example 1:* To request that a console have automatic message deletion mode, code the following parameter on the CONSOLE statement for the console:

```
DEL(Y)
```

*Example 2:* To request that roll mode go into effect and that ten messages be rolled every 30 seconds, code the following parameters on the CONSOLE statement for the console:

```
DEL(R) RNUM(10) RTME(30)
```

- **Using the CONTROL Command**

Use the CONTROL S,DEL command to change the message deletion specifications for the duration of the IPL.

*Example 1:* To request automatic message deletion mode, enter:

```
CONTROL S,DEL=Y
```

If there are no messages marked with a vertical line when a message is waiting to be displayed, the following message appears in the warning line:

```
IEE159E MESSAGE WAITING
```

You must then delete messages manually by using the CONTROL command, cursor, or the selector pen to make room for the waiting message.

*Example 2:* To request that roll mode go into effect and that ten messages be rolled every 30 seconds, enter:

```
CONTROL S,DEL=R,RNUM=10,RTME=30
```

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

## Finding Appropriate Message Deletion Specifications

Because system workload can vary, you must manipulate the RNUM and RTME values until messages stay on the screen long enough to be read but not long enough to cause messages to back up. First, find the appropriate RNUM and RTME values on the CONTROL S command as follows (the test values used are for a 3277 model 2 console):

1. Have the system running at the normal load.
2. Issue CONTROL S,DEL = RD,RNUM = 5,RTME = 2 on the console. This command causes five messages to roll every two seconds. If fewer than five messages are waiting to be displayed, only the number waiting rolls.
3. Watch the two-digit number that appears on the left side of the screen. This number indicates the number of messages waiting to be displayed on this console. If AA is displayed, over 99 messages are backed up. When the screen is updated, this number appears on the first message line added to the screen, causing the number to move up and down the left side of the screen.
4. If the waiting message count keeps rising, change the RNUM and RTME specifications (try raising RNUM by 5 and/or decreasing RTME). If the count goes to AA and you think a high number of waiting messages will cause problems for the system, issue CONTROL S,RNUM = 19,RTME = 1 to reduce the message backlog. Then, reset RNUM to 5 and RTME to 2.
5. Keep adjusting the RNUM and RTME values with the CONTROL S command until the messages are readable and there is little message backlog. An average backlog of zero to five messages is acceptable, but zero is the best.
6. Once the console looks as if it can keep up with the message traffic, issue:

```
CONTROL E,D      to remove any out-of-line displays
CONTROL A,NONE   to remove any out-of-line areas
```

7. To test the specifications, issue, in quick succession, the following commands:

```
DISPLAY C,L
DISPLAY M
DISPLAY U
DISPLAY C,L
DISPLAY U
DISPLAY M
```

These commands should cause many messages to back up on the console. Try to get the count of waiting messages to reach AA. If the RNUM and RTME specifications are correct, the count of waiting messages should start to drop and reach zero in a few minutes without causing the system any problems. If this does not happen, change the specifications and test them again.

Once you have decided on the best values for the RNUM and RTME operands, code the RNUM and RTME parameters on the CONSOLE statement in CONSOLxx member of SYS1.PARMLIB.



To illustrate how you would code the values in CONSOLxx, assume for a console with device number 0C6, that you want to define roll-deletable mode with five messages being deleted every 30 seconds. Also, you want each message to display the jobname/job id of its issuer. Code the statement in CONSOLxx as follows:

```
CONSOLE DEVNUM(0C6) DEL(RD) RTME(30) MFORM(J)
```

Keeping action messages on the screen for too long reduces the screen space available for informational messages and increases the number of messages backing up for the console. Therefore, perform a requested action as soon as possible so the system can remove the corresponding message. If a message asks for a reply but you want to respond later, remove the message from the screen. You can find the reply id later by issuing a DISPLAY R command. If the action message retention facility is active, you can run the console in roll mode because the system retains action messages and you can retrieve them later with a DISPLAY R command.

If, after following the procedure described earlier for finding correct values for RNUM and RTME, you still cannot get messages on the console to remain readable without causing waiting messages to back up, consider ways of reducing the message traffic to the console. Think about directing fewer routing codes or message levels to the console, or using the message processing facility to suppress unwanted messages. See “Controlling System Messages” on page 3-19 for more information.

## Checking the Message Deletion Specifications

To determine which specifications are in effect, enter:

```
CONTROL S,REF or CONTROL S
```

In response to the CONTROL S,REF command the following display appears in the entry area:

```
K S,DEL=x,SEG=xx,CON=x,RNUM=xx,RTME=xxx,MFORM=(x)
```

where K is the abbreviation for CONTROL.

In the actual display, each x is replaced by the specifications currently in effect. The display can reflect the values defined in CONSOLxx member, or those defined by the CONTROL command, or the IBM defaults. You can change any specification at this time by following the procedures for changing information in the entry area described earlier in “Changing Information in the Entry Area” on page 2-43.

When you enter CONTROL S,REF and, after examining the display, want to retain the values as they are shown, perform a CANCEL or ENTER action.

## Controlling System Status Displays

A status display is a formatted, multiple-line display of information about some part of the system. It is written to consoles in full-capability or status display mode in response to certain commands, such as DISPLAY or CONFIG, or certain subsystem commands.

You can use status displays to obtain information, such as the status of system devices and the identification of the jobs active in the system, that can help you decide how best to use system resources.

On consoles in full-capability or status display operating mode, status displays are usually presented in display areas set aside for their use. If you do not define one or more display areas, status displays appear in the general message traffic. The information you need in the status display could, therefore, roll off the screen before you can find it.

A status display is either static or dynamic. A display is static if it remains the same until it is removed from the screen; a dynamic display is created by the TRACK command and is updated by the system each time a preset time interval elapses. The different procedures required for dynamic displays are described under “Establishing Dynamic Status Displays” on page 3-49.

Think of display areas as addressable blocks of message area lines that overlay parts of the message area. On consoled operating in full-capability mode, the blocks of lines assigned to a display area function as message area lines until a status display is written to them. When this happens, any messages occupying the lines of the display area are replaced by the status display. If these messages are general operator messages, they are not lost; they reappear, higher up on the screen if screen lines above the status display become available for their use. When messages are overlaid by a status display and the device is not in roll or roll-deletable mode, the following message appears in the warning line:

```
IEE160I UNVIEWABLE MESSAGE
```

Erasing a display from a display area restores the display area lines to general message use. Unless there are other status displays above the restored display area, any messages that were overlaid by the erased display reappear. If there are status displays in display areas above the restored area, the lines of the restored area remain blank. General messages are not displayed in any line in the message area below a display area containing a status display.

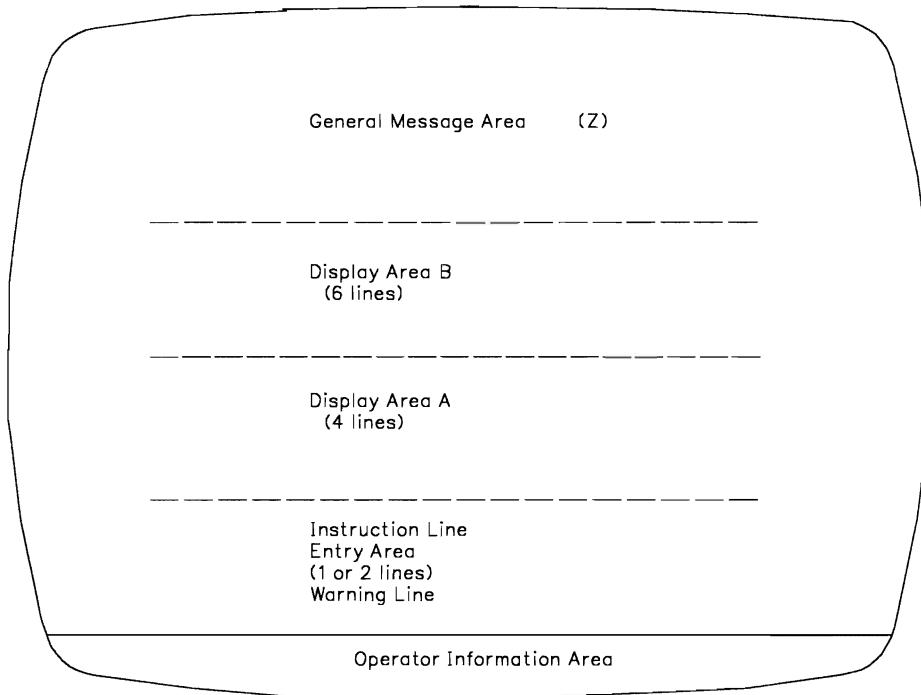
It is good practice to erase status displays from display areas when the displays are no longer required. The undisplayed frames of lengthy status displays can fill up buffer space that might be required for other messages.

### Setting Up Display Areas on a Console

You can control the number of display areas on a console screen and the size of each area. You can specify up to 11 different display areas, the location of the areas, and the number of screen lines in each area. Display areas are defined from the bottom of the message area to the top of the area. Each area can consist of four or more screen lines designated to receive the status displays.

Figure 3-11 shows the screen format for a display console in full-capability mode when two typical display areas are defined for the screen. The first (bottom-most) area has four lines, and the second has six lines. Status displays can be routed (using the location operand of the

DISPLAY, TRACK, CONFIG, and MSGRT commands) to area A or B, or to the general message area.



**Figure 3-11. Sample Screen Format Showing Two Display Areas**

The system assigns the alphabetic display area identifiers. The bottom-most area is assigned identifier A and additional areas are assigned identifiers in alphabetic order, working toward the top of the screen. The identifier Z always refers to the portion of the message area that is not assigned to a display area. You can obtain the console identifier and display area identifiers for each console by issuing a DISPLAY CONSOLES command.

The AREA parameter on the CONSOLE statement in CONSOLxx member of SYS1.PARMLIB controls the number of display areas and their sizes. If you do not code the AREA parameter, the system defines two display areas for status display consoles and one display area for full-capability consoles. The number of lines in each area depends on the type of device. Figure 3-12 shows the maximum display area sizes for all devices that MVS/XA supports as consoles. Included in parentheses in the column “Full-Capability” is the size (14 lines) the system assigns if the AREA parameter is not coded for a console operating in full-capability mode. Included in parentheses in the column “Status Display” are the two values the system assigns if the AREA parameter is not coded for a console operating in status display mode. The two default values represent areas B and A.

| Screen Size | Device Type  | Full-Capability | Status Display | Message Stream |
|-------------|--|-----------------|----------------|----------------|
| 20 x 80     | 3278-2A, 3279-2C   | 16(14)          | 20(10,10)      | 19             |
| 24 x 80     | 3180-1, 3278-2,<br>3279-2A, 3278-2B<br>3279-2X, 3279-S2B<br>3290 | 20(14)          | 24(12,12)      | 23             |
|             | 3277-2   | 19(14)          | 23(11,12)      | 23             |
| 27 x 132    | 3180-1 3278-5<br>3290  | 24(14)          | 27(13,14)      | 26             |
| 31 x 80     | 3290   | 27(14)          | 31(15,16)      | 30             |
| 31 x 160    | 3290   | 28(14)          | 31(15,16)      | 30             |
| 32 x 80     | 3180-1, 3278-3<br>3279-3A, 3279-3B<br>3279-S3G, 3290             | 28(14)          | 32(16,16)      | 31             |
| 43 x 80     | 3180-1, 3278-4<br>3290   | 39(14)          | 43(21,22)      | 42             |
| 50 x 106    | 3290   | 46(14)          | 50(25,25)      | 49             |
| 62 x 80     | 3290   | 58(14)          | 62(31,31)      | 61             |
| 62 x 160    | 3290   | 59(14)          | 62(31,31)      | 61             |

**Figure 3-12. Display Area Sizes for Consoles**

To check the display area specifications in effect at any time, enter CONTROL A,REF. In response, the system displays the specifications in the entry area in CONTROL command form. For example, if three display areas of eight, four, and four lines are defined for a screen, entering CONTROL A causes the following to appear in the entry area:

```
K A ,08 ,04 ,04
```

where K is the abbreviation for CONTROL.

You can then change the specifications by following the procedures described earlier under “Changing Information in the Entry Area” on page 2-43. To change the specification in the above example to two display areas of four and six lines, position the cursor at the 8, type in 4,6; blank out the remainder of the area, and enter the command.

**To define display areas:**

- **Using the CONSOLxx Member of SYS1.PARMLIB**

The AREA parameter on the CONSOLE statement in the CONSOLxx member of SYS1.PARMLIB controls define display areas for your status displays.

*Example:* Figure 3-13 shows two display areas on a console operating in full-capability mode. Assume that display area B has 6 lines and area A has 4 lines. Such a display would have the following definition in CONSOLxx:

```
AREA(4,6)
```

- **Using the CONTROL Command**

Use the CONTROL A command to change the number and size of display areas on display consoles. The effect of this command lasts only for the duration of the IPL.

*Example:* To set up two display areas at your console, B with 6 lines and A with 4 lines, enter:

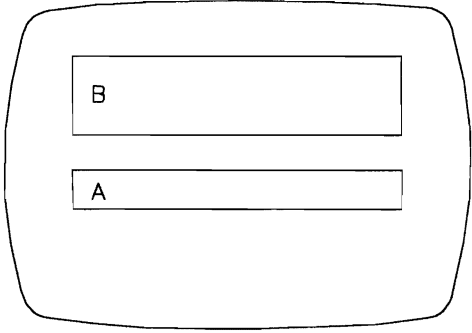
```
CONTROL A,4,6
```

Figure 3-13 shows how to set up a static display and a dynamic display in the display areas you defined for status display consoles. The figure gives the commands you use to set up the displays, the functions they perform, and the topics under which these functions are described in this chapter. If the console in the figure was operating in status display mode, it would not be able to accept any commands. In this case, you would issue the command from another console and include the L= operand to route the action of the command to the console in status display mode.

| Command     | Function   | Described Under                            |
|-------------|--|--|
| TRACK A     | Establishes a dynamic display in area A to provide a constant indication of system activity.         | "Establishing Dynamic Status Displays"     |
| DISPLAY U   | Establishes a static status display in area B.   | "Routing Status Displays to Display Areas" |
| CONTROL D,F | Frames the display in area B. Frames the dynamic display in area A if there is no display in area B. | "Framing Status Displays"                  |
| CONTROL E,D | Erases the display in area B.  | "Erasing Status Displays"                  |

**Result**



The diagram illustrates the result of the commands. It shows a large rounded rectangular frame containing two smaller rectangular boxes stacked vertically. The top box is labeled 'B' and the bottom box is labeled 'A', representing the static and dynamic display areas respectively.

**Figure 3-13. Example of Setting Up and Using Display Areas**

## Routing Status Displays to Display Areas

You can route many of the displays that normally come to your console to another console's general message area, to a display area on another console, or to a specific display area on your console. There are three ways to route status displays of the DISPLAY, TRACK, and CONFIG commands:

- The location operand (L = ) operand of these commands
- The MSGRT parameter on the CONSOLE statement in the CONSOLxx member
- The MSGRT command

Use the location operand on the DISPLAY and CONFIG commands to route one occurrence of a display. Use the MSGRT command to route all displays of these commands for the duration of the IPL or until you issue another MSGRT command. For a more permanent routing assignment, use the MSGRT parameter in the CONSOLxx member.

Displays of the TRACK command continue for the duration of the IPL, or until you issue another routing request.

If you use the location operand with the DISPLAY, TRACK, or CONFIG command, the system presents the status display on the console (L = cc) in the display area (L = a), or in the display area on the console (L = cca) that you specify. If the area is already occupied, the new status display replaces the old display (except that dynamic displays are not replaced). If you do not use the location operand, the system presents the display according to the MSGRT instructions in effect. If there are no MSGRT instructions in effect, the system presents the display in the issuing console's lowest unoccupied display area or, if all areas are full, in the area containing the oldest display. If no display area can be found (because the screen has no display areas or because all areas are being used by dynamic displays), the system presents the status display in the portion of the message area not assigned to a display area (area Z).

### Using the Location Operand

Use the location operand of the DISPLAY, TRACK, and CONFIG commands to route status displays to specific display areas on the console you are using or to route displays to output-only consoles. The routing location operand specifying a console other than the requesting console can be issued only through the master console or through consoles that have CONS (console control) authority.

*Example:* To request a display of active jobs in display area B of console 12, enter:

```
DISPLAY A,LIST,L=12B
```

### Using the CONSOLxx Member of SYS1.PARMLIB

Use the MSGRT parameter on the CONSOLE statement in SYS1.PARMLIB to route displays from the DISPLAY, MONITOR, or CONFIG commands. When the MSGRT parameter controls the routing of a display, each time that display would normally appear at the issuing console, it will appear at the console or display area you specify in the location operand on the MSGRT parameter.

One MSGRT parameter can define routing instructions for all or any combination of the commands for which MSGRT defines routing instructions. You can define different routing instructions for each console in the system.

*Example 1:* To direct the displays of the DISPLAY and CONFIG commands that are entered on the console with device number 09E to different areas of console 4, code the following statement in CONSOLxx:

```
CONSOLE DEVNUM(09E) MSGRT('D=A,L=4A')('CF,L=4B')
```

Note that the operands on the MSGRT parameter on the CONSOLE statement in CONSOLxx are not always the same as the operands on the MSGRT command. For example, to route the DMN command to console 6, you would code MSGRT('D=DMN,L=6') on the CONSOLE statement; or, you would enter MSGRT(D=D,L=6).

### **Using the MSGRT Command**

Use the MSGRT command to route the displays from the DISPLAY, CONFIG, MONITOR, and TRACK commands. You can enter multiple command routings on one MSGRT command by enclosing the operands in parentheses and separating them by commas.

*Example 1:* To direct the displays of the DISPLAY and CONFIG commands to different areas of console 4, enter:

```
MSGRT (D=A,L=4A),(CF,L=4B)
```

The effect of the MSGRT command lasts for the duration of the IPL.

*Example 2:* To display the MSGRT command operands currently in effect, enter:

```
MSGRT REF
```

In response to this command, a display of MSGRT operands appears in the entry area. You can change the routing instructions at this time by changing the command form that appears in the entry area. See "Changing Information in the Entry Area" on page 2-43.

If no display area id is specified on the routing instruction, an asterisk is displayed in the area position of the output. Also, if, after you enter MSGRT REF, the operand CONT appears in the display, there are additional routing instructions. To examine them, perform the ENTER action, or clear the entry area and enter:

```
MSGRT CONTn
```

This command specifies the continuation of the *n*th line of the display issued, where *n* is a decimal number from 1 to 9. If there are not enough additional routing instructions to continue the *n*th display line, the system displays the last possible line.

*Example 3:* To stop the message routing you established with the MSGRT command, enter the applicable MSGRT command without the location operand. For example, to remove the routing instructions for the TRACK and CONFIG commands established in the earlier example, enter:

```
MSGRT (TR=A),(CF)
```

You can remove all routing instructions defined by the MSGRT command by issuing the following form of the MSGRT command:

```
MSGRT NONE
```

## Framing Status Displays

Depending on the size of the display and the number of lines available in the message area or display area, status displays can be divided into two or more frames. The control line (first line) of each display contains a frame number, and the last frame is marked FRAME LAST.

*Note:* You can only frame forward through a display; to re-examine previous frames of the display, you must request a new display.

- **Framing with the CONTROL Command**

To move a status display forward to the next frame, use the CONTROL D,F command. If you omit the location operand, the command affects the bottom-most display on the console where you enter the command.

*Example:* To display the next frame of a status display, in display area B of console 12, enter the following:

```
CONTROL D,F,L=12B
```

- **Framing with the Cursor**

The control line of a status display contains a framing indicator which appears as F on the screen. To frame with the cursor, position the cursor at the F and perform an ENTER action.

- **Framing with the Selector Pen**

The control line of a status display contains a framing indicator which appears as F on the screen. To frame with the selector pen, press the selector pen against the screen over the F.

## Erasing Status Displays

It is not necessary to erase a status display from a display area in order to present another status display in that area. If a new status display is routed to an area that already contains a status display, the new display replaces the old display. However, it is good practice to erase status displays to avoid using buffer space and screen space that might be required for other messages.

The method of deleting status displays depends on whether they are displayed in the message area of the screen (inline) or in a defined display area (out-of-line).

### Erasing an Out-of-line Status Display

To erase an out-of-line status display (one that is presented in a display area), position the cursor at the E indicator in the control line and perform an enter action, press the selector pen against the screen over the same indicator, or use the CONTROL E,D command.

*Example:* To erase the display in display area A of console number 12, enter the following:

```
CONTROL E,D,L=12A
```



## Erasing an Inline Status Display

To erase an inline status display (one that is not presented in a display area), use the message deletion methods used for other inline messages (CONTROL E,SEG; CONTROL E,nn,nn; the selector pen; or the cursor). These methods are described under “Deleting Messages from the Console Screen” earlier in this chapter.

An inline status display that has not been completely displayed (that is, a display that is in progress) can be halted by performing the selector pen or cursor detect on the C indicator in the control line or by using the CONTROL C,D,id command. *id* is the three-digit identification number located in the control line of the display to be cancelled.

*Example:* To halt the status display with identification number 121, which is in progress in the general message area of console number 10, enter:

```
CONTROL C,D,121,L=10Z
```

*Note:* Because of the speed of display consoles, this command has limited usefulness in controlling inline displays on the screen. It is used primarily to halt displays in progress on printer consoles, such as the 3284.

## Establishing Dynamic Status Displays

A dynamic status display is one that you request with the TRACK command. You request it once, and the system then updates and redisplay it whenever a preset time interval elapses. Dynamic status displays can appear only in display areas. Because you cannot define display areas on message stream consoles, you cannot request dynamic status displays for these consoles. Also, once the dynamic display appears in an area, it has exclusive use of the area until it is terminated; it cannot be overlaid by another status display.

The TRACK command presents dynamic status displays of the following 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

## Holding, Framing, and Updating a Dynamic Status Display

The CONTROL D command can hold, frame, and update dynamic status displays. To understand how to use this command, first request a dynamic display of active batch jobs, started tasks, and logged-on time-sharing users in display area A on console 12:

```
TRACK A,L=12A
```

The following examples show you how to use the CONTROL D command to hold, update, and frame this display.

*Example 1:* To suspend updating of the dynamic display in area A of console 12, enter:

```
CONTROL D,H,L=12A
```

You can also put updating of a dynamic display on hold by positioning the cursor at the H indicator in the first line of the display and performing an ENTER action, or by pressing the selector pen against the screen over the H indicator.

*Example 2:* To resume updating of the display in display area A of console 12 (assuming that you have previously put the display on hold), enter:

```
CONTROL D,U,L=12A
```

You can also resume updating by positioning the cursor under the U indicator in the control line of the display and performing an ENTER action, or by pressing the selector pen against the screen over the same indicator. The U indicator appears in the control line only when the display is on hold.

You can use the CONTROL D command to update a display immediately (without having to wait for the time interval to elapse).

*Example 3:* To frame a dynamic display in display area A of console 12, first put the display on hold by entering:

```
CONTROL D,H,L=12A
```

Then, display the next frame by entering:

```
CONTROL D,F,L=12A
```

When you want to resume updating of the display, enter:

```
CONTROL D,U,L=12A
```

Updating resumes immediately and the first frame of the updated display appears on the screen. If the time interval is changed by a CONTROL T command, the next update occurs after the new interval. See “Changing the Time Interval for Updating a Dynamic Status Display” later in this section.

You can also perform the same hold, frame, and update function with the cursor as follows:

1. Put the display on hold by positioning the cursor at the H indicator in the control line.
2. Perform an ENTER action.
3. Frame forward by positioning the cursor at the F indicator (the F indicator appears only when the display is on hold).
4. Perform an ENTER action once for each frame that you want to see.
5. To resume updating of the display by positioning the cursor at the U indicator in the control line (the U indicator appears only when the display is on hold).
6. Perform an ENTER action.

You can also perform the same hold, frame, and update function with the selector pen as follows:

1. Put the display on hold by pressing the pen against the screen over the H indicator.
2. Display the frames by pressing the pen against the screen over the F indicator *once per frame*.
3. Resume updating by pressing the pen against the screen over the U indicator.

## Changing the Time Interval for Updating a Dynamic Status Display

Each console where a dynamic status display is in progress has a time interval value that determines how often the system updates the TRACK command displays. The time interval value depends on: how you code:

- The UTME parameter on the CONSOLE statement in SYS1.PARMLIB
- The CONTROL T,UTME command

If you do not define a time interval value, the system updates the display every 30 seconds.

To check the current time interval for updating dynamic displays on your console, issue the CONTROL T,REF command. In response, the system displays the current time interval in the entry area in CONTROL command form. You can then change the time interval in the entry area by using the procedures described in “Changing Information in the Entry Area” on page 2-43.

**You control the time interval for updating a dynamic display by:**

- **Using the CONSOLxx Member of SYS1.PARMLIB**

The UTME parameter on the CONSOLE statement in SYS1.PARMLIB controls the time interval for updating dynamic displays.

*Example:* To set the time interval for updating dynamic displays on your console to be every two minutes (120 seconds), code the following parameter on the CONSOLE statement in the CONSOLxx member for your console:

```
UTME(120)
```

- **Using the CONTROL Command**

Use the CONTROL T,UTME command to set the time interval for updating the dynamic status displays. The effect of this command lasts only for the duration of the IPL.

*Example:* To set the interval for updating dynamic displays on your console to be every two minutes, enter:

```
CONTROL T,UTME=120
```

## Stopping a Dynamic Status Display

You can stop and erase a dynamic display initiated by a TRACK JOBS, TRACK TS, or TRACK A command with the STOPTR command. See Chapter 4 for a complete description of the STOPTR command.

You can also stop and erase a dynamic status display by positioning the cursor at the PT indicator in the control line of the display and performing an ENTER action (the PT indicator does not appear when the display is on hold). You can also stop and erase the display by performing a selector pen detect on the same PT indicator.

## Displaying Jobname, Data Set Status, and TSO Information

You can request that the system notify you when certain events occur:

| When this Event Occurs:          | This Information Appears:               | Option:  |
|----------------------------------|---|----------|
| Jobs start and start             | Name of job                             | JOBNAMES |
| Datasets are freed               | Dataset names and volume serial numbers | STATUS   |
| TSO session beginning and ending | User identifier                         | SESS     |

You can add a time stamp to the user identifier or job name information. This command affects only the issuing console or another console that you specify.

You request that the system display jobname, data set status, and TSO information by:

- **Using the CONSOLxx Member of SYS1.PARMLIB:**

The MONITOR parameter on the CONSOLE statement in the CONSOLxx member requests that the system send a message to a console when certain events occur.

*Example:* To request that the system display the user identifier for each TSO terminal whenever a user initiates or ends a session on the terminal, code the following parameter on the CONSOLE statement:

```
MONITOR (SESS)
```

- **Using the MONITOR Command:**

Use the MONITOR command to request that the system send a message to a console when certain events occur.

*Example:* To request that the system display on a console the user identifier for each TSO terminal whenever a user initiates or ends a session on that console, enter:

```
MONITOR SESS
```

You can use the location operand to direct the MONITOR messages to another console.

The effect of the MONITOR command lasts for the duration of the IPL or until a STOPMN command is issued for the console.

## Adding Information to Mount and Demount Messages

You can request that the system add certain information to all mount and demount messages on consoles.

**DSNAME** In mount messages, the name of the first nontemporary data set allocated on the volume to which the messages refer

**SPACE** In demount messages, the available space on the direct access volume

You request that the system add information to mount and demount messages by:

- **Using the CONSOLxx Member of SYS1.PARMLIB:**

The MONITOR parameter on the INIT statement in the CONSOLxx member controls whether the system adds information to mount and demount messages.

*Example:* To display, in mount messages, the available space on the direct access volume, code the following parameter on the INIT statement in the CONSOLxx member:

```
MONITOR (SPACE)
```

- **Using the MONITOR Command:**

Use the MONITOR command to add information to mount and demount messages.

*Example:* To display, in mount messages, the available space on the direct access volume, enter:

```
MONITOR SPACE
```

If you enter the location operand on the SPACE and DSNAME operands, the system ignores the routing request.

The effect of the MONITOR command lasts for the duration of the IPL or until a STOPMN for DSNAME or SPACE is issued. If the console on which you issued the command goes down, the effect of the command continues at other consoles in the system.

## Managing Messages with a Console Cluster

A console cluster is a group of several consoles located together that you can use in place of a single console to divide up the functions and message traffic of the single console. In a system with display consoles, using a console cluster can help you manage your system from a console location more efficiently because using more than one console has the following advantages:

- You have more console screen space. With more console screen space, you can both display more messages at once and define more display areas. Displaying more messages helps you keep the message rate to the console location under control. Defining more display areas might give you the space you need to hold:
  - A TRACK command display. With the status display a TRACK command gives you, you do not have to issue commands frequently to check on system status.
  - Responses to selected commands. You can pick out the responses to commands more easily if they are not mixed in with other message traffic.
- You can sort information among the consoles. You can, for example, direct messages of the same message type to the same place on the same console.

You can set up a console cluster for any console location. Because of the high volume of message traffic to the master console in some installations, however, the usual place for a console cluster is in the master console area.

You can design a master console cluster to suit the special needs of your installation. A typical master console cluster might consist of four consoles, placed next to each other with one console as the master console. You could set up the consoles to receive these messages:

- The master console in full-capability mode to receive the action messages that you **must** see
- A console in message stream mode to receive the information messages that you **must** see
- A console in message stream mode to receive ordinary system message traffic (This console gives you basic information on how the system is running.)
- A console in status display mode to dynamically display the active jobs in the system and provide display areas for system status displays.

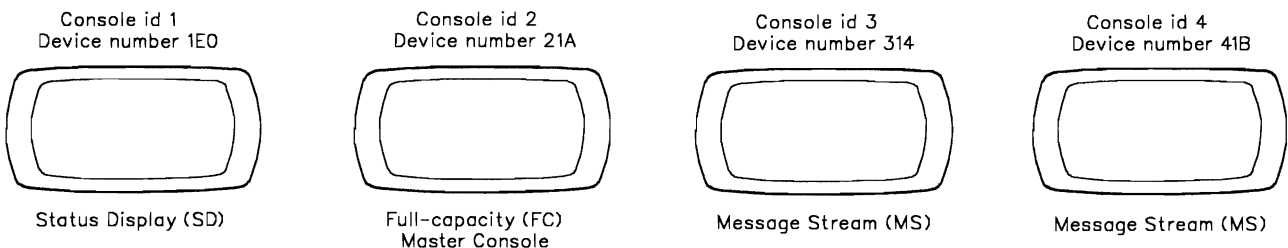
Include other consoles in the cluster if you want to divide the master console message traffic even more.

## Setting Up and Using a Master Console Cluster

If you decide to set up a master console cluster, you might want to follow the procedures outlined in the following detailed example. You need not follow the example exactly as it is given. Depending on your needs and the characteristics of your consoles, choose your own values for area sizes and numbers, PFK definitions, commands, and so forth.

This example describes how to set up and use a master console cluster that consists of four consoles. The example assumes that:

- Each of the devices is a 3278 model 4. The 3278 model 4 has a screen that holds 43 lines. It can, therefore, display many more messages than a 3277 model 2. Also, a 3278 has 24 PFKs (program function keys), while the 3277 has only 12. A device you use in a cluster does not have to be a 3278. The device you use to receive status displays, however, is far more useful if it has a large screen. (For the physical characteristics of all MVS/XA display consoles, see “Specific Characteristics of Display Consoles” in Chapter 2.)
- Two consoles require keyboards: the master console and the console you specify as its alternate.
- All devices in the cluster come online during the IPL process. They come online with the characteristics that you define in CONSOLxx member. The PFKs on the consoles are defined at IPL with the definitions you establish in the PFK table you assign to the console.
- The ids and device numbers of the consoles used in this example are as follows (the mode each console is in when you finish setting up the cluster is also shown):



You should put the four devices in the cluster on different control units, if possible, to make recovery easier if a control unit fails.

Many of the statements you define in the CONSOLxx member serve to divide the message traffic among the consoles and set up the message roll rate for each screen. When you complete the procedure described on the following pages:

- Console 1, the status display console, will receive the output from the DISPLAY and TRACK commands.
- Console 2, the full-capability console, will receive the messages that the master console operator **must** act on. The console will be in roll-deletable mode. (In roll-deletable mode, outstanding action messages are not automatically removed from the screen.)

- Console 3, a message stream console, will receive the messages that operators at other consoles must act on. The console will be in roll-deletable mode.
- Console 4, another message stream console, will receive all the information messages in the system. The console will be in roll mode. (In roll mode, a specified number of flagged messages roll off the screen after a specified time interval.)

The procedure for setting up a console cluster involves coding the statements in CONSOLxx and placing PFK definitions in a PFK table in the PFKTABxx member of SYS1.PARMLIB. In this description, each CONSOLxx statement is followed by the MVS command that matches the statement. The commands are shown so that you can see how to change console characteristics after the IPL. (These changes will not be in place when you bring up the console cluster again.)

Setting up a master console cluster requires several steps. This example describes:

- How to set up an alternate console chain
- How to define routing codes for the consoles
- How to define the operating modes and message levels for the consoles
- How to set up display areas
- How to set up a TRACK display
- How to set message roll rates and message deletion specifications for the consoles
- How to direct command responses to specific consoles
- How to set up a periodic display of outstanding requests for JES2 and JES3
- How to define program function keys (PFKs)

## Setting Up an Alternate Console Chain

If a console fails, MVS/XA searches the console's chain of alternates to find an active console to take over the failing console's functions. In this example, the master console's chain of alternates should include console 3 so MVS/XA can find an active console in the same area if the master console fails. Code the following statement in CONSOLxx with the device numbers appropriate to your installation. For the consoles in the example, the statements are:

```

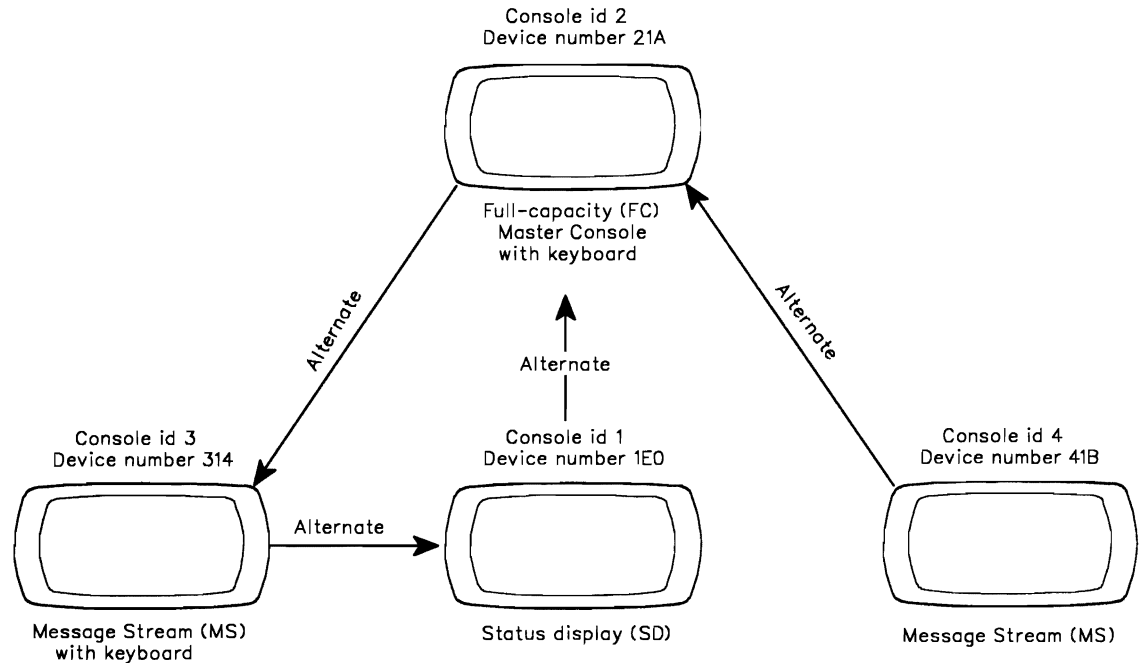
CONSOLE DEVNUM(21A) ALTERNATE(314) AUTH(MASTER)
CONSOLE DEVNUM(314) ALTERNATE(1E0)
CONSOLE DEVNUM(1E0) ALTERNATE(21A)
CONSOLE DEVNUM(41B) ALTERNATE(21A)

```

If the master console fails, the system assigns console 3 as the master console, changes its operating mode to full-capability, and activates its keyboard.



The chain of alternate consoles in this master console cluster looks like this:



- To change the alternate consoles after the IPL, use the following MVS/XA commands:

```

VARY 21A,CONSOLE,ALTCONS=314
VARY 314,CONSOLE,ALTCONS=1E0
VARY 1E0,CONSOLE,ALTCONS=21A
VARY 41B,CONSOLE,ALTCONS=21A
  
```

The effect of these commands lasts only for the duration of the IPL.

## Defining Routing Codes for the Consoles

Use routing codes to set up the master console (console 2) so that it receives only messages for which the master console operator is responsible. Direct other messages to console 3 and all routing codes to console 4. In the next section, you will see how the LEVEL parameter in CONSOLxx further limits messages to these consoles. Code the following statements to set up the routing codes for the master consoles (console 2) and the message stream console (consoles 3 and 4), substituting device numbers appropriate to your installation:

```

CONSOLE DEVNUM(21A) ROUTCODE(1,2,9,10)
CONSOLE DEVNUM(314) ROUTCODE(3-8,12-15)
CONSOLE DEVNUM(41B) ROUTCODE(ALL)
  
```

*Note:* You do not have to direct tape, DASD, unit record, and teleprocessing messages (routing codes 3, 4, 5, 6, 7, and 8) to console 3 if a console other than the master is receiving these routing codes and the master console operator does not have to take any action for such messages.

As a result of these statements, consoles 2, 3, and 4 display the complete range of master console messages. You do not need to define routing codes for console 1 because you are going to put console 1 in status display mode. In status display mode, a console can display only system status display messages that you direct specifically to the console.

- To make the same routing code assignments after the IPL, use the following MVS/XA commands:

```
VARY 21A,CONSOLE,ROUT=(1,2,9,10)
VARY 314,CONSOLE,ROUT=(3-8,12-15)
VARY 41B,CONSOLE,ROUT=ALL
```

The effect of the commands lasts only for the duration of the IPL.

## Defining the Operating Modes and the Message Levels for the Consoles

Code the following statement in CONSOLxx to change the operating mode of console 1 to output-only for system status displays:

```
CONSOLE DEVNUM(1E0) USE(SD)
```

Use statements in CONSOLxx to define the operating modes and the message levels for consoles 2, 3, and 4. To further reduce the messages that appear at the master console, which is already in full-capability mode, eliminate non-action messages from console 2. Code the following statement in CONSOLxx:

```
CONSOLE DEVNUM(21A) LEVEL(R,I,CE,E,NB)
```

As a result of this statement console 2 receives all action messages with routing codes 1, 2, 9, and 10; it receives no informational or broadcast messages.

Define consoles 3 and 4 as message stream consoles. Set up message levels for the two consoles so that console 3 receives all the action messages that the master console does not receive, and console 4 receives all the informational messages for the system. Code the following statement in CONSOLxx:

```
CONSOLE DEVNUM(314) USE(MS) LEVEL(R,I,CE,E,NB)
CONSOLE DEVNUM(41B) USE(MS) LEVEL(IN)
```

As a result of these statements, console 3 receives all messages with routing codes 3, 4, 5, 6, 7, 8, 12, 13, 14, and 15 that require operator response; console 4 receives all informational messages that the system issues.

- To make the same operating mode and routing code assignments after the IPL, use the following MVS/XA commands:

```
CONTROL V,USE=SD,L=1
CONTROL V,LEVEL=(R,I,CE,E,NB),L=2
CONTROL V,USE=MS,LEVEL=(R,I,CE,E,NB),L=3
CONTROL V,USE=MS,LEVEL=IN,L=4
```

The effect of these commands lasts only for the duration of the IPL.

## Setting Up Display Areas

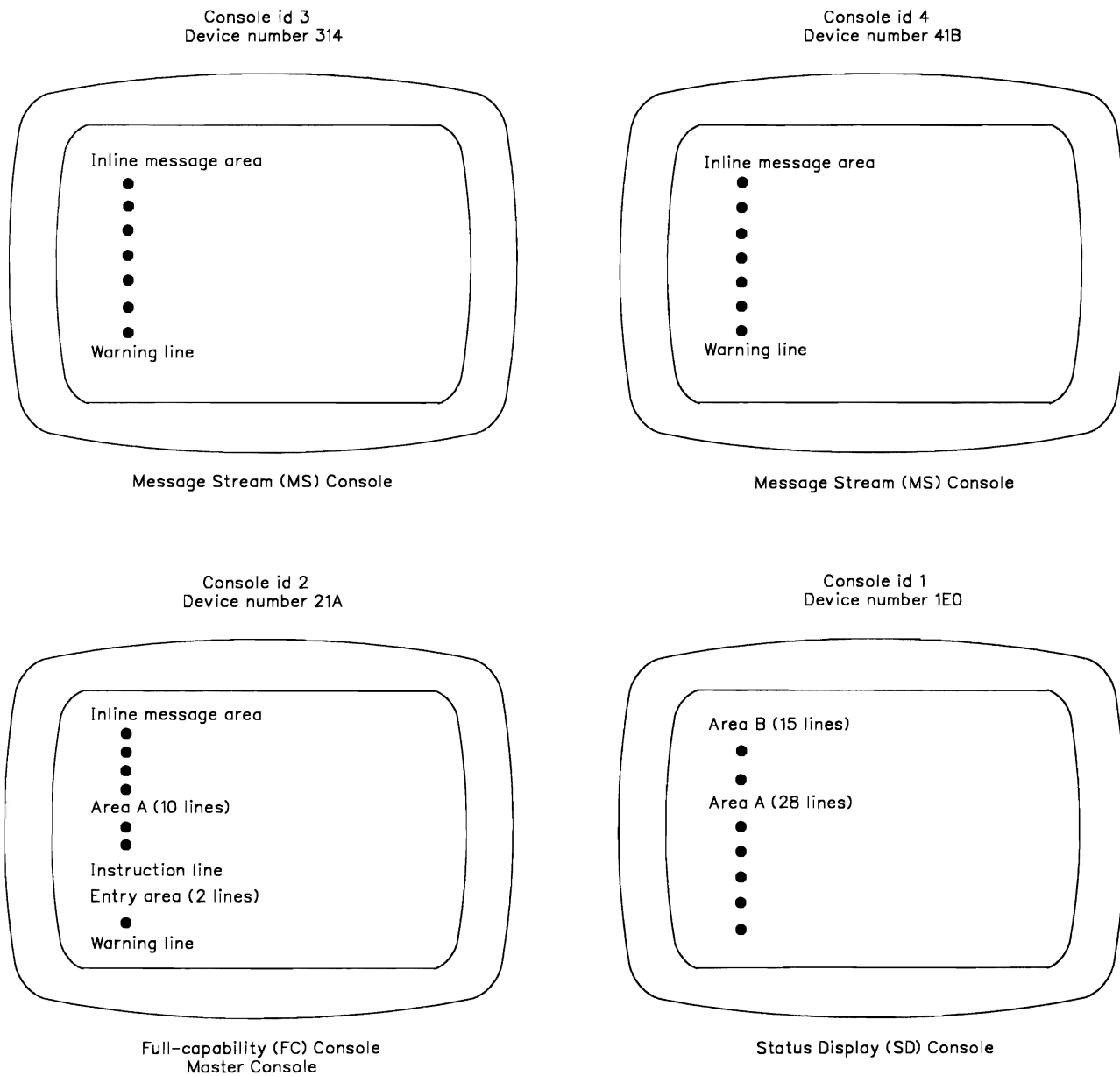
The next step is to define two display areas on console 1. The bottommost area (area A) should have 28 lines. When you put a TRACK command display in this area, you can monitor the activity of up to 50 jobs. The second area (area B) should have 15 lines. You can direct system status displays to this area. To define these areas, code the following statement in CONSOLxx:

```
CONSOLE DEVNUM(1E0) AREA(28,15)
```

Console 2 should also have a display area because display area B on console 1 might not be able to receive all system status displays. An area of ten lines should be enough. To establish this area, code the following statement in CONSOLxx:

```
CONSOLE DEVNUM(21A) AREA(10)
```

The display areas you have established on the consoles are:



**Figure 3-14. Display Areas on Consoles in the Console Cluster**

- To make the same display areas for consoles 1 and 2 after the IPL, use the following MVS/XA commands:

```
CONTROL A,28,15,L=1
CONTROL A,10,L=2
```

The effect of these commands lasts only for the duration of the IPL.

## Setting Up a TRACK Display

Output from the TRACK command is basically the same as output from the DISPLAY A command. The main difference is that the system periodically updates the display produced by the TRACK command.

Output from the TRACK command starts with general system statistics: the number of jobs, mounts and starts, time-sharing users, system address spaces, initiators, actual TSO/VTAM users, and allowed TSO/VTAM users. This information appears on the first three lines of the display area. Each of the other lines lists information for two specific jobs in the system. The information displayed for each job includes: the job name, the step name, the procedure step name, and the status of the job's address space.

If fewer than 50 jobs are executing and time-sharing is active, the status of time-sharing users follows the status of jobs on the first frame. If there are no time-sharing users, the bottom lines of the display area are blank. If more than 50 jobs are active and you want to see the status of the jobs or time-sharing users that are not displayed, you might want to use a PFK to frame through the display. The statement in CONSOLxx that defines this PFK is described later in this example under "Defining the Program Function Keys (PFKs) for Console 2."

When you issue a TRACK command, the system automatically updates the TRACK display every 30 seconds. Because you might need more frequent status updates to keep track of a specific job's execution, you should cause the system to update the TRACK display every ten seconds. Once the TRACK display is updated this frequently, you might not want its output to appear in the hardcopy log. To keep the TRACK display out of the hardcopy log, code the following statement in CONSOLxx:

```
HARDCOPY CMDLEVEL(STCMDS)
```

To update the TRACK display every ten seconds, code the following statement in CONSOLxx:

```
CONSOLE DEVNUM(1E0) UTME(10)
```

- To send all messages except those from the TRACK command to the hardcopy log and update the TRACK display every 10 seconds, use the following MVS/XA command:

```
CONTROL T,UTME=10,L=1  
VARY ,HARDCPY,STCMDS
```

The effect of these commands lasts only for the duration of the IPL.

The mode of the TRACK display can be either update mode or hold mode. In update mode, the system updates the display, for this example, every ten seconds. Do not try to frame forward through the TRACK display when it is in update mode. When the system updates the display, it returns the display to its first frame, and you might have to start framing all over again to find the information you are looking for. If you put the TRACK display in hold mode with a CONTROL D,H command, you can frame through the display with the CONTROL D,F command. The system does not update a TRACK display in hold mode until you put the display back in update mode with a CONTROL D,U command.

Place the TRACK command in COMMNDxx:

```
TRACK A,L,L=1A
```

When you bring up the console, the TRACK display appears in area A of console 1. In “Defining PFKs for Console 2” later in this section, PFKs 22, 23, and 24 are defined to allow you to use the TRACK display efficiently.

## Setting Message Roll Rates and Message Deletion Specifications for the Consoles

The message roll rate appropriate for a console depends on the message traffic to that console. To establish a starting message roll rate for consoles 2, 3, and 4, code the following statements in CONSOLxx:

```
CONSOLE DEVNUM(21A) DEL(RD) SEG(39) CON(N) RNUM(10) RTME(2)
CONSOLE DEVNUM(314) DEL(RD) SEG(39) RNUM(10) RTME(2)
CONSOLE DEVNUM(41B) DEL(R) SEG(39) RNUM(10) RTME(2)
```

These statements put consoles 2 and 3 in roll-deletable mode and console 4 in roll mode. (See “Defining Automatic Message Deletion” for a full description of the roll and roll-deletable modes of message deletion.) Adjust the RNUM and RTME specifications until the roll rate is appropriate for the message traffic on each console. Use the adjustment procedure described under “Finding Appropriate Message Deletion Specifications.” Once you determine the appropriate values for RNUM and RTME, code the values in the RNUM and RTME parameters in CONSOLxx.

- To establish the same message roll rates for consoles 1 and 2 after the IPL, use the following MVS/XA commands:

```
CONTROL S,DEL=RD,SEG=39,CON=N,RNUM=10,RTME=2,L=2
CONTROL S,DEL=RD,SEG=39,RNUM=10,RTME=2,L=3
CONTROL S,DEL=R,SEG=39,RNUM=10,RTME=2,L=4
```

The effect of the commands lasts only for the duration of the IPL.

## Directing Command Responses to Specific Consoles

Use the MSGRT statement in CONSOLxx to direct the responses to certain MVS/XA commands to specific consoles without having to specify the location operand (L=) on each command. For this example, direct the responses to all DISPLAY commands you enter on console 2 to display area B of console 1. You should also cause any TRACK commands you issue on console 2 to appear in display area A of console 1. Code the following statement in CONSOLxx:

```
CONSOLE DEVNUM(1E0) MSGRT ('D=(A,C,CONSOLES,D,DUMP,GRS,M,MPF
PFK,R,S,SMF,U,3850),L=1B'), 'TR=A,L=1A')
```

As a result of this statement:

- Any of the following DISPLAY commands you enter on console 2 appear in display area B of console 1:

|                  |             |              |
|------------------|-------------|--------------|
| DISPLAY A        | DISPLAY GRS | DISPLAY SLIP |
| DISPLAY C,K      | DISPLAY M   | DISPLAY SMF  |
| DISPLAY CONSOLES | DISPLAY MPF | DISPLAY U    |
| DISPLAY DMN      | DISPLAY PFK | DISPLAY 3850 |
| DISPLAY DUMP     | DISPLAY R   |              |

- The responses to all TRACK commands you enter on console 2 appear in display area A on console 1.

- To route the same DISPLAY and TRACK commands to message areas on console 1 after the IPL, use the following MVS/XA commands:

```
MSGRT D=(A,C,CONSOLES,D,DUMP,GRS,M,MPF,PFK,R,S,SMF,U,3850),L=1B
MSGRT TR=A,L=1A
```

The effect of the commands lasts only for the duration of the IPL.

If your system includes JES2, when you bring up the console, you can use the \$TM JES2 command to direct the responses to certain JES2 \$D commands to specific consoles. For this example, direct the responses to the JES2 commands \$DA, \$DF, \$DI, \$DJ, \$DN, \$DO, \$DQ, and \$DU to display area B of console 1. Issue the following command to make this change:

```
$TM1B,AFIJNOQU
```

*Note:* Put this command in the JES2 initialization data set so that it is issued automatically once JES2 is initialized.

If your system includes JES3 and JES3 is retaining action messages, you can periodically enter the JES3 \*I,R command or the MVS DISPLAY R command to display outstanding JES3 requests.

## Setting Up a Periodic Display of Outstanding Requests

If your system includes JES2, you can have the system periodically display outstanding requests so that you always know how many there are.

You can set up such a periodic display through the JES2 automatic command facility, telling JES2 to issue a command at a defined interval. (The minimum time interval you can specify is 30 seconds.) You must use the \$TA JES2 command to define both the command you want issued and the number of seconds in the interval between commands. To cause JES2 to issue a DISPLAY R command every 60 seconds and to direct the command output to display area B of console 1, issue the following command:

```
$TA,I=60,'$VS','D R,L,L=1B'''
```

*Note:* Put this command in the JES2 initialization data set so that it is issued automatically once JES2 is initialized.

You use the \$ZA JES2 command to temporarily stop JES2 from issuing the defined commands. You use the \$SA command to cause JES2 to resume issuing the defined commands. Use the \$CA command to cancel both the defined commands and time interval:

```
$CAxxxx
```

where xxxx is the ID of the periodic display.

If your system includes JES3, use JES3 commands to direct messages to specified consoles.

## Summary of Contents of CONSOLxx for the Cluster

The statements you placed in CONSOLxx to initialize the cluster are:

```
CONSOLE DEVNUM(1EO) ALTERNATE(21A)
      USE(SD)
      AREA(28,15)
      MSGRT ('D=(A,C,CONSOLES,D,DUMP,GRS,M,
            PFK,R,S,SMF,U,3850),L=1B',TR=A,L=1A')

CONSOLE DEVNUM(21A) ALTERNATE(314) AUTH(MASTER
      ROUTCODE(1,2,9,10)
      LEVEL(R,I,CE,E,NB)
      AREA(10)
      DEL(RD) SEG(39) CON=(N) RNUM(10) RTME(2)
      MSGRT ('D=(A,C,CONSOLES,D,DUMP,GRS,M,MPF,PFK,R,S,
            SMF,U,3850),L=1B');('TR=A,L=1A')
      PFKTAB(MASTCMDS)

CONSOLE DEVNUM(314) ALTERNATE(1EO)
      ROUTCODE(3-8,12-15)
      USE(MS) LEVEL(R,I,CE,E,NB)
      DEL(RD) SEG(39) RNUM(10) RTME(2)

CONSOLE DEVNUM(41B) ALTERNATE(21A)
      ROUTCODE(ALL)
      USE(MS) LEVEL(IN)
      DEL(R) SEG(39) RNUM(10) RTME(2)

HARDCOPY CMDLEVEL(STCMDS)

INIT PFK(02)
```

### Notes:

1. *Substitute the device numbers that are appropriate to your installation.*
2. *Adjust SEG, RNUM, RTME, and other values, as appropriate to the devices in your console cluster.*
3. *If you have JES2 at your installation, place the following commands in the initialization data set:*
  - *\$TM1B,AFIJNOQU*
  - *\$TA,I=60,\$VS,"D R,L"*
4. *If JES3 uses an MCS console, define the console both in the JES3 initialization stream and in the CONSOLxx member. To initialize the JES3 consoles that use MCS, see JES3 Initialization and Tuning.*



## Defining PFKs for Console 2

You have to redefine some of the PFKs 1 through 9 that the system assigns at IPL and define additional PFKs for console 2 because:

- The nine PFKs are not enough to set up and use the console cluster effectively.
- PFKs 1 through 9 do not put the commands you need to operate the console cluster in the most convenient places.

You need to define PFKs on the master console (console 2 in the example) for the common operator command functions and the commands to control the console cluster because the master console is the only full-capability console in the cluster.

Place all your definitions for PFKs in a PFK table that you create with the name `MASTCMDS`. All the definitions in this section follow the first statement in the table:

```
PFKTAB TABLE(MASTCMDS)
```

The commands you define in this table go into effect at IPL, providing you activate the table. The section “Activating the PFK Table” on page 3-69 describes how you activate `MASTCMDS` by defining the `PFKTABxx` member in `SYS1.PARMLIB` member that contains it. If you want to dynamically redefine a PFK, use the `CONTROL N,PFK` command. See “Defining PFKs Using the CONTROL Command” on page 3-31 for help in using this command.

So that you don’t have to press the ALT key on the master console every time you enter a commonly-used PFK function, you should define PFKs on the PFK pad (rather than on the top row of the keyboard) to enter commands you need to use often. Define PFKs 13, 14, 17, and 18 to enter the functions defined for PFKs 1, 2, 5, and 6 at IPL. Add the following entries to `MASTCMDS` to control erasing and displaying of messages on console 2:

```
PFK(13) CMD('K E,1')
PFK(14) CMD('K E')
PFK(17) CMD('K S,DEL=N')
PFK(18) CMD('K S,DEL=RD')
```

For controlling the cluster, define PFKs 15 and 16 to erase and frame system status displays on console 1, code the following entry in `MASTCMDS`:

```
PFK(15) CMD('K E,D,L=1B')
PFK(16) CMD('K D,F,L=1B')
```

As a result of these definitions, PFK 15 erases a status display from display area B of console 1 and PFK 16 frames a status display in display area B of console 1.

To establish the message routing instructions for JES2 messages, add the following entry to MASTCMDS:

```
PFK(3) CMD('$TM1B,AFIJNOQU'; '$TA,I=60, '$VS, '' 'D R,L,L=1B'' '' ''')
```

As a result of this definition, pressing PFK 3

- Directs the output of any of the following JES2 \$D commands you enter on console 2 to display area B of console 1:

|      |      |      |
|------|------|------|
| \$DA | \$DJ | \$DQ |
| \$DF | \$DN | \$DU |
| \$DI | \$DO |      |

- Makes the JES2 automatic command facility issue a DISPLAY R command every 60 seconds and direct the command response to display area B of console 1.

If your system includes consoles managed by JES3, you would use JES3 procedures for defining your PFKs.

You should define PFKs to remove action messages quickly from the screens of the master and message stream consoles (consoles 2 and 3) because both consoles will be in roll-deletable mode when the console cluster is fully active. In roll-deletable mode, outstanding action messages are not automatically removed from the screen. Therefore, if you do not remove the action messages, the screen eventually fills with these messages and messages that are waiting to appear start to use up the message buffer space. Define PFK 11 by adding the following entry to MASTCMDS:

```
PFK(11) CMD('K V,USE=SD,L=3'; 'K V USE=MS,L=3')
```

As a result of this command, pressing PFK 11:

- Causes the message stream console (console 3) to enter status display mode
- Causes console 3 to return to message stream mode

When the system puts console 3 in status display mode, it erases all messages displayed on its screen and removes all backed-up messages from its message queue.

Define PFK 12 by adding the following entry to MASTCMDS:

```
PFK(12) CMD('K S,DEL=R')
```

As a result of this command, pressing PFK 12 causes the master console (console 2) to roll all messages.

When PFK 12 makes console 2 roll all its messages, it reduces the number of backed-up messages, in effect, by displaying them all.

*Notes:*

- 1. If the action message retention facility is active, you can issue a DISPLAY R command to display again any action messages that are retained (that is, the messages that roll off or are erased from a screen).*
- 2. If action messages fill up a console screen frequently, first make sure that you are responding to the messages. If you do not respond to them, the system cannot remove them automatically from the screen. If you are responding to the messages as you should, check the configuration of the console cluster. You might have to:*
  - Add another console to the cluster so you can split up the message traffic even more*
  - Include more large-screen devices (such as the 3290) in the cluster*
  - Keep the master console in the roll mode of message deletion (instead of the roll-deletable mode) so that all messages roll off the screen*

You should define a PFK to display all the outstanding requests at once so you can always keep track of or respond to them. Define PFK 21 by adding the following entry to MASTCMDS:

```
PFK(21) CMD('K V,USE=MS,L=1;K V,USE=SD,L=1;K A,15,18,10,
            L=1;$DU,L=1A;D R,L,L=1B;$DO,L=1C')
```

As a result of this command, pressing PFK 21:

- Changes console 1 to message stream mode
- Puts console 1 back in status display mode
- Defines new display areas A (15 lines), B (18 lines), and C (10 lines) for console 1
- Displays JES2 unit record device status in display area A of console 1
- Displays outstanding requests in display area B of console 1
- Displays JES2 device-related operator actions in display area C of console 1

When console 1 enters message stream mode, the system stops the TRACK display in area A and erases any display in area B, thus clearing the screen for the display of outstanding requests.

You should define a PFK to restore the original TRACK command display when you no longer need the display of outstanding requests. Define PFK 22 by issuing:

```
PFK(22) CMD('K V,USE=MS,L=1;K V,USE=SD,L=1;K A,28,15,L=1;
            TR A,L;K T,UTME=10,L=1')
```

As a result of this command, pressing PFK 22:

- Erases any system status displays on console 1's screen by changing the mode of console 1 from status display to message stream and back to status display.
- Defines display areas A and B on console 1.
- Starts a TRACK display on console 1.
- Sets the time interval for updating the TRACK display to 10 seconds.

You should define a PFK to let you frame through the TRACK display on console 1. Define PFK 23 by adding the following entry to MASTCMDS:

```
PFK(23) CMD('K D,H,L=1A;K D,F,L=1A')
```

As a result of this command, pressing PFK 23:

- Puts the TRACK command display area in hold mode. (For information on the modes of a TRACK display, see "Setting Up a TRACK Display" earlier in these procedures.)
- Causes the next frame of the TRACK display to appear.

If you want to frame further through the display, press PFK 23 again.

You should define another PFK to cause the system to continue updating the TRACK display after you are finished framing through it. Define PFK 24 by adding the following entry to MASTCMDS:

```
PFK(24) CMD('K D,U,L=1A')
```

As a result of this command, pressing PFK 24 puts the TRACK display back in update mode.

## Summary of the PFK Definitions for the Cluster

The PFK table named MASTCMDS now contains the definitions that have been defined in the previous section. If you issue `DISPLAY PFK, TABLE = MASTCMDS`, the definitions, including those that IBM supplies, display. In message IEE235I, the NO that appears in the column labelled CON, indicates that the commands are non-conversational. The display appears as follows:

```
PFK DEFINITIONS FOR CONSOLE 2    TABLE=MASTCMDS IN PFKTAB02

KEY# CON  -----DEFINITION-----
1      NO   K E,1           ERASE TOP LINE FROM SCREEN
2      NO   K E             ERASE ONE SEGMENT FROM SCREEN
3      NO   $TM1B,AFIJNOQU;$TA,I=60, '$VS, '' 'D R,L,L=1B''''''''
4      NO   K D,F          FRAME DISPLAY FORWARD IN AREA
5      NO   K S,DEL=N      HOLD IN-LINE OUTPUT
6      NO   K S,DEL=RD     RESUME IN-LINE OUTPUT
7      NO   D A,L         LIST ACTIVE JOBS AND TSO USERS
8      NO   D R,L         LIST OPERATOR REQUESTS
9      NO   K D,U         UPDATE DYNAMIC DISPLAY
10     NO   NOT DEFINED
11     NO   K V,USE=SD,L=3;K V,USE=MS,L=3
12     NO   K S,DEL=R
13     NO   K E,1
14     NO   K E
15     NO   K E,D,L=1B
16     NO   K D,F,L=1B
17     NO   K S,DEL=N
18     NO   K S,DEL=RD
19     NO   NOT DEFINED
20     NO   NOT DEFINED
21     NO   NOT DEFINED
22     NO   K V,USE=MS,L=1;K V,USE=SD,L=1;K A,28,15,L=1;
           TR A,L;K T,UTME=10,L=1
23     NO   K D,H,L=1A;K D,F,L=1A'
24     NO   K D,U,L=1A
```

Notes:

1. The PFKs that are noted *NOT DEFINED* are available for you to define according to your needs.
2. If you put the console into message stream or display status mode, you can no longer use the PFKs.

## Activating the PFK Table

The PFK table named MASTCMDS must reside in a PFKTABxx member in SYS1.PARMLIB. In this example, assume that the member is named PFKTAB02. The following statements in CONSOLxx activate MASTCMDS:

```
CONSOLE DEVNUM(21A) PFKTAB(MASTCMDS)
INIT PFK(02)
```

The PFK commands you defined in MASTCMDS go in effect for console 2 at the next IPL.

## Using the Master Console Cluster and Setting It Up Again

Now that you have set up all the consoles and defined all the PFKs that you need in the cluster, you can work normally with the console message traffic at the master console (console 2). Press PFK 21 once in a while to display both the JES2 unit record device status and the outstanding requests on console 1. When you are done with this display, press PFK 22 to restore the TRACK display on console 1.

When you IPL again, the definitions you established in the CONSOLxx and PFKTABxx members are in effect. If you have JES2 on your system, as soon as JES2 becomes active, press PFK 3 to set up the JES2 message routing for the cluster and a periodic display of outstanding requests.

At this point, you have fully set up the cluster as before. You can now work normally again from the master console, using PFKs 21 and 22 as described earlier.

## The 3290 as a Console Cluster

You can set up a single 3290 as an entire console cluster, as described in Figure 3-15. This example shows a four-part 3290 screen initialized like the four 3278 consoles in Figure 3-14. The consoles on the 3290 are arranged so that the master console is the lower left-hand quadrant with the status display console beside it; the two message stream consoles are above them. To set up the cluster, follow the steps described earlier in this section. (Because each of the four consoles screens on the 3290 is shorter than the 3278-4 used in the console cluster example, adjust the values for SEG, RNUM, RTME, and area sizes.)

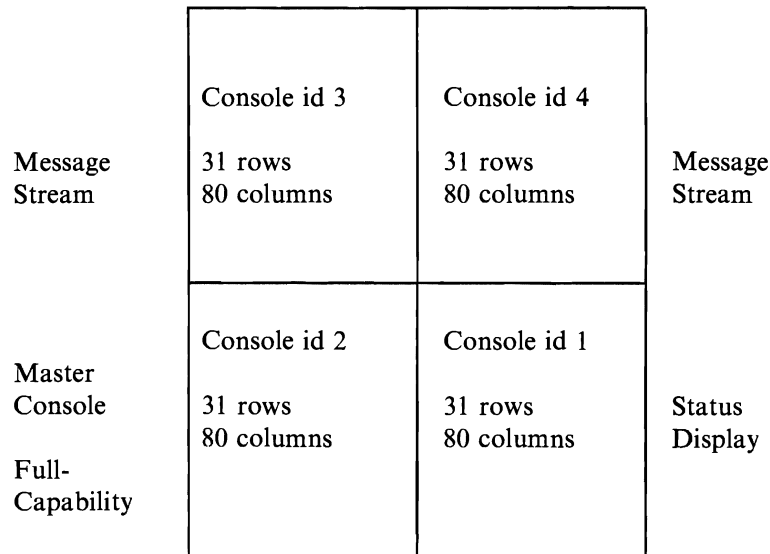


Figure 3-15. 3290 Information Panel as a Console Cluster

## Chapter 4. MVS/XA System Commands Reference

The tasks of starting, running, and stopping an MVS/Extended Architecture (MVS/XA) system involve:

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

Chapter 1 of this book describes how to operate the system itself. Chapters 2 and 3 describe how to define and operate the MCS consoles.

This chapter describes, in reference form, the functions, syntaxes, and parameters of all the MVS/XA system commands, which you can use to control both the system itself and MCS consoles.

Figure 4-1 sums up all the MVS/XA system commands and their functions. The figure also shows the operator command groups for each command and tells whether you can enter the command from the job stream or a time-sharing terminal as well as from a console.

Following the summary figure is a section on command format and syntax. The syntax rules are shown in Figure 4-2.

The rest of this chapter describes, in alphabetical order, each of the commands in more detail. For each command, the functions that the command performs are described followed by the command's complete syntax and all its parameters. Complex commands are broken down into sub-functions and sub-syntaxes. Parameters and keywords are described in the order in which they appear in the complete syntax.

| Command (Abbr)   | Function  | Acceptable From   | Command Group  |
|--|---|---|--|
| ASSIGN   | Assign primary host for MSS multi-host environment  |   | Mass storage command (Note 1)  |
| CANCEL (C)   | Cancel a MOUNT command<br>Cancel a job in execution<br>Cancel an external writer allocation<br>Cancel the writing of a SYSOUT data set by an external writer<br>Cancel a time-sharing terminal session  | Console or job stream<br><br><br><br>Console, job stream, or time-sharing terminal (Note 6) | SYS  |
| CHNGDUMP (CD)  | Override dump options specified in SYS1.PARMLIB, on the ABEND, CALLRTM, and SETRP macros, and in the SDUMP parameter list   | Console or job stream   | SYS  |
| CONFIG (CF)  | Place a processor online or offline<br>Place real storage elements online or offline<br>Place ranges of real storage online or offline<br>Place extended storage elements online or offline<br>Place channel paths online or offline<br>Place a Vector Facility online or offline   | Console   | MC   |
| CONTROL (K)  | Change display area specifications<br>Delete certain messages<br>Halt printing of a status display<br>Activate, deactivate, or display the status of the action message retention facility<br>Change or display the number of allowed message and reply buffers<br>Change or display message deletion or format specifications<br>Remove messages from screen<br>Change or display the status of WTO user exit IEAVMXIT<br>Define commands for PFks<br>Reroute message queue<br>Change or display time interval for updating a display<br>Change operating mode of console<br>Select the message levels for a console | Console   | INFO<br>INFO<br>INFO<br>MC<br><br>MC<br>INFO<br>INFO<br>MC<br>INFO<br>INFO or CONS or MC<br>INFO<br>INFO<br>INFO |
| <b>Notes:</b> <ol style="list-style-type: none"> <li>For information about MSS commands, see <i>Mass Storage System Extensions Operation</i>.</li> <li>CONS command group when message routing is specified.</li> <li>For information about VTAM commands, see <i>VTAM Operation</i>.</li> <li>For information on using SLIP, see <i>SPL: System Macros and Facilities and Diagnostic Techniques</i>.</li> <li>For information about TCAM commands, see <i>TCAM Operation</i>.</li> <li>For information about TSO commands, see <i>SPL: TSO</i>.</li> <li>I/O command group when specifying a non-console device; CONS when specifying a console.</li> </ol> |   |   |  |

Figure 4-1 (Part 1 of 6). System Command Summary



| Command (Abbr)  | Function   | Acceptable From  | Command Group |
|---|--|--|---------------|
| DEVSERV (DS)  | Display current status of devices and corresponding channel paths  | Console or job stream  | INFO          |
| DISPLAY (D)   | Display console configuration information<br>Display CONTROL command functions<br>Display configuration information<br>Display device allocation<br><br>Display current system status<br>Display system information requests<br>Display local and Greenwich mean time and date<br>Display domain descriptor table<br>Display status or contents of SYS1.DUMP data sets<br>Display dump options in effect<br>Display SMF options in effect or SMF data sets<br>Display page and swap data set information<br>Display SLIP trap information<br><br>Display commands defined for PFks<br>Display the messages MPF is processing and color, intensity, and highlighting display options in effect<br>Display state of the systems, a particular system's CTCs, and the contents of RNLs in the global resource serialization complex | Console or job stream<br><br>Console, job stream, or time-sharing terminal (Note 6)<br><br>Console or job stream<br><br>Console, job stream, or time-sharing terminal (Notes 4 and 6) (Note 6) | INFO (Note 2) |
| DUMP  | Request a dump of virtual storage to be stored in a SYS1.DUMP data set   | Console  | MC            |
| DUMPDS (DD)   | Change the system's list of SYS1.DUMP data sets<br>Clear full SYS1.DUMP data sets and make them available for dumps  | Console  | SYS           |
| FORCE   | Force termination of: <ul style="list-style-type: none"> <li>● A MOUNT command</li> <li>● A job in execution</li> <li>● An external writer allocation</li> <li>● The writing of a SYSOUT data set by an external writer</li> <li>● A time-sharing terminal session</li> <li>● A non-cancellable job, time-sharing user, or started task</li> </ul>   | Console  | MC            |
| <b>Notes:</b> <ol style="list-style-type: none"> <li>1. For information about MSS commands, see <i>Mass Storage System Extensions Operation</i>.</li> <li>2. CONS command group when message routing is specified.</li> <li>3. For information about VTAM commands, see <i>VTAM Operation</i>.</li> <li>4. For information on using SLIP, see <i>SPL: System Macros and Facilities and Diagnostic Techniques</i>.</li> <li>5. For information about TCAM commands, see <i>TCAM Operation</i>.</li> <li>6. For information about TSO commands, see <i>SPL: TSO</i>.</li> <li>7. I/O command group when specifying a non-console device; CONS when specifying a console.</li> </ol> |  |  |               |

Figure 4-1 (Part 2 of 6). System Command Summary

| Command (Abbr)   | Function   | Acceptable From  | Command Group         |
|--|--|--|-----------------------|
| HALT (Z)   | Halt the system while preserving statistics and data records. (Must first stop subsystem processing with a subsystem command)  | Console  | SYS                   |
| HOLD (H)   | Suspend transmission to a station  | Console or job stream                                  | TCAM command (Note 5) |
| LOG (L)  | Enter comments in the system log   | Console or job stream                                  | INFO                  |
| MODE   | Control recording of system recovery and degradation machine check interruptions on the SYS1.LOGREC data set<br><br>Control the monitoring of hard machine check interruptions   | Console  | SYS                   |
| MODIFY (F)   | Change characteristics of a job by modifying the job parameters<br><br>Specify criteria an external writer uses to select data sets for processing<br><br>Cause an external writer to pause for operator intervention<br><br>Start TSO/TCAM time-sharing once TCAM is active<br><br>Stop TSO/TCAM time-sharing<br><br>Modify TSO/VTAM time-sharing<br><br>Rebuild a new LNKLST directory | Console or job stream                                  | SYS                   |
| MONITOR (MN)   | Continuously display data set status<br><br>Continuously display job status<br><br>Monitor time-sharing terminal users logging on and off the system   | Console, job stream, or time-sharing terminal (Note 6) | INFO (Note 2)         |
| MOUNT (M)  | Mount volumes  | Console or job stream                                  | I/O                   |
| MSGRT (MR)   | Establish message routing instructions for certain options of DISPLAY, TRACK, STOPTR, MONITOR, STOPMN, CONFIG, and CONTROL commands<br><br>Stop message routing  | Console  | INFO                  |
| <b>Notes:</b> <ol style="list-style-type: none"> <li>For information about MSS commands, see <i>Mass Storage System Extensions Operation</i>.</li> <li>CONS command group when message routing is specified.</li> <li>For information about VTAM commands, see <i>VTAM Operation</i>.</li> <li>For information on using SLIP, see <i>SPL: System Macros and Facilities and Diagnostic Techniques</i>.</li> <li>For information about TCAM commands, see <i>TCAM Operation</i>.</li> <li>For information about TSO commands, see <i>SPL: TSO</i>.</li> <li>I/O command group when specifying a non-console device; CONS when specifying a console.</li> </ol> |  |  |                       |

**Figure 4-1 (Part 3 of 6). System Command Summary**

| Command (Abbr)  | Function   | Acceptable From  | Command Group                 |
|---|--|--|-------------------------------|
| PAGEADD (PA)  | Add page or swap data sets<br>Specify data sets as non-VIO page data sets  | Console  | SYS                           |
| PURGE   | Demount all 3330V volumes from specified host  | Console  | Mass storage command (Note 1) |
| QUIESCE   | Put system in MANUAL state without affecting step timing   | Console  | MC                            |
| RELEASE (A)   | Remove station from interrupted status   | Console or job stream                                  | TCAM command (Note 5)         |
| REPLY (R)   | Reply to system information requests<br>Specify system parameters<br>Set the time-of-day clock and specify the installation performance specification<br>Specify SMF options<br>Specify DUMP options   | Console or job stream                                  | INFO                          |
| RESET (E)   | Change performance group of a job currently in execution   | Console or job stream                                  | SYS                           |
| SEND (SE)   | Communicate with other operators<br>Communicate with specific time-sharing users<br>Communicate with all time-sharing users<br>Save messages in the broadcast data set for issuance at TSO LOGON time or when requested<br>List messages accumulated in the notices section of the broadcast data set<br>Delete a message from the notices section of the broadcast data set | Console, job stream, or time-sharing terminal (Note 6) | INFO                          |
| <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. For information about MSS commands, see <i>Mass Storage System Extensions Operation</i>.</li> <li>2. CONS command group when message routing is specified.</li> <li>3. For information about VTAM commands, see <i>VTAM Operation</i>.</li> <li>4. For information on using SLIP, see <i>SPL: System Macros and Facilities</i> and <i>Diagnostic Techniques</i>.</li> <li>5. For information about TCAM commands, see <i>TCAM Operation</i>.</li> <li>6. For information about TSO commands, see <i>SPL: TSO</i>.</li> <li>7. I/O command group when specifying a non-console device; CONS when specifying a console.</li> </ol> |  |  |                               |

**Figure 4-1 (Part 4 of 6). System Command Summary**

| Command (Abbr)   | Function  | Acceptable From   | Command Group |
|--|---|---|---------------|
| SET (T)  | Change local time and date<br>Change system resources manager (SRM) parameters<br>Change MPF parameters<br>Restart SMF or change SMF parameters by changing the active SMFPRMxx member of SYS1.PARMLIB<br>Change the dump analysis and elimination (DAE) parameters<br>Change SLIP processing by changing the active IEASLPxx member of SYS1.PARMLIB<br>Change available PFK tables | Console or job stream   | SYS           |
| SETDMN (SD)  | Respecify domain parameters   | Console or job stream   | SYS           |
| SETSMF (SS)  | Change SMF parameters without changing the active SMFPRMxx member of SYS1.PARMLIB   | Console or job stream   | SYS           |
| SLIP (SL)  | Set SLIP traps<br>Modify SLIP traps<br>Delete SLIP traps  | Console, job stream, or time-sharing terminal (Notes 4 and 6) | SYS           |
| START (S)  | Start a job from a console<br>Start an external writer<br>Start the generalized trace facility (GTF)<br>Start TSO/VTAM time-sharing<br>Start LNKLST lookaside (LLA) address space   | Console or job stream   | SYS           |
| STOP (P)   | Stop a job in execution<br>Stop an external writer<br>Stop the generalized trace facility (GTF)<br>Stop TSO/VTAM time-sharing<br>Terminate LNKLST lookaside (LLA) address space   | Console or job stream   | SYS           |
| <b>Notes:</b> <ol style="list-style-type: none"> <li>For information about MSS commands, see <i>Mass Storage System Extensions Operation</i>.</li> <li>CONS command group when message routing is specified.</li> <li>For information about VTAM commands, see <i>VTAM Operation</i>.</li> <li>For information on using SLIP, see <i>SPL: System Macros and Facilities and Diagnostic Techniques</i>.</li> <li>For information about TCAM commands, see <i>TCAM Operation</i>.</li> <li>For information about TSO commands, see <i>SPL: TSO</i>.</li> <li>I/O command group when specifying a non-console device; CONS when specifying a console.</li> </ol> |   |   |               |

Figure 4-1 (Part 5 of 6). System Command Summary

| Command (Abbr)   | Function   | Acceptable From   | Command Group                            |
|--|--|---|--|
| STOPMN (PM)  | Stop continual display of data set status<br>Stop continual display of job status<br>Stop monitoring of time-sharing terminal activity   | Console, job stream, or time-sharing terminal (Note 6)  | INFO                                     |
| STOPTR (PT)  | Halt or reduce information displayed with the TRACK command  | Console   | INFO                                     |
| SWAP (G)   | Move a volume from one device to another   | Console   | I/O                                      |
| SWITCH (I)   | Manually switch recording of SMF data from one data set to another   | Console   | SYS                                      |
| TRACE  | Start, stop, or modify system trace<br>Start, stop, or modify master trace<br>Display the status of system trace and master trace  | Console   | SYS<br>MC<br>SYS                         |
| TRACK (TR)   | Periodically display job information on display consoles   | Console   | INFO                                     |
| UNLOAD (U)   | Remove a volume from system use  | Console or job stream                                   | I/O                                      |
| VARY (V)   | Change the master console<br>Control the hardcopy log<br>Change the status of a secondary console<br>Place I/O devices online or offline<br>Assign and control consoles<br>Place I/O paths online or offline<br>Change a system's participation in a global resource serialization complex<br>Change routing codes for a console | Console<br><br><br>Console or job stream<br><br>Console | (Note 7)<br><br><br>MC<br>(Note 7)<br>MC |
| WRITELOG (W)   | Schedule printing of system log<br>Change system log output class<br>Close system log and discontinue log function<br>Restart system log after closing   | Console or job stream                                   | SYS                                      |
| <p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. For information about MSS commands, see <i>Mass Storage System Extensions Operation</i>.</li> <li>2. CONS command group when message routing is specified.</li> <li>3. For information about VTAM commands, see <i>VTAM Operation</i>.</li> <li>4. For information on using SLIP, see <i>SPL: System Macros and Facilities and Diagnostic Techniques</i>.</li> <li>5. For information about TCAM commands, see <i>TCAM Operation</i>.</li> <li>6. For information about TSO commands, see <i>SPL: TSO</i>.</li> <li>7. I/O command group when specifying a non-console device; CONS when specifying a console.</li> </ol> |  |   |  |

**Figure 4-1 (Part 6 of 6). System Command Summary**

# Command Syntax

You must follow certain rules (“syntax”) when you code commands. As you use code the MVS commands described in Chapter 4, refer to Figure 4-2 for help with the syntax.

| Notation                           | Meaning  | Example  |                            |
|------------------------------------|--|--|----------------------------|
|                                    |  | Book Syntax  | Sample Entry               |
| Apostrophes                        | Must be entered as shown.  | SEND 'message', NOW  | SEND 'listings ready', NOW |
| Single parameter in brackets       | The parameter is optional.   | DISPLAY DMN[= domainum]  | DISPLAY DMN= 5             |
| Stacked items in brackets          | Optional mutually exclusive parameters. Enter one or none.   | CD RESET [ ,SDUMP<br>,SYSABEND<br>,SYSUDUMP<br>,SYSMDUMP<br>,ALL ] | CD RESET,SDUMP             |
| Underline                          | If you do not enter one of the parameters, the system supplies the underlined parameter.           | K T [ ,REF<br>,UTME= nnn ]   | K T                        |
| Stacked items in braces            | You must enter one of the items. You cannot enter more than one.                                   | { MONITOR } { DSNAME<br>MN } { SPACE<br>STATUS }                   | MN SPACE                   |
| Comma                              | Must be entered as shown.  | DISPLAY C,K  | DISPLAY C,K                |
| Ellipsis (...)                     | The parameter can be repeated. Do not enter the ellipsis.  | VARY (devnum[,devnum] ...),ONLINE                                  | VARY (282,283,287),ONLINE  |
| Lower Case                         | A parameter must be substituted. You can enter the parameter in either upper or lower case.        | MOUNT devnum   | MOUNT A30 or mount a30     |
| Parentheses and special characters | Must be entered as shown.  | DUMP COMM =(text)  | DUMP COMM =(PAYROLL)       |
| Upper Case                         | The parameter must be spelled as shown. You can enter the parameter in either upper or lower case. | DISPLAY T  | DISPLAY T or display t     |

Figure 4-2. System Command Syntax

Operator commands can be entered 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'
```

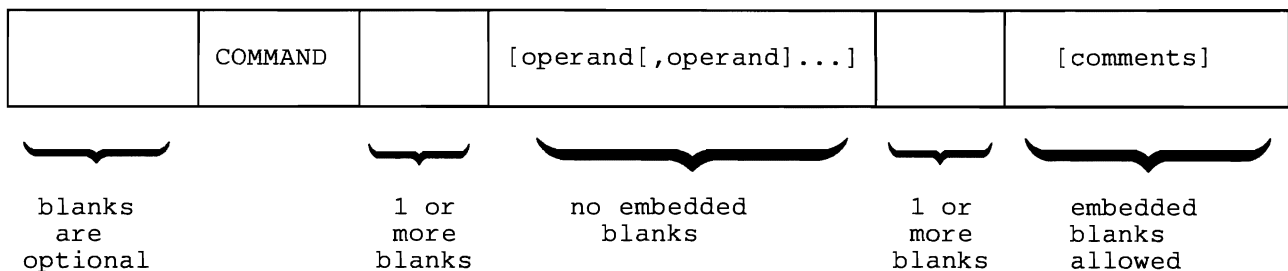
In a JES2 system, you can enter system commands through the input stream (card reader) or through a multiple-console support (MCS) console. In a JES3 system, you can enter system commands through the input stream, through an MCS console, or through a JES3 console. Figure 4-1 indicates which commands you can enter through the input stream. All examples in this book show the format for MCS console entry. See *Operations: JES3 Commands* for details on entering system commands through a JES3 console.

*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 ||b system command when you enter the command within a job. In a JES3 system, enter ||\*\*T, sysname; syscommand.*
2. *Do not use the JES backspace character within a system command.*

## Command Format

The system command format is shown in Figure 4-3.



**Figure 4-3. System Command Format**

*Note:* If you specify no operands on a command, put a blank after the command and a comma and a blank before the comment. Enter only one command per line, with a maximum of:

- 126 characters from a console.
- 80 characters through a card reader.

## CANCEL Command

## CANCEL Command

Use the CANCEL command to immediately terminate:

- A job in execution
- A time-sharing user
- A cataloged procedure started by a START command
- A MOUNT command
- An external writer allocation
- The output processing for a job

*Notes:*

1. *If the CANCEL command fails after several attempts, consider using the FORCE command.*
2. *When global resource serialization is in use (GRS=START or GRS=JOIN was specified at IPL) and the system is either inactive or quiesced, the CANCEL command might not work for jobs that own any global resources.*

The complete syntax for the CANCEL command is:

|   |   |
|---|---|
| $\left\{ \begin{array}{l} \text{CANCEL} \\ \text{C} \end{array} \right\}$ | $\left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{jobname} \\ \text{STARTING} \end{array} \right\} \quad [ , \text{DUMP} ] [ , \text{A}=\text{asid} ] \\ \left\{ \begin{array}{l} \text{U}=\text{userid} \\ \text{U}=\text{*LOGON*} \end{array} \right\} \quad [ , \text{DUMP} ] [ , \text{A}=\text{asid} ] \\ \text{devnum} \\ \text{devicetype} \\ \text{identifier} \end{array} \right\}$ |
|---|---|

### **jobname or STARTING**

The name of the job you want to cancel. Usually, you specify, for *jobname*, the unique name of the job in the system. However, if the job is just starting and does not yet have a unique name, or if other jobs in the system have the same name as the job you want to cancel, you must specify the job name in special ways.

If the job is just starting and does not yet have a name in the system, enter a DISPLAY JOBS,ALL command and note the address space identifiers of those tasks listed as "STARTING." Then, to cancel the starting job, enter CANCEL STARTING,A=asid, specifying for *asid* the address space identifier of the job you want to cancel.

When two or more jobs or started tasks are running with the *jobname* you specify, the system rejects the command unless you specify A=asid along with the *jobname*, where *asid* is the address space identifier of the job you want to cancel.

*Note:* If a job is executing, you can terminate it using either the CANCEL system command or the appropriate subsystem command. However, if the job is not executing, you must cancel the job using the subsystem command.



## **U=userid or U=\*LOGON\***

The identifier of the time-sharing user you want to cancel. Usually, you specify, for *userid*, the unique identifier of the user. If, however, the user is just logging on and does not yet have a unique name, you must specify the identifier in a special way.

Enter DISPLAY TS,ALL to display the identifiers of the logged-on time-sharing users. If the user you want to cancel has a unique identifier in the system, enter CANCEL

U=userid, where *userid* is the unique identifier. If, however, the user is just logging on and does not yet have a unique name, note the address space identifiers of those tasks listed as “\*LOGON\*.” Then, to cancel the user, enter CANCEL

U=\*LOGON\*,A=asid, specifying for *asid* the address space identifier of the user you want to cancel.

## **devnum**

The device number specified when the MOUNT command that you want to cancel was issued.

## **devicetype**

The type of device specified when the MOUNT command that you want to cancel was issued.

*Note:* A MOUNT command for a tape unit can end without the volume being mounted. If the MOUNT command has ended and the mount request is not satisfied, issue the UNLOAD command to free the tape unit.

## **identifier**

The identifier or the device name specified on the START command for the external writer you want to terminate or whose output processing you want to terminate. The identifier can be:

- The identifier specified on a START command for an external writer
- The device type (for example, 3410) or device number specified on a START command for an external writer
- The device type associated with the cataloged procedure specified on a START command for an external writer

Entering this command during device allocation terminates the writer. Entering this command when the writer is processing output for a job terminates the output processing but leaves the writer to process other data sets. You can cancel an output writer, therefore, *only* during device allocation.

## **DUMP**

A dump is to be taken. The type of dump (SYSABEND, SYSUDUMP, or SYSDUMP) depends on the JCL for the job.

## **A=asid**

The hexadecimal address space identifier of the job, time-sharing user, or started procedure you want to cancel.

## CANCEL Command

*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 unit 232.

```
c 232
```

*Example 8:* Cancel the MOUNT command that requests a volume to be mounted on a 3330 device.

```
c 3330
```

*Example 9:* Terminate the device allocation for writer 00E.

```
c 00e
```

*Example 10:* Terminate the output processing being done on device 00E and cause another output data set to be processed.

```
c 00e
```

## CHNGDUMP Command

Use the CHNGDUMP command to change the mode and system dump options list for any dump type. The dump types are SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP. If you issue multiple CHNGDUMP commands, the changes to the system dump options are cumulative.

### Dump Options and Modes

MVS/XA checks the dump mode and dump options each time the system or a user requests a dump. The dump mode determines whether MVS/XA 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/XA 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 SYS1.PARMLIB members IEAABD00, IEADMP00, and IEADMR00 respectively. Because the SDUMP dump type has no corresponding SYS1.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 SYS1.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 SYS1.PARMLIB options *and* any options set by previous CHNGDUMP commands.

## CHNGDUMP Command

- If you change the mode for a dump type from ADD to OVER, the system removes the dump type's SYS1.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.

The complete syntax for the CHNGDUMP command is:

|                              |   |
|------------------------------|---|
| <pre>{ CHNGDUMP   CD }</pre> | <pre> SET [ ,NODUMP      ,ADD      ,OVER      ,SDUMP [(option[,option]...) [ ,Q={ YES  NO } ] ] [ ,NODUMP      ,TYPE={ XMEM              XMEME } ] [ ,OVER      ,ADD ] [ ,SYSABEND [ ,NODUMP [ ,SDATA=(option[,option]...) [ ,PDATA=(option[,option]...) ] ]   ,SYSUDUMP [ ,OVER              ,ADD ] ]   ,SYSMDUMP [(option[,option]...) [ ,NODUMP   ,OVER   ,ADD ] ]  DEL [ ,SDUMP [(option[,option]...) [ ,Q={ YES  NO } ] ] ] [ ,SYSABEND [ [ ,SDATA=(option[,option]...) [ ,PDATA=(option[,option]...) ] ]   ,SYSUDUMP [ ,ALL ] ]   ,SYSMDUMP [(option[,option]...)   ,ALL ]  RESET [ ,SDUMP        ,SYSABEND        ,SYSUDUMP        ,SYSMDUMP        ,ALL ]</pre> |
|------------------------------|---|

## 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.

|                                |   |
|--------------------------------|---|
| <pre>{ CHNGDUMP } { CD }</pre> | <pre>SET {   ,NODUMP   ,ADD   ,OVER   ,SDUMP [(option[,option]...)] [ ,Q={ YES } ] [ ,NODUMP     ,TYPE={ XMEM } [ ,OVER     XMEME ] [ ,ADD ]   { ,SYSABEND [ ,NODUMP ] [ ,SDATA=(option[,option]...)] [ ,PDATA=(option[,option]...)]     ,SYSUDUMP [ ,OVER     ,ADD ]   ,SYSMDUMP [(option[,option]...)] [ ,NODUMP     ,OVER     ,ADD ] }</pre> |
|--------------------------------|---|

### 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 was previously in OVER mode, add its SYS1.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 was previously in ADD mode, remove its SYS1.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.

### SDUMP,OVER or ADD

Set the SDUMP dump mode to the specified mode.

### SDUMP=(options)

Put the specified options in the SDUMP system dump options list.

### 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.

## CHNGDUMP Command

### **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 SVC Dump is to quiesce the system while dumping SQA and CSA.

### **SYSABEND**

Set the SYSABEND dump mode to ADD.

### **SYSABEND,NODUMP**

Set the SYSABEND dump mode to NODUMP.

### **SYSABEND,OVER or ADD**

Set the SYSABEND dump mode to the specified mode.

### **SYSABEND,SDATA = (options)**

Put the specified SDATA options in the SYSABEND system dump options list.

### **SYSABEND,PDATA = (options)**

Put the specified PDATA options in the SYSABEND system dump options list.

### **SYSUDUMP**

Set the SYSUDUMP dump mode to ADD.

### **SYSUDUMP,NODUMP**

Set the SYSUDUMP dump mode to NODUMP.

### **SYSUDUMP,OVER or ADD**

Set the SYSUDUMP dump mode to the specified mode.

### **SYSUDUMP,SDATA = (options)**

Put the specified SDATA options in the SYSUDUMP system dump options list.

### **SYSUDUMP,PDATA = (options)**

Put the specified PDATA options in the SYSUDUMP system dump options list.

### **SYSMDUMP**

Set the SYSMDUMP dump mode to ADD.

### **SYSMDUMP,NODUMP**

Set the SYSMDUMP dump mode to NODUMP.

### **SYSMDUMP,OVER or ADD**

Set the SYSMDUMP dump mode to the specified mode.

### **SYSMDUMP = (options)**

Put the specified options in the SYSMDUMP system dump options list.

## Removing Options from or Resetting the System Dump Options Lists

Use the following form of the CHNGDUMP command to remove specified options from a dump options list or to reset all dump options lists to values specified at system initialization.

|                  |  |
|------------------|--|
| {CHNGDUMP<br>CD} | DEL [<br>,SDUMP [= (option [, option] ...) [ ,Q= { YES } ] ]<br>,TYPE= { XMEM }<br>{ XMEME }<br>,ALL<br>{ ,SYSABEND } [ [ ,SDATA= (option [, option] ...) ] [ ,PDATA= (option [, option] ...) ] ]<br>{ ,SYSUDUMP } [ ,ALL ]<br>,SYSDUMP [= (option [, option] ...) ]<br>{ ,ALL }<br>,ALL         ] |
|------------------|--|

### 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 SYSDUMP from SYS1.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 = (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.

### 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,Q = YES or NO

Specifies whether or not SDUMP is to quiesce the system while dumping SQA and CSA.

### 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 SYS1.PARMLIB member IEAABD00.)

## CHNGDUMP Command

### **SYSABEND,SDATA = (options)**

Remove any of the specified SDATA options that previous CHNGDUMP commands put in the SYSABEND system dump options list. SYS1.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 SYS1.PARMLIB options.

### **SYSABEND,PDATA = (options)**

Remove any of the specified PDATA options that previous CHNGDUMP commands put in the SYSABEND system dump options list. SYS1.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 SYS1.PARMLIB options.

### **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 SYS1.PARMLIB member IEADMP00.)

### **SYSUDUMP,SDATA = (options)**

Remove any of the specified SDATA options that previous CHNGDUMP commands put in the SYSUDUMP system dump options list. SYS1.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 SYS1.PARMLIB options.

### **SYSUDUMP,PDATA = (options)**

Remove any of the specified PDATA options that previous CHNGDUMP commands put in the SYSUDUMP system dump options list. SYS1.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 SYS1.PARMLIB options.

### **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 SYS1.PARMLIB member IEADMR00.)

### **SYSMDUMP = (options)**

Remove any of the specified options that previous CHNGDUMP commands put in the SYSMDUMP system dump options list. SYS1.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 SYS1.PARMLIB options.

### **SYSMDUMP,ALL**

Same as DEL,SYSMDUMP.

### **ALL**

Same as DEL.

## **Options for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP Dump Types**

The options that you can specify on the CHNGDUMP command are as follows (underlined options are the default options from the IEAABD00, IEADMP00, and IEADMR00 members of SYS1.PARMLIB):

### **SDUMP options:**

**ALLNUC** — all of the DAT-on nucleus, including page-protected areas, and all of the DAT-off nucleus.  
**ALLPSA** or **NOALLPSA** — prefix storage area for all processors.  
**NOALL** **NOALLPSA** or **NOALL** specifies that these areas are not to be dumped.

**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.

**SQA** or **NOSQA** — system queue area. **NOSQA** specifies that this data not be dumped.

**SUMDUMP** or **NOSUMDUMP** — **SUM** or **SUMDUMP** requests the summary dump function. See *SPL: System Macros and Facilities* for a description of the summary dump function. **NOSUM** or **NOSUMDUMP** requests that the function not be performed.

**SWA** — scheduler work area.

**TRT** — GTF, system trace, and master trace data.

### **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.

## CHNGDUMP Command

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. See *SPL: System Macros and Facilities* for a description of the summary dump function.  
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.)

# CHNGDUMP Command

## Resetting the Dump Mode to ADD and the Dump Options to Initial Values

Use the following form of the CHNGDUMP command to reset the dump mode to ADD and the dump options list to values established during system initialization.

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{CHNGDUMP} \\ \text{CD} \end{array} \right\}$ | RESET $\left[ \begin{array}{l} , \text{SDUMP} \\ , \text{SYSABEND} \\ , \text{SYSUDUMP} \\ , \text{SYSMDUMP} \\ , \underline{\text{ALL}} \end{array} \right]$ |
|--|---|

### 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 SYS1.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 SYS1.PARMLIB members IEAABD00, IEADMP00, and IEADMR00 respectively. The system empties the SDUMP dump options list.)

### RESET,ALL

Same as RESET.

## Example Showing How CHNGDUMP Commands Affect Dump Modes and Options

Figure 4-4 (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 the SYS1.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.

| FUNCTION (* indicates operator commands/actions)  | OPTIONS  | MODE |
|---|--|------|
| <p><b>* IPL</b></p> <p>During IPL, SYS1.PARMLIB options are set on, and mode is set to ADD.</p>   | CB      IO<br>DM      LSQA<br>ENQ      SUM<br>ERR      TRT                           | ADD  |
| <p><b>ABEND SDATA=(NUC,SWA)</b></p> <p>Because mode is ADD, options specified on dump request are added to options set on in options list to determine areas to dump.</p>   | CB      LSQA<br>DM      NUC<br>ENQ      SUM<br>ERR      SWA<br>IO      TRT           | ADD  |
| <p><b>* CD SET,SYSABEND,SDATA=PCDATA</b></p> <p>An additional option is set on in options list.</p>   | CB      LSQA<br>DM      PCDATA<br>ENQ      SUM<br>ERR      TRT<br>IO                 | ADD  |
| <p><b>ABEND</b></p> <p>Because dump request specified no additional options, only options set on in options list determine areas to dump.</p>   | CB      LSQA<br>DM      PCDATA<br>ENQ      SUM<br>ERR      TRT<br>IO                 | ADD  |
| <p><b>ABEND SDATA=(NUC,SWA)</b></p> <p>Because mode is ADD, options specified on dump request are added to options set on in options list to determine areas to dump.</p>   | CB      NUC<br>DM      PCDATA<br>ENQ      SUM<br>ERR      SWA<br>IO      TRT<br>LSQA | ADD  |
| <p><b>* CD SET,SYSABEND,OVER,SDATA=(CB,IO)</b></p> <p>Mode is changed to OVER and SYS1.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.</p> | CB<br>IO<br>PCDATA   | OVER |
| <p><b>ABEND SDATA=(SWA,TRT)</b></p> <p>Because mode is OVER, options specified on dump request are ignored. Options set on in options list determine areas to dump.</p>   | CB<br>IO<br>PCDATA   | OVER |
| <p><b>* CD SET,SYSABEND,SDATA=(ENQ,LSQA),OVER</b></p> <p>Two more options are set on in options list, and mode is unchanged.</p>  | CB      LSQA<br>ENQ      PCDATA<br>IO  | OVER |
| <p><b>* CD SET,SYSABEND,SDATA=NUC</b></p> <p>Mode is changed to ADD (the default) for SYSABEND, SYS1.PARMLIB options are set on, and CD command options are set on. Previous CD command options remain on.</p>  | CB      LSQA<br>DM      NUC<br>ENQ      PCDATA<br>ERR      SUM<br>IO      TRT        | ADD  |

**Figure 4-4 (Part 1 of 2). Example of How CHNGDUMP Commands Affect Dump Modes and Options**

## CHNGDUMP Command

| FUNCTION (* indicates operator commands/actions)   | OPTIONS  | MODE   |
|--|--|--------|
| ABEND SDATA=(CB,SWA,NUC)<br><br>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<br>DM        PCDATA<br>ENQ       SUM<br>ERR       SWA<br>IO        TRT<br>LSQA | ADD    |
| * CD SET,SYSABEND,NODUMP<br><br>Mode is changed to NODUMP. All options in options list are set off.  |  | NODUMP |
| ABEND SDATA=(CB,SWA,NUC)<br><br>Because mode is NODUMP, request is ignored.  |  | NODUMP |
| * CD SET,SYSABEND,SDATA=SQA<br><br>Mode is changed to ADD. SYS1.PARMLIB options on this CD command are set.  | CB        LSQA<br>DM        SQA<br>ENQ       SUM<br>ERR       TRT<br>IO                      | ADD    |
| * CD DEL,SYSABEND,SDATA=(DM,IO)<br><br>Because DEL is specified, specified options in options list are set off.  | CB        SQA<br>ENQ       SUM<br>ERR       TRT<br>LSQA                                      | ADD    |
| * CD DEL,SYSABEND,SDATA=(SQA,LSQA,TRT,CB,ENQ,ERR,SUM)<br><br>Specified options are set off in options list.  |  | ADD    |
| ABEND SDATA=(SQA)<br><br>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<br><br>Mode and options list are reset to values established at system initialization.   | CB        IO<br>DM        LSQA<br>ENQ       SUM<br>ERR       TRT                             | ADD    |

Figure 4-4 (Part 2 of 2). Example of How CHNGDUMP Commands Affect Dump Modes and Options

# CONFIG Command

Use the CONFIG command to change or check the configuration of the system. You can, with the CONFIG command, change the online or offline status of available processors, Vector Facilities attached to online processors, real and extended storage elements, and channel paths:

1. Directly
2. In response to a configuration display
3. With the options in a CONFIGxx member of SYS1.PARMLIB that you specify

The CONFIG command reconfigures, both logically and physically, available processors, Vector Facilities attached to online processors, real storage ranges, amounts, and elements, and extended storage elements, and channel paths. You can enter the CONFIG command only from the master console.

The complete syntax for the CONFIG command is:

|  |  |   |  |
|--|--|---|--|
| $\left\{ \begin{array}{l} \text{CONFIG} \\ \text{CF} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{CPUAD} \\ \text{CPU} \end{array} \right\}$                                 | $(x[,x] \dots)$   | $\left[ \begin{array}{l} \left[ \text{,ONLINE} \right] \left[ \text{,VFON} \right] \\ \text{,ON} \left[ \text{,VFOFF} \right] \\ \text{,OFFLINE} \\ \text{,OFF} \end{array} \right]$   |
|  | VF   | $(x[,x] \dots)$   | $\left[ \begin{array}{l} \text{,ONLINE} \\ \text{,ON} \\ \text{,OFFLINE} \\ \text{,OFF} \end{array} \right]$   |
|  | $\left\{ \begin{array}{l} \text{STORAGE} \\ \text{STOR} \end{array} \right\}$                              | $\left\{ \begin{array}{l} (\text{ddddM}) \\ (\text{ddddM-ddddM}) \\ (\text{E=id}) \end{array} \right\}$                   | $\left[ \begin{array}{l} \text{,ONLINE} \\ \text{,ON} \\ \text{,OFFLINE} \\ \text{,OFF} \end{array} \right]$   |
|  | ESTOR  | $(\text{E=id})$   | $\left[ \begin{array}{l} \text{,ONLINE} \\ \text{,ON} \\ \text{,OFFLINE} \\ \text{,OFF} \end{array} \right]$   |
|  | CHP  | $\left\{ \begin{array}{l} (\text{xx}) \\ (\text{aa-bb}) \\ (\text{list}) \\ (\text{ALL}, \text{id}) \end{array} \right\}$ | $\left[ \begin{array}{l} \text{,ONLINE} \\ \text{,ON} \\ \left[ \text{,OFFLINE} \right] \left[ \text{,UNCOND} \right] \\ \text{,OFF} \left[ \text{,FORCE} \right] \end{array} \right]$ |
|  | $\left\{ \begin{array}{l} \text{ONLINE} \\ \text{ON} \\ \text{OFFLINE} \\ \text{OFF} \end{array} \right\}$ | $\left[ \text{,L=} \left\{ \begin{array}{l} \text{a} \\ \text{cc} \\ \text{cca} \end{array} \right\} \right]$             |  |
|  | MEMBER[(member id)]  |   |  |

# CONFIG Command

## Reconfiguring the System Directly

Use the following form of the CONFIG command to directly change (without invoking a configuration display or a CONFIGxx member of SYS1.PARMLIB) the online or offline status of available processors, Vector Facilities attached to online processors, real and extended storage elements, or channel paths. The section "Reconfiguring Resources" in Chapter 1 describes how to reconfigure the resources associated with a processor or a processor complex.

|  |   |   |   |  |   |
|--|---|---|---|--|---|
| $\left\{ \begin{array}{l} \text{CONFIG} \\ \text{CF} \end{array} \right\}$ | $\left\{ \begin{array}{l} \\ \\ \\ \\ \\ \end{array} \right.$ | $\left\{ \begin{array}{l} \text{CPUAD} \\ \text{CPU} \end{array} \right\}$    | $(x[,x]...)$  | $\left[ \begin{array}{l} \text{,ONLINE} \\ \text{,ON} \\ \text{,OFFLINE} \\ \text{,OFF} \end{array} \right]$ | $\left[ \begin{array}{l} \text{,VFON} \\ \text{,VFOFF} \end{array} \right]$   |
|  |   | VF  | $(x[,x]...)$  | $\left[ \begin{array}{l} \text{,ONLINE} \\ \text{,ON} \\ \text{,OFFLINE} \\ \text{,OFF} \end{array} \right]$ |   |
|  |   | $\left\{ \begin{array}{l} \text{STORAGE} \\ \text{STOR} \end{array} \right\}$ | $\left\{ \begin{array}{l} (\text{ddddM}) \\ (\text{ddddM-ddddM}) \\ (\text{E=id}) \end{array} \right\}$                   | $\left[ \begin{array}{l} \text{,ONLINE} \\ \text{,ON} \\ \text{,OFFLINE} \\ \text{,OFF} \end{array} \right]$ |   |
|  |   | ESTOR   | $(\text{E=id})$   | $\left[ \begin{array}{l} \text{,ONLINE} \\ \text{,ON} \\ \text{,OFFLINE} \\ \text{,OFF} \end{array} \right]$ |   |
|  |   | CHP   | $\left\{ \begin{array}{l} (\text{xx}) \\ (\text{aa-bb}) \\ (\text{list}) \\ (\text{ALL}, \text{id}) \end{array} \right\}$ | $\left[ \begin{array}{l} \text{,ONLINE} \\ \text{,ON} \\ \text{,OFFLINE} \\ \text{,OFF} \end{array} \right]$ | $\left[ \begin{array}{l} \text{,UNCOND} \\ \text{,FORCE} \end{array} \right]$ |

### CPUAD or CPU

The system is to reconfigure one or more processors, both logically and physically. The system is also to reconfigure, logically and physically, one or more Vector Facilities 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 clock. Each processor's Vector Facility remains in the state it was in when the system last varied the processor offline.

### ONLINE,VFON

The system is to bring online the Vector Facility attached to each specified processor.



## **ONLINE,VFOFF**

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

## **OFFLINE or OFF**

The system is to take the specified processor(s) offline.

## **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 real storage, both logically and physically. Note that storage reconfiguration is not supported on a 4381 processor.

## **ddddM**

The amount of real storage to be reconfigured. Specify up to four decimal digits followed by M (megabytes) for this amount. Check the configuration of your processor to see which size storage elements are supported. The value for *dddd* must be a multiple of the storage increment size (usually 2, 4, or 8).

## **ddddM-ddddM**

The starting and ending addresses of the real storage section to be reconfigured. Specify up to four decimal digits followed by M (megabytes) for each address. The value for each *dddd* must be a multiple of the storage increment size (usually 2, 4, or 8).

## **E = id**

The storage element to be reconfigured, identified by the storage element id (0, 1, 2, or 3). Each storage element is 8, 16, or 32 megabytes in size. Use this parameter only under the direction of a system programmer to take a storage element offline for maintenance and then bring it online again.

## **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

# CONFIG Command

## 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 extended storage elements, both logically and physically.

**(E = id)**

The extended storage element to be reconfigured, identified by its storage element id (0 or 2).

## ONLINE or ON

The system is to bring the specified extended storage element online. The system issues message IEE524 to display the ranges of extended storage that come online. The system rejects the command if you specify an id for an extended 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 extended storage element offline. The system issues message IEE510I to display the ranges of extended storage that went offline with the extended storage element. If the system cannot reconfigure the extended 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 extended 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 extended storage element offline, the system issues a display that tells what address ranges have been removed.*

### CHP

The system is to reconfigure one or more channel paths, both logically and physically.

#### (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, or more 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 3084 or 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.

### 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

To remove the last path to all other devices, use the CONFIG command without the UNCOND or FORCE parameters.

## CONFIG Command

### 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

By specifying OFFLINE,UNCOND, you can 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 the only active console in the system. By specifying OFFLINE,FORCE, you can remove the last path to a device that is in use by a system function, allocated, or a TP device. You cannot do this by specifying OFFLINE or OFFLINE,UNCOND. 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 remove the channel path but leave the affected devices online and allocated. 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 IEA019A 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 finds alternate paths for the devices or terminates I/O, message IEA421E asks you to start the systems you stopped.

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.

*Example 1:* To take processor 2 both logically and physically offline, enter:

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

*Example 2:* To bring online a storage range from 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 both logically and physically online, enter:

```
cf chp(4-9,12),online
```

*Example 5:* To take channel paths 0-6 both logically and physically 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 both logically and physically online, enter:

```
cf chp(all,1),online
```

## CONFIG Command

*Example 7:* To bring processor 1 logically and physically 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 both logically and physically, 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, both logically and physically, issue:

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

*Example 10:* To take offline the extended storage on side 1, issue:

```
cf estor(e=2),offline  
cf estor(e=3),offline
```

*Note:* The extended storage elements on side 1 of a partitioned processor complex have ids of 2 and 3. See Figure 1-3 on page 1-29 for more information about identifying the resources in a 3090 Model 400 processor.

## Reconfiguring the System in Response to a Configuration Display

Use the following form of the CONFIG command to display the processors, total amount of storage, and channel paths available to the system and the online or offline status of each before making any changes to the system configuration. (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, total amount of real storage, real storage elements, extended 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 real and extended storage, and 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.

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{CONFIG} \\ \text{CF} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{ONLINE} \\ \text{ON} \\ \text{OFFLINE} \\ \text{OFF} \end{array} \right\} \left[ , L = \left\{ \begin{array}{l} \text{a} \\ \text{cc} \\ \text{cca} \end{array} \right\} \right]$ |
|--|---|

## ONLINE or ON

The system is to display the system configuration so that you can decide which processors, real and extended storage elements, and channel paths you want to bring online. The system brings online the processors, storage elements, and channel paths you specify in response to message IEE522D.

## OFFLINE or OFF

The system is to display the system configuration so that you can decide which processors, real and extended storage elements, and channel paths you want to take offline. The system takes offline the processors, storage elements, and channel paths you specify in response to message IEE522D.

## L = a, cc, or cca

The display area (a), console (cc), or both (cca) where the system is to display the system configuration. If you do not specify this option, the system displays the system configuration in the first available display area or the message area of the console through which you enter 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, real and extended 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.

# CONFIG Command

## Reconfiguring the System with a CONFIGxx Member of SYS1.PARMLIB

Use the following form of the CONFIG command when you want the system to use options in a CONFIGxx member of SYS1.PARMLIB to change the online or offline status of available processors, storage sections, and channel paths. See *Initialization and Tuning* for detailed information on the syntax and contents of the CONFIGxx member of SYS1.PARMLIB.

|  |                     |
|--|---------------------|
| $\left\{ \begin{array}{l} \text{CONFIG} \\ \text{CF} \end{array} \right\}$ | MEMBER[(member id)] |
|--|---------------------|

### MEMBER

The system is to use a CONFIGxx member of SYS1.PARMLIB 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 a member id, the default member is CONFIG00.

*Example 1:* To reconfigure available processors, real and extended storage, and channel paths in the system to match the options in the CONFIGT3 member of SYS1.PARMLIB, enter:

```
cf member(t3)
```



**CONTROL Command**

Use the CONTROL system command to control consoles. Use this command specifically to:

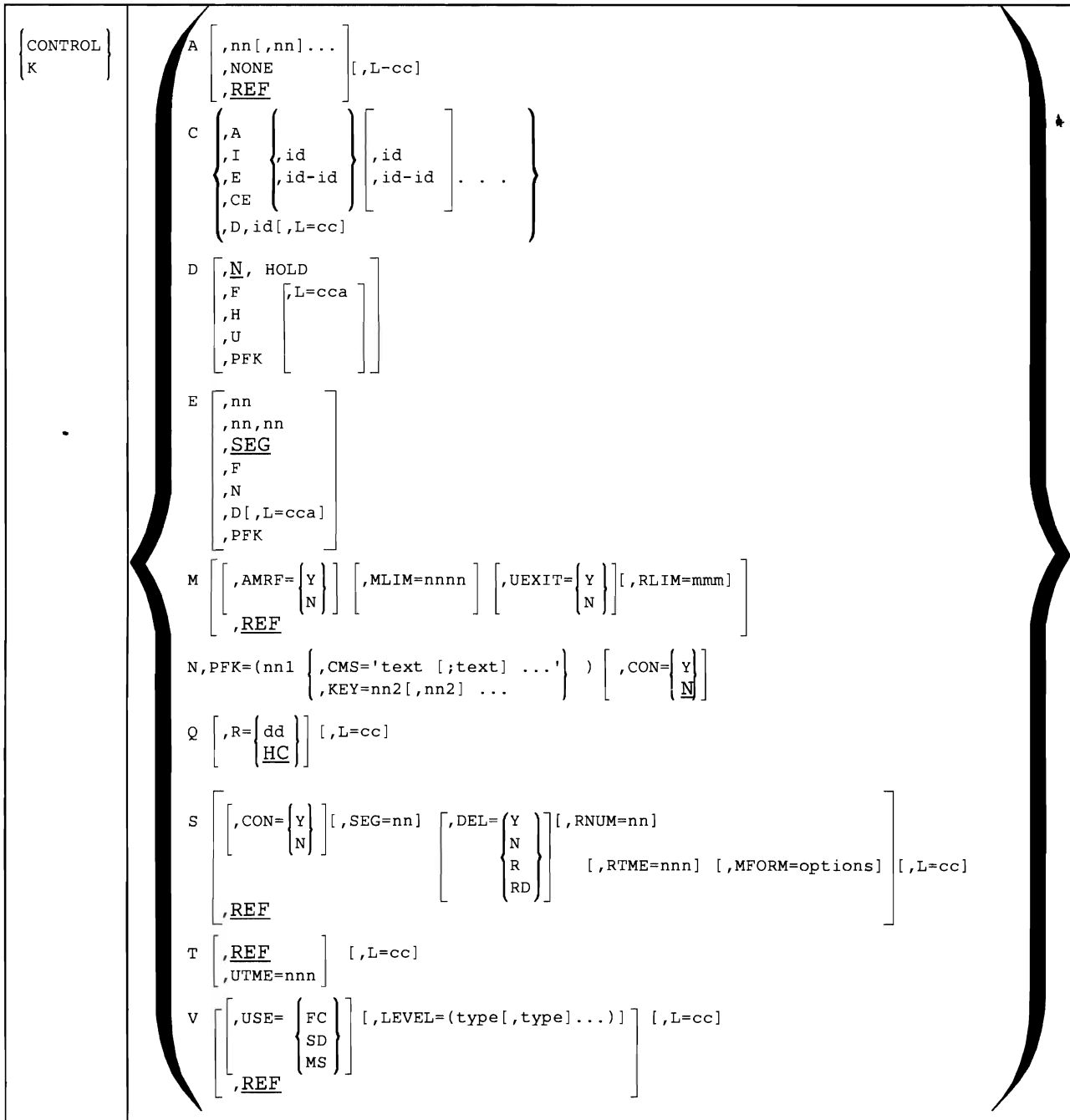
- Alter display area specifications
- Delete retained action messages
- Halt the printing of status displays
- Control displays in areas
- Remove messages from the screen
- Control the status of the action message retention facility
- Change or display the number of allowed message buffers
- Change or display the status of WTO user exit IEAVMXIT
- Change a PFK definition
- Reroute message queues
- Change or display message deletion and format specifications
- Change or display time intervals for dynamic displays
- Select the operating mode of a console
- Change the message levels for a console

Listed on the next page are the keywords and operands used on the CONTROL command followed by a description of their functions. See Chapter 3 of this book for more information about the functions of the CONTROL command.

Many of the functions of the CONTROL command are controlled at IPL by parameters in the CONSOLxx member of SYS1.PARMLIB. 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 “Using CONSOLxx to Control Console Characteristics” on page 3-2.

# CONTROL Command

The complete syntax for the CONTROL command is:



## Changing Display Area Specifications

Use the CONTROL A command to change, remove, or examine display area specifications for any display console in the system.

The AREA parameter on the CONSOLE statement in the CONSOLxx member of SYS1.PARMLIB controls the display area specifications at IPL. Figure 3-12 shows the maximum message area sizes for all devices that MVS/XA supports as consoles and the defaults for the AREA parameter.

|  |  |
|--|--|
| $\left\{ \begin{array}{l} \text{CONTROL} \\ \text{K} \end{array} \right\}$ | A $\left[ \begin{array}{l} ,nn[ ,nn] \dots \\ ,NONE \\ ,REF \end{array} \right]$ $[ ,L=cc ]$ |
|--|--|

### 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 11.

#### NONE

No display areas are defined for the console.

#### REF

The display area sizes are to be displayed in the entry area in CONTROL command form.

#### L=cc

The display console whose display area is to be changed or referenced. Use this operand when changing display area specifications for output-only consoles. 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
```

# CONTROL Command

## Deleting Retained Action Messages

Use the following form of the CONTROL command to delete the outstanding action messages that the action message retention facility has retained.

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{CONTROL} \\ \text{K} \end{array} \right\}$ | $C \left\{ \begin{array}{l} ,A \\ ,I \\ ,E \\ ,CE \end{array} \right\} \left\{ \left\{ ,id \right\} \left[ ,id-id \right] \dots \right\}$ |
|--|---|

### C

The system is to delete one or more action messages that the action message retention facility has retained.

### A

The system is to delete one or more outstanding retained immediate action messages (descriptor code 1 or 2), eventual action messages (descriptor code 3), or critical eventual action messages (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 identification number of the message to be deleted. This number is the identification number listed in response to the DISPLAY R command.

### id-id

The one-to-ten-digit decimal 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, but the requests relating to them may still be outstanding. When a range is specified, certain conditions are not flagged as errors that would be errors if individual identification numbers were specified. For example, message identification numbers that do not exist but are within the range are not flagged. If you specify a non-existent identification number by itself, you receive an error message.

*Example 1:* To delete all immediate action, eventual action, and critical eventual action messages that have been retained with identification numbers from 0 to 99999, enter:

```
CONTROL C,A,0-99999
```

*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 messages and then use K C,E to delete the three messages.

```
DISPLAY R,E
K C,E,id,id,id
```

*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
```

## Halting the Printing or the Display of a Status Display

Use the following form of the CONTROL command to halt the printing or the display of a status display:

- On a printer or typewriter console that is not the hardcopy log
- On a display console that does not have display areas

The following command must be entered while the display is being printed or displayed:

|  |                |
|--|----------------|
| $\left\{ \begin{array}{l} \text{CONTROL} \\ \text{K} \end{array} \right\}$ | $C,D,id[L=cc]$ |
|--|----------------|

### **C,D**

The inline display, indicated by the id operand, is to be stopped.

### **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=cc**

The identification number of the active MCS console where the display is to be stopped.

*Example:* 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
```

# CONTROL Command

## Controlling Displays in Areas

Use the following form of the CONTROL command to control displays, message numbering, and the PFK display line:

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{CONTROL} \\ \text{K} \end{array} \right\}$ | D $\left[ \begin{array}{l} ,N, \text{HOLD} \\ ,F \\ ,H \\ ,U \\ ,\text{PFK} \end{array} \right] \left[ ,L=\text{cca} \right]$ |
|--|---|

### D

Information is to be displayed on the screen.

### N

Messages on the screen are to be consecutively numbered. The numbers are to be removed from the screen when the operator deletes a message or performs a cancel action.

### HOLD

In conjunction with the N operand, HOLD specifies that consecutive numbers are to be displayed for each message on the screen and that messages are to be re-numbered after each message deletion that the operator performs.

### F

The next frame of a status display is to be displayed.

### H

The updating of a dynamic status display is to be suppressed.

### U

The updating of a dynamic status display is to be resumed.

### 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.

### L = cca

The console and display area where the the information is to be displayed.

*Example:* To display the next frame of a status display, enter:

k d,f

## Removing Information From the Screen

Use the following form of the CONTROL command to remove various types of messages from the screen or to cause message numbers to be deleted:

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{CONTROL} \\ \text{K} \end{array} \right\}$ | E $\left[ \begin{array}{l} ,nn \\ ,nn,nn \\ ,\underline{\text{SEG}} \\ ,F \\ ,N \\ ,D[ ,L=cca] \\ ,PFK \end{array} \right]$ |
|--|---|

### E

Messages are to be removed from the screen.

### nn

Single messages (nn) or deletable messages within a range of messages (nn,nn) are to be removed from the screen. The value of nn must be a decimal number from 1 to the highest message number on the screen. *nn,nn* must be a range of numbers.

### SEG

Deletable messages in the predefined message segment are to be removed from the screen.

### F

All flagged messages (messages marked with a vertical or horizontal bar in position 3) are to be removed from the screen.

### N

Messages on the screen are to be consecutively numbered. The numbers are to be removed when the operator deletes a message or performs a cancel action.

### D

Specifies that a status display is to be deleted.

### L = cca

For the D operand, specifies the console and display area where the specified action is to take place.

### PFK

Specifies that the numbers are to be erased from the program function key (PFK) display line.

*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 nonaction messages from a segment of messages, enter:

```
K E,SEG or K E
```

## CONTROL Command

*Example 3:* To delete the nonaction messages 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
```

### Activating, Deactivating, or Displaying the Status of the Action Message Retention Facility

Use the following form of the CONTROL 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 subsystem console associated with the master console.

The AMRF parameter on the INIT statement in the CONSOLxx member of SYS1.PARMLIB 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:

|  |  |
|--|--|
| $\left\{ \begin{array}{l} \text{CONTROL} \\ \text{K} \end{array} \right\}$ | M $\left[ \begin{array}{l} , \text{AMRF} = \left\{ \begin{array}{l} \text{Y} \\ \text{N} \end{array} \right\} \\ , \text{REF} \end{array} \right]$ |
|--|--|

#### M

Indicates that this command is a master console command.

#### AMRF =

The status of the action message retention facility is to be modified.

#### Y

The action message retention facility is to become active.

#### N

The action message retention facility is to be deactivated.

#### REF

The current values for CONTROL M operands are to be displayed in the entry area of the master console in CONTROL command form if the master console is a display console. If the master console is not a display console, the operands and values are to be printed in message IEE144I. If no operands are specified, REF is the default.

*Example:* To activate the action message retention facility, enter:

```
K M,AMRF=Y
```



## Changing or Displaying the Number of Allowed WTO and WTOR Message Buffers

Use the following form of the CONTROL 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 the master console, or from a subsystem console associated with the the master console.

The MLIM and RLIM parameters on the INIT statement in the CONSOLxx member of SYS1.PARMLIB 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:

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{CONTROL} \\ \text{K} \end{array} \right\}$ | $M \left[ , \text{MLIM}=\text{nnnn} \left[ , \text{RLIM}=\text{mmm} \right] \right]$<br>$\left[ , \text{REF} \right]$ |
|--|---|

### M

Indicates that this command is a master console command.

### MLIM = nnnn

The maximum number of WTO message buffers to be allowed, 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 a JES3 system, do not set this value higher than the value of the JES3 staging area count. See the MAINPROC statement in *JES3 Initialization and Tuning*.

### RLIM = mmm

The maximum number of outstanding WTOR messages that the system holds in buffers, where *mmm* can be a decimal number from 5 to 100. Each WTOR buffer holds a WTOR message that the system has displayed but that the operator has not responded to.

### REF

The current values for CONTROL M operands are to be displayed in the entry area of the master console in CONTROL command form if the master console is a display console. If the master console is not a display console, the operands and values are to be printed in message IEE144I. If no operands are specified, REF is the default. Unless you specifically change it, the MLIM and RLIM values that the system displays are the values that were specified on the INIT statement in CONSOLxx member of SYS1.PARMLIB.

*Example:* To allow the maximum number of WTO message buffers and WTOR message buffers, enter:

```
K M,MLIM=9999,RLIM=100
```

# CONTROL Command

## Changing or Displaying the Status of WTO User Exit IEAVMXIT

Use the following form of the CONTROL command to change or display the status of the WTO user exit IEAVMXIT. This exit receives control when the system issues a WTO message unless your installation names another WTO user exit routine for the message. To learn what messages are currently processed by IEAVMXIT, issue the DISPLAY MPF command.

You can enter the CONTROL M,UEXIT command from the master console, from a subsystem console associated with the master console, or from a pseudo-master console.

The UEXIT parameter on the INIT statement in the CONSOLxx member of SYS1.PARMLIB controls whether IEAVMXIT is active at IPL. If the UEXIT parameter is not coded, IEAVMXIT is active.

The syntax of the command is:

|  |   |   |
|--|---|---|
| $\left\{ \begin{array}{l} \text{CONTROL} \\ \text{K} \end{array} \right\}$ | M | $\left[ \begin{array}{l} ,\text{UEXIT}=\left\{ \begin{array}{l} \text{Y} \\ \text{N} \end{array} \right\} \\ ,\text{REF} \end{array} \right]$ |
|--|---|---|

### M

Indicates that this command is a master console command.

### UEXIT =

The status of the general WTO user 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 LNKST lookaside (LLA), and then reactivate IEAVMXIT.

### N

The general user exit routine IEAVMXIT is to become inactive.

### REF

The current values for CONTROL M operands are to be displayed in the entry area of the master console in CONTROL command form if the master console is a display console. If the master console is not a display console, the operands and values are to be printed in message IEE144I. If you do not specify any operands on the K M command, REF is the default.

*Example 1:* To activate the general WTO user exit routine IEAVMXIT, enter:

```
K M,UEXIT=Y
```

*Example 2:* To load a new copy of the general WTO user 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 LNKLST lookaside (LLA) is refreshed, enter:

```
K M,UEXIT=Y
```

to activate the new copy.

## Changing a PFK Definition

Use the following form of the CONTROL 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 reside in a PFK table in a PFKTABxx member of SYS1.PARMLIB.

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 3-31.

Use the following form of the CONTROL command to define commands for program function keys (PFKs) or assign a PFK table.

|  |  |
|--|--|
| $\left\{ \begin{array}{l} \text{CONTROL} \\ \text{K} \end{array} \right\}$ | $N,PFK = \left\{ \begin{array}{l} (nn1 \left[ \begin{array}{l} ,CMD='text[;text]...' \\ ,KEY=nn2[,nn2]... \end{array} \right] ) \left[ \begin{array}{l} ,CON = \left\{ \begin{array}{l} Y \\ N \end{array} \right\} \end{array} \right] \\ nnnnnnnn[,L=cc] \end{array} \right\}$ |
|--|--|

### 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 generation.

### 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 by means of 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.

## CONTROL Command

### **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 including 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**

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 console 8, enter:

```
K N ,PFK=PFK22 ,L=8
```

## 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 hardcopy log. It affects only messages currently on the queue. Subsequent messages are queued as usual.

The command gives you extended control over MCS console message queues. If there is a WTO buffer shortage, you can requeue the messages to speed up console processing or alleviate storage problems.

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 hardcopy log.
- All inline 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 hardcopy device. First, you must use the VARY command to direct hardcopy to another device. Then, you can requeue the messages being routed to the hardcopy device.

In addition to being routed to another console, 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 inline 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:

|  |  |
|--|--|
| $\left[ \begin{array}{c} \text{CONTROL} \\ \text{K} \end{array} \right]$ | $Q, R = \left[ \begin{array}{c} \text{dd} \\ \text{HC} \end{array} \right] \quad [ , L = cc ]$ |
|--|--|

**Q**

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

**R**

The message queue is to be rerouted.

## CONTROL Command

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

**HC** The hardcopy device is to receive the rerouted messages.

**L=cc** The id 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:* To reroute any messages on the issuing full-capability console's queue to console 03 and to the hardcopy log, enter:

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

*Note:* 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 redisplayed with some of the lines missing.

## 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 member of SYS1.PARMLIB control, at IPL, the same console specifications as the CONTROL S command. The system defaults are in parentheses.

**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. (See Figure 3-10 on page 3-35 for the defaults.)

The complete syntax for the CONTROL S command is:

|  |  |
|--|--|
| $\left\{ \begin{array}{l} \text{CONTROL} \\ \text{K} \end{array} \right\}$ | $S \left[ \left[ \left[ \begin{array}{l} \text{CON} \\ \text{Y} \\ \text{N} \end{array} \right] \right] \left[ \text{,SEG=nn} \right] \left[ \left[ \begin{array}{l} \text{DEL} \\ \text{Y} \\ \text{N} \\ \text{R} \\ \text{RD} \end{array} \right] \right] \left[ \text{,RNUM=nn} \right] \left[ \text{,RTME=nnn} \right] \left[ \text{,MFORM=options} \right] \left[ \text{,L=cc} \right] \right. \\ \left. \left[ \text{,REF} \right] \right]$ |
|--|--|

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

**CON** Conversational message deletion is requested or cancelled.

## CONTROL Command

**Y**  
Requests conversational message deletion.

**N**  
Requests conversational message deletion. (non-conversational message deletion is to go into effect).

**REF**  
The current console specification values are to be displayed in the entry area in CONTROL command form. **REF** is the default value.

**SEG = nn**  
The size of the message segment is to be altered, where *nn* specifies the number of lines to be included in the segment of messages deleted when a CONTROL E,SEG command is entered. The *nn* value is a number from one to the number of lines in the message area.

**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.

**RNUM = nn**  
The number of lines in the message roll. The *nn* value is a decimal number from 1 to the number of lines in the message area.

**RTME = nnn**  
The time interval in seconds between message rolls. The *nnn* value can be any decimal number from 1 to 999.

**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

## CONTROL Command

You can enter more than one of the operands. If you do, place parentheses around the operands 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.

- |          |   |
|----------|---|
| <b>T</b> | Requests that each message appear with a time stamp.  |
| <b>S</b> | Requests that each message appear with the name of the system that sent the message.  |
| <b>J</b> | Requests that each message appear with the job name or job ID of its issuer.  |
| <b>M</b> | 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 member of SYS1.PARMLIB is not coded, the system displays the message text without timestamp, system name, or job name/ job id. |

### **L = cc**

The console this command is to affect. Before using the L operand, understand that:

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

*Example 1:* To set SEG equal to 10 lines, enter:

```
K S,SEG=10
```

*Example 2:* To cancel roll mode on console 10, enter:

```
CONTROL S,DEL=N,L=10
```

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 display all messages on the full-capability console 2 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=2
```

*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 following form of the CONTROL 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 member of SYS1.PARMLIB controls the time interval value at IPL. If the UTME parameter is not coded, the time interval is 30 seconds.

The syntax for the CONTROL T command is:

|                          |   |
|--------------------------|---|
| <pre>{ CONTROL K }</pre> | <pre>T [ ,REF       ,UTME=nnn ] [ ,L=cc ]</pre> |
|--------------------------|---|

**T**

The time interval for updating status displays is to be changed or displayed.

**REF**

The current value of the time interval is to be displayed in the entry area in CONTROL command form. **REF** is the default.

**UTME**

The time interval for updating status displays in seconds, where *nnn* can be any decimal number from 10 to 999.

**L = cc**

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.

# CONTROL Command

## Changing the Operating Mode of a Console

Use the CONTROL V command to change the operating mode of a console.

The USE parameter on the CONSOLE statement in the CONSOLxx member of SYS1.PARMLIB 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.

The syntax for the CONTROL V command is:

|  |   |  |                             |
|--|---|--|-----------------------------|
| $\left\{ \begin{array}{l} \text{CONTROL} \\ \text{K} \end{array} \right\}$ | V | $\left[ \begin{array}{l} , \text{USE} = \left\{ \begin{array}{l} \text{FC} \\ \text{SD} \\ \text{MS} \end{array} \right\} \\ , \text{REF} \end{array} \right]$ | $[ , \text{L} = \text{cc}]$ |
|--|---|--|-----------------------------|

### V,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:* 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.

### REF

The system is to display the current COMMAND V operand values in the CONTROL V command form in the entry area of the screen. REF is the default value.

### L = cc

The console where the specified action is to take place.

## 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 member of SYS1.PARMLIB 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:

|  |  |
|--|--|
| $\left\{ \begin{array}{l} \text{CONTROL} \\ \text{K} \end{array} \right\}$ | $V \left[ \begin{array}{l} ,LEVEL=(type[,type]...) \\ ,REF \end{array} \right] \quad [ ,L=cc]$ |
|--|--|

### 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 operands and separate them with commas.

|        |  |
|--------|--|
| 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 <b>not</b> 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 log. |

#### Notes:

1. If an informational message is not directed to any console, it is routed to the hardcopy 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 master console and the hardcopy log.
4. If a message is directed to a specific console, it appears there regardless of the message level of the console.

## CONTROL Command

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

The system is to display the current CONTROL V command values in the CONTROL V command form in the entry area. **REF** is the default value.

### L = cc

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 20, enter:

```
K V,LEVEL=IN,L=20
```

*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)
```

In this case, broadcast messages and informational messages with certain routing codes will be sent directly to the hardcopy log without appearing at the issuing console. If your consoles are set up so that informational messages with certain routing codes, and/or broadcast messages are not sent to any 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.

This example is of special interest to those installations that use consoles to print tape labels.

## DEVSERV Command

Use the DEVSERV command to display the status of DASD and tape devices. The command displays the following information for one device or a group of devices:

- the logical mode of the device
- the number of data sets allocated on the volume
- the volume serial label
- the channel path id
- the status of the path

To compare the DEVSERV PATHS command with the DISPLAY U and DISPLAY M commands, see “Display the Status of Devices and Availability of Paths” in Chapter 1.

You can use this command only if you have installed MVS/XA DFP Version 2.

The complete syntax for the DEVSERV command is:

|   |   |  |  |                    |   |
|---|---|--|--|--------------------|---|
| $\left\{ \begin{array}{l} \text{DEVSERV} \\ \text{DS} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{PATHS} \\ \text{P} \end{array} \right\}, \text{ddd} [ , \text{nn}]$ | $\left[ \begin{array}{l} , \text{ONLINE} \\ , \text{ON} \\ , \text{OFFLINE} \\ , \text{OFF} \end{array} \right]$ | $\left[ \begin{array}{l} , \text{NOSYM} \\ , \text{NOS} \end{array} \right]$ | $[ , \text{DUMP}]$ | $\left[ , \text{L} = \left\{ \begin{array}{l} \text{a} \\ \text{cc} \\ \text{cca} \end{array} \right\} \right]$ |
|---|---|--|--|--------------------|---|

### PATHS or P

The system is to display (in message IEE459I) the status of specified devices.

#### ddd

The device number, in hexadecimal, for which the system is to display the information. You can specify any device that MVS/XA supports.

#### 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 system displays information about the one device you specify.

### ONLINE or ON

The system is to display information about only those specified devices that are online. If you do not specify ONLINE or OFFLINE, the system displays information about both online and offline devices.

### OFFLINE or OFF

The system is to display information about only those specified devices that are offline. If you do not specify ONLINE or OFFLINE, the system displays information about both online and offline devices.

### NOSYM or NOS

The system is not to display the definitions of the symbols with message IEE459I. If you omit NOSYM, the system displays the definition of all symbols.

## DEVSERV Command

### DUMP

The system is to issue an SVC dump after it executes the DEVSERV PATHS command. If you specify both DUMP and a value for nn, the system ignores the value for nn.

### L = a, cc, or cca

The display area (a), console (cc), or both (cca) where the display is to be presented. If you omit this operand, the display is presented in the first available display area or the message area of the console through which you entered the command.

*Example 1:* To display the status of a DASD with the number 380, enter:

```
DS P,380
```

*Example 2:* To display the status of all devices numbered 380 through 38F that are online, enter:

```
DS P,380,16,ON
```

## DISPLAY Command

Using the DISPLAY system command, you can display information about the operating system, the jobs and application programs that are running, the processor, devices that are online and offline, real and extended storage, and the time of day. Figure 4-5 summarizes the information that the DISPLAY command provides. Use it to access the pages on which you can find details about a particular use of the DISPLAY command.

| To display:   | Use this command:                       | On page: |
|---|---|----------|
| Information about the page and swap data sets currently in use by the system  | DISPLAY ASM                             | 4-60     |
| Status and allocation of resources, such as processors, Vector Facilities, and storage devices  | DISPLAY U                               | 4-62     |
| System configuration information, including the contents and status of a side and the location and length of the hardware system area (HSA) | DISPLAY M                               | 4-64     |
| The configuration and status of system consoles   | DISPLAY CONSOLES                        | 4-69     |
| CONTROL command functions   | DISPLAY C,K                             | 4-72     |
| Dump options or dump data set status  | DISPLAY DUMP                            | 4-72     |
| The number of active jobs, time-sharing users, and active initiators  | DISPLAY JOBS<br>DISPLAY TS<br>DISPLAY A | 4-75     |
| System requests and the status of the action message retention facility   | DISPLAY R                               | 4-78     |
| The status of SMF data sets or the SMF options currently in effect  | DISPLAY SMF                             | 4-82     |
| Available PFK tables and PFK definitions in a specified table or for a specified console  | DISPLAY PFK                             | 4-83     |
| The current time and date in local time and Greenwich mean time   | DISPLAY T                               | 4-84     |
| Information about SLIP traps  | DISPLAY SLIP                            | 4-84     |
| Domain information and the name of the current installation performance specifications (IPS)  | DISPLAY DMN                             | 4-85     |
| Message suppression, retention, color, intensity, and highlighting options  | DISPLAY MPF                             | 4-86     |
| Configuration, resource name list (RNL), and resource information in the global resource serialization complex                              | DISPLAY GRS                             | 4-87     |

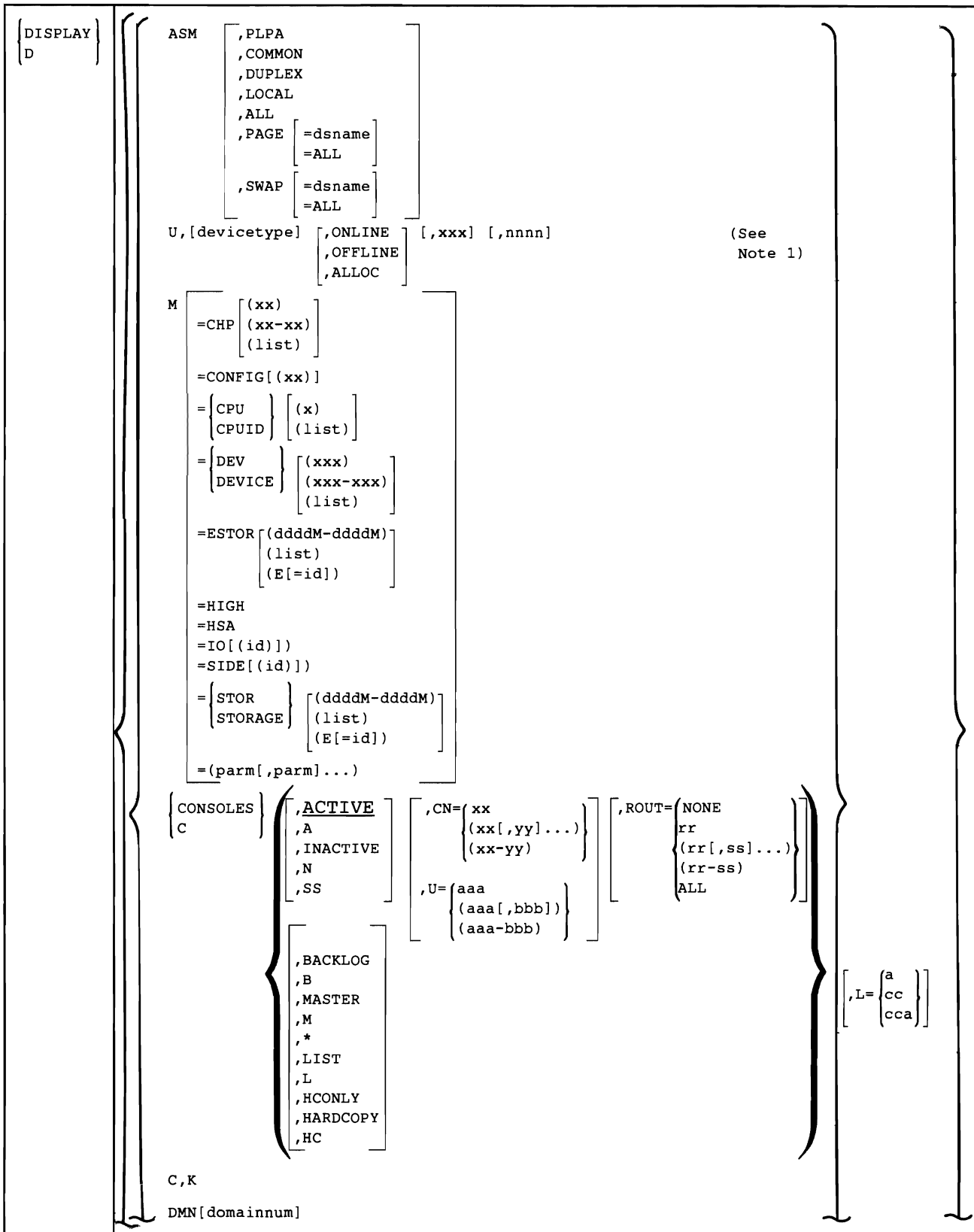
Figure 4-5. Summary of the DISPLAY Command

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

- TCAM teleprocessing functions and activity. See *TCAM Operation*.
- VTAM network activity and functions. See *VTAM Operation*.
- 3850 activity and functions. See *Mass Storage System Extensions Operation*.

# DISPLAY Command

The complete syntax for the DISPLAY command is:





|   |  |  |  |
|---|--|--|--|
| <pre>{ DISPLAY D (cont'd.</pre>   | <pre>{ DUMP D   { ,STATUS     { ,ST       { ,S     }   }   { ,TITLE     { ,T   }   { ,ERRDATA     { ,ER       { ,E     }   }   { ,OPTIONS     { ,O   }   { ,DSN=     { nn       { (nn[,nn]...)         { nn-nn           { (nn-nn[,nn-nn]...)             { (nn[,nn]...nn-nn[,nn-nn]...)               { ALL                 { (ALL)             }           }         }       }     }   } }</pre>   | <pre>(See Note 1)</pre>  |  |
| <pre>{ TS   { J[OBS]   { A     { ,L       { ,LIST         { ,ALL           { ,A             { ,NAME           }         }       }     }   } }</pre> | <pre>R   { ,U     { ,KEY [,SYS=sysname] [,CN=(ALL)]   }   { ,I     { ,E       { ,CE         { ,R           { ,M             { ,LIST               { ,L                 { ,ALL                   { ,A                     { ,                 }               }             }           }         }       }     }     { ,T       { ,S         { ,J           { ,M         }       }     }     { ,MSG=msgid [,SYS=sysname] [,KEY=keyname]   }   { ,CN=     { xx       { (ALL)     }   }   { ,ROUT=     { ALL       { (rrr[,sss]...)         { (rrr-sss[,rrr-sss]...)       }     }   } }</pre> | <pre>(See Note 3)</pre>  |  |
| <pre>SMF { ,S       { ,O     }</pre>  | <pre>SLIP [=xxxx] PFK [ { ,TABLE       { ,T     }       [,CN=cc     ]</pre>  | <pre>MPF [ ,MSG       { ,M         { ,COLOR           { ,C         }       }     ]</pre> | <pre>GRS [ ,SYSTEM       { ,LINK         { ,ALL           { ,A         }       }       { ,CONTENTION         { ,C       }       { ,RNL=         { (CONVERSION)           { CON             { C               { EXCLUSION                 { EXCL                   { E                     { INCLUSION                       { INCL                         { I                           { ALL                             { A                           }                         }                       }                     }                   }                 }               }             }           }         }       }     ]</pre> |
| <pre>{ ,RES= ( { qname             { *           }           { ,rname             { ,*           }         }         { ,HEX       }</pre>           | <pre>T TP (see Note 2) NET 3850</pre>  |  |  |

# DISPLAY Command

Notes:

1. All commas between *DISPLAY U* or *DISPLAY R* and a specified positional operand must be supplied. For example, *DISPLAY U,,ONLINE*.
2. See *TCAM Operation*, *VTAM Operation*, and *Mass Storage System Extensions Operation for the TP, NET, and 3850 DISPLAY command operands*.
3. *L=a, cc, and cca* applies to operands on this page.

The *DISPLAY* command can include the location operand, which is used to route status displays to a specified display area or console. The syntax for the location operand is as follows:

$$L = \left\{ \begin{array}{l} a \\ cc \\ cca \end{array} \right\}$$

This operand specifies the display area (*a*), console (*cc*), or both (*cca*) where the display is to be presented. You must specify, for *cc*, 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 and Swap Data Set Information

Use the *DISPLAY ASM* command to identify the page and swap datasets 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 *IIEE200I* for each data set of the specified type that the system is currently using:

- Type of data set
- Percent full
- Status (that is, bad, full, or ok)
- Device number
- Data set name

If you request information about the *PLPA*, common, or duplex data set, or about a specific page or swap data set, you receive all the preceding information, plus:

- Volume serial number
- Device type
- Data set size (in slots or swap sets)
- The number of slots or swap sets that are currently in use
- The number of slots or swap sets that are currently available
- The 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 and swapping configuration; for this information, see "Page/Swap Data Set Activity Report" in *RMF Monitor I and II Reference and Users Guide*.

|  |   |   |
|--|---|---|
| $\left\{ \begin{array}{l} \text{DISPLAY} \\ \text{D} \end{array} \right\}$ | ASM $\left[ \begin{array}{l} , \text{PLPA} \\ , \text{COMMON} \\ , \text{DUPLEX} \\ , \text{LOCAL} \\ , \text{ALL} \\ \\ , \text{PAGE} \left[ \begin{array}{l} =\text{dsname} \\ =\text{ALL} \end{array} \right] \\ \\ , \text{SWAP} \left[ \begin{array}{l} =\text{dsname} \\ =\text{ALL} \end{array} \right] \end{array} \right]$ | $\left[ , \text{L} = \left\{ \begin{array}{l} \text{a} \\ \text{cc} \\ \text{cca} \end{array} \right\} \right]$ |
|--|---|---|

## ASM

The system is to display information about the page and/or swap data sets the system is currently using. If you specify DISPLAY ASM with no operands, the system displays information about all page and swap data sets that it is currently using.

## PLPA

Requests information about the PLPA page data set.

## COMMON

Requests information about the common page data set.

## DUPLEX

Requests information about the duplex page data set, if it is in use.

## LOCAL

Requests information about any local page data set.

## ALL

Requests information about all page and swap data sets.

## PAGE

Requests information about page data sets.

## ALL

Requests information about all page data sets. ALL is the default for the PAGE operand.

## dsname

Requests information about the page data set named **dsname**.

## SWAP

Requests information about swap data sets.

## ALL

Requests information about all swap data sets, if any. ALL is the default for the SWAP operand.

## dsname

Requests information about the swap data set named **dsname**.

## DISPLAY Command

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

This operand specifies the display area (a), console (cc), or both (cca) where the display is to be presented.

*Example 1:* To display summary information about all page and swap data sets, enter:

```
DISPLAY ASM,ALL or D ASM
```

*Example 2:* To display detailed information about the PLPA data set, enter:

```
D ASM,PLPA
```

## 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
- A range of devices

The DISPLAY U command can also display the jobnames and ASIDs of device users. Knowing the jobs and ASIDs using a particular device can help you determine why you cannot take a particular device offline.

|  |                  |   |        |         |   |
|--|------------------|---|--------|---------|---|
| $\left\{ \begin{array}{l} \text{DISPLAY} \\ \text{D} \end{array} \right\}$ | U, [devicetype], | $\left[ \begin{array}{l} \text{ONLINE} \\ \text{OFFLINE} \\ \text{ALLOC} \end{array} \right]$ , | [xxx], | [nnnn], | $\left[ \text{L} = \left\{ \begin{array}{l} \text{a} \\ \text{cc} \\ \text{cca} \end{array} \right\} \right]$ |
|--|------------------|---|--------|---------|---|

*Notes:*

1. *Supply all commas between DISPLAY U and a specified operand. For example, DISPLAY U,,OFFLINE.*
2. *To compare the DEVSERV PATHS command with the DISPLAY U command, see "Displaying the Status of Devices and Availability of Paths" in Chapter 1 of this book.*

**U**

The system is to display unit status information about all devices or about specified devices, including non-supported devices (those devices specified at system generation in the IODEVICE macro with the DUMMY = parameter) and channel-to-channel (CTC) adapters.

## **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 devices  
*ALL* — all above options displayed (equivalent to specifying DISPLAY U)

## **ONLINE**

The system is to display information (in message IEE450I) 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 IEE450I) 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. Allocation information includes the jobname and address space identifier (ASID) of each job to which the device is currently allocated.

*Note:* If you do not specify ONLINE, OFFLINE, or ALLOC, the system displays status information, without allocation information, about both online and offline devices.

## **xxx,nnnn**

The system is to display unit status information about devices starting with device number *xxx* for *nnnn* number of devices. If you omit the device number (*xxx*), the system assumes the starting device number is 000. If you omit the number of devices (*nnnn*) and do not specify ALLOC, the system assumes the number of devices is 16. If you omit the number of devices (*nnnn*) and specify ALLOC, the system assumes the number of devices is 8. The system displays status information for primary paths only.

## **L = a, cc, or cca**

This operand specifies the display area (a), console (cc), or both (cca) where the display is to be presented.

## *Notes:*

1. *If you specify a device number that was not specified during system generation in the IODEVICE macro, the resultant display starts with the next higher device number that was specified.*
2. *For multiple-exposure devices, the value you specify for xxx must be the same as that specified in the ADDRESS parameter of the IODEVICE macro during system generation.*
3. *When the system displays the device type for an MCS console SYSGENed as a 3270 model X, it identifies it as a 3270 console.*

## DISPLAY Command

*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 250 or higher, enter:

```
D U,,ALLOC,250,8
```

or

```
D U,,ALLOC,250
```

## Displaying System Configuration Information

Use the DISPLAY M command to display the status of sides, processors, Vector Facilities, I/O engines (or EXDCs), channel paths, devices, and storage or to compare the current hardware configuration to the configuration in a CONFIGxx member of SYS1.PARMLIB.

|                              |   |   |
|------------------------------|---|---|
| <pre>{ DISPLAY } { D }</pre> | <pre>M =CHP [ (xx)       (xx-xx)       (list) ] =CONFIG [ (xx) ] ={ CPU   CPUAD } [ (x)             (list) ] ={ DEV   DEVICE } [ (xxx)             (xxx-xxx)             (list) ] =ESTOR [ (ddddM-ddddM)          (list)          (E[=id]) ] =HIGH =HSA =IO [ (id) ] =SIDE [ (id) ] ={ STOR   STORAGE } [ (ddddM-ddddM)               (list)               (E[=id]) ] =(parm[,parm]...)</pre> | <pre>[ ,L= { a         cc         cca } ]</pre> |
|------------------------------|---|---|

### 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, I/O engines, real storage, extended 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

## DISPLAY Command

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.

For description of the display format, see messages IEE097I and IEE174I in *System Messages*.

### 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. For description of the display format, see message IEE174I in *Message Library: System Messages*.

#### (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 member CONFIG $xx$  of SYS1.PARMLIB. If you omit *xx*, the system assumes that you mean CONFIG00. For a description of the display format, see message IEE097I in *System Messages*.

### CPU or CPUAD

The system is to display the online and offline status of one or more processors and Vector Facilities. For a description of the display format, see message IEE174I in *Message Library: System Messages*. If you do not specify any processor identifiers, the system displays the online and offline status of all processors and Vector Facilities attached to these processors.

#### (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.

### DEV or DEVICE

The system is to display the number of online channel paths to devices. Specify any hexadecimal number between 0 and FFF. If you do not specify any device, the system displays the status of all devices and the number of online channel paths, if any, to each device.

## DISPLAY Command

For a description of the display format, see message IEE174I in *Message Library: System Messages*.

**(xxx)**

A single device identified by *xxx*.

**(xxx-xxx)**

A range of devices.

**(list)**

One or more device numbers, or ranges of device numbers, or a combination of single device numbers and ranges of device numbers, each separated by a comma.

### ESTOR

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

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

If you do not specify any qualifiers, such as *(ddddM-ddddM)*, *(list)*, or *(E[=id])*, the system displays information for all installed extended storage. Use ESTOR only when your system has extended storage installed. For a description of the display format, see message IEE174I in *Message Library: System Messages*.

**(ddddM-ddddM)**

A range of addresses (in decimal) of extended storage. *ddddM* must be a multiple of 4 megabytes.

**(list)**

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

**(E[=id])**

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

### HIGH

The system is to display the highest possible real storage and extended 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 in *Message Library: System Messages*.

### 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 in *Message Library: System Messages*.



## IO[id]

The system is to display the channel paths that are on I/O engine (or EXDC) *id*, where *id* is the identifier (0 or 1) of the I/O engine. If the system is partitioned and the specified I/O engine is part of another configuration, no information is provided. If you do not specify an *id*, the system displays the information for all online and offline I/O engines. Use IO[*(id)*] only when your processor complex has more than one I/O engine.

For MVS/SP 2.1.7, the DISPLAY M=IO command causes the same display as the DISPLAY M=SIDE command. For a description of the display format, see message IEE174I in *Message Library: System Messages*.

## SIDE[id]

The system is to display the resources installed in side *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.

An example of the output of the DISPLAY M=SIDE command appears in Figure 3-6 on page 3-14. For a complete description of the display format, see message IEE174I in *Message Library: System Messages*.

## STOR or STORAGE

The system is to display the status of real 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

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 (*ddddM-ddddM*), (*list*), or (*E[=id]*), the system displays the status of all real storage.

For a description of the display format, see message IEE174I in *Message Library: System Messages*.

## (ddddM-ddddM)

The starting and ending addresses of the real storage for which you want the status display. Specify up to four decimal digits followed by M (megabytes) for each address. The value for each *dddd* must be a multiple of 4. (If the total amount of storage on a 3084 processor complex is greater than 32 megabytes, the value for *dddd* must be a multiple of 8.)

## (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

## DISPLAY Command

online, whether the storage element is online or offline and if offline, whether it is in a service, maintenance, repair (SMR) configuration. If you omit the *id*, the system displays this information for all installed storage elements.

*Notes:*

1. The *E=id* parameter is not supported on the 4381 processor.
2. If the processor complex is partitioned and the specified storage element is part of another configuration, no information is provided.

### (parm[,parm]...)

The system is to display the status of each resource you specify as *parm*. The list of *parms* you specify within the parentheses may contain any combination of CHP, CPU, DEV, HIGH, HSA, ESTOR, IO, 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, or cca

This operand specifies the display area (*a*), console (*cc*), or both (*cca*) where the display is to be presented.

*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 real storage address available
- The status of real storage

enter:

```
D M=(CPU,DEV,HIGH,STOR)
```

*Example 3:* To display the number of megabytes of storage the system owns in storage element 2 and the status of the storage element, enter:

```
D M=STOR(E=2)
```

*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, and 5, and the high storage addresses for real and extended storage, enter:

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

# DISPLAY Command

*Example 6:* To display the number of megabytes assigned and available to the extended storage ranges 4M through 8M and 12M through 16M, enter:

```
D M=ESTOR(4M-8M,12M-16M)
```

*Example 7:* To display the number of megabytes assigned and available to extended storage element 0 and the status of the element, enter:

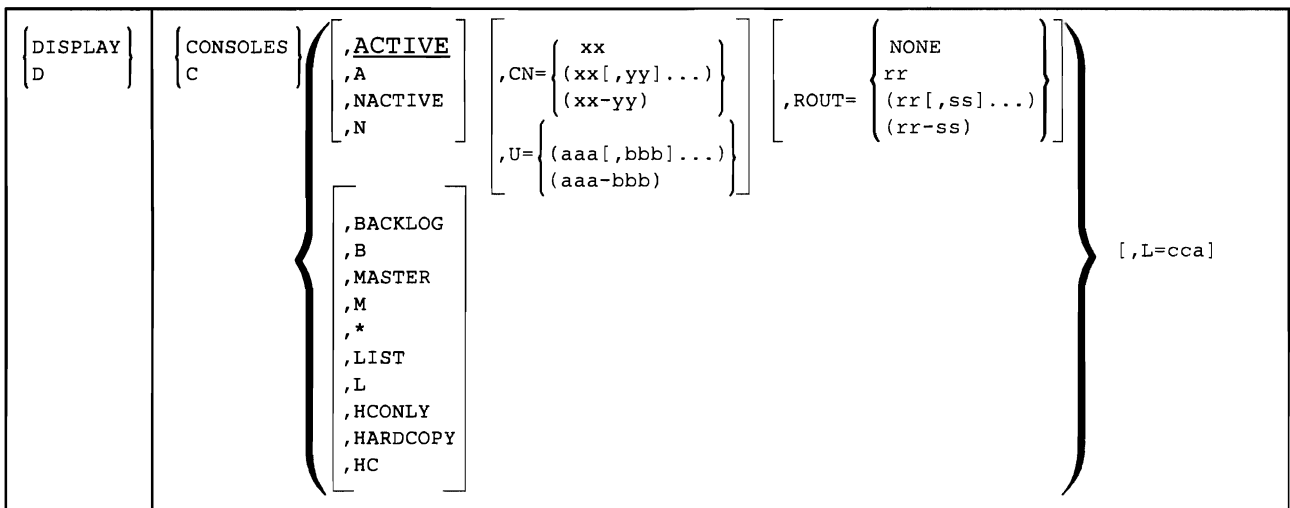
```
D M=ESTOR(E=0)
```

## Displaying Console Status Information

Use the DISPLAY C command to display the status of all consoles or specified consoles.

See Figure 3-6 on page 3-14 and Figure 2-13 on page 2-52 for examples of the resulting display of the DISPLAY CONSOLES and DISPLAY CONSOLES,BACKLOG commands.

The syntax of the command is:



### CONSOLES or C

Console information is to be displayed in message IEE249I. See *Message Library: System Messages*, for a description of the output.

### ACTIVE or A

The system is to display the status of all active consoles. **ACTIVE** is the default.

### NACTIVE or N

The system is to display the status of all consoles that are not active.

### SS

The system is to display the status of all subsystem-allocatable consoles.

### CN=

The system is to display the status of a console or consoles identified by console id.

## DISPLAY Command

**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

**(xx-yy)**

The system is to display the status of all consoles with ids in the range of *xx* to *yy*.

**U=**

The system is to display the status of the console or consoles, identified by device number.

**aaa**

The system is to display the status of the console with a device number of *aaa*.

**(aaa[,bbb]...)**

The system is to display the status of all consoles with device numbers listed as *aaa,bbb*, and so forth.

**(aaa-bbb)**

The system is to display the status of all consoles with device numbers in the range of *aaa* to *bbb*.

**ROUT =**

The system is to display the status of all consoles that receive messages identified by the routing code you specify.

**NONE**

The system is to display the status of all consoles to which no messages are routed by routing code.

**rr**

The system is to display the status of the console that accepts messages with a routing code of *rr*.

**(rr[,ss]...)**

The system is to display the status of all consoles that accept messages with routing codes listed as *rr,ss*, and so forth

**(rr-ss)**

The system is to display the status of all consoles that accept messages with routing codes in the range of *rr* to *ss*.

**ALL**

The system is to display the status of all consoles that accept messages by routing codes.

**BACKLOG or B**

The system is to display the status of all consoles with a message backlog. The information is listed in descending order by quantity of backlogged messages.

**MASTER or M**

The system is to display the status of the master console and pseudo-master consoles.

\*

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 all consoles defined to the system.

**HCONLY**

The system is to display information on messages that are not directed to any console (Those messages that appear only in the hardcopy log.)

**HARDCOPY or HC**

The system is to display the following information about the hardcopy log:

- Whether it is SYSLOG or a device
- Whether it is to receive operator commands, responses, and status displays
- The routing codes for messages the system is to send
- The number of messages waiting to be placed in it

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

This operand specifies the display area (a), console (cc), or both (cca) where the display is to be presented.

*Example 1:* To display information about all consoles with console IDs in the range of 5 to 20, enter:

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

*Example 2:* To display information about all active consoles that receive messages with routing code 3, enter:

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

*Example 3:* To display information about all subsystem consoles that receive routing code 15, enter:

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

*Example 4:* To display information about the console whose device number is 81B, enter:

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

*Example 5:* To display information about all inactive consoles, enter:

```
D C,N
```

*Example 6:* To display information about the hardcopy log on console 13, enter:

```
D C,HC,L=13
```

# DISPLAY Command

## Displaying CONTROL Command Functions

Use the DISPLAY C,K command to request a summary of the CONTROL command operands and the functions they perform.

|  |  |
|--|--|
| $\left\{ \begin{array}{l} \text{DISPLAY} \\ \text{D} \end{array} \right\}$ | $C, K \left[ , L = \left\{ \begin{array}{l} a \\ cc \\ cca \end{array} \right\} \right]$ |
|--|--|

### C,K

A summary of CONTROL command operands is to be displayed.

### L = a, cc, or cca

This operand specifies the display area (a), console (cc), or both (cca) where the display is to be presented.

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

D C,K,L=10A

## Displaying Dump Options or Dump Data Set Status

Use the DISPLAY DUMP command to determine the dump options and dump mode currently in effect for each dump type, or to determine which SYS1.DUMP data sets are full.

|  |   |  |   |
|--|---|--|---|
| $\left\{ \begin{array}{l} \text{DISPLAY} \\ \text{D} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{DUMP} \\ \text{D} \end{array} \right\} \left\{ \begin{array}{l} \left\{ \begin{array}{l} , \text{STATUS} \\ , \text{ST} \\ , \text{S} \end{array} \right\} \\ \left\{ \begin{array}{l} , \text{TITLE} \\ , \text{T} \end{array} \right\} \\ \left\{ \begin{array}{l} , \text{ERRDATA} \\ , \text{ER} \\ , \text{E} \end{array} \right\} \\ \left\{ \begin{array}{l} , \text{OPTIONS} \\ , \text{O} \end{array} \right\} \end{array} \right\}$ | $\left\{ \begin{array}{l} , \text{DSN} = \left\{ \begin{array}{l} nn \\ (nn[, nn] \dots) \\ nn-nn \\ (nn-nn[, nn-nn] \dots) \\ (nn[, nn] \dots, nn-nn[, nn-nn] \dots) \\ \underline{ALL} \\ (ALL) \end{array} \right\} \end{array} \right\}$ | $\left[ , L = \left\{ \begin{array}{l} a \\ cc \\ cca \end{array} \right\} \right]$ |
|--|---|--|---|

### DUMP or D

The system is to display dump information.

### **STATUS or ST or S**

The system is to display a summary of the full or available status of each defined SYS1.DUMP data set. The system displays this information in two sections: one for direct access data sets and the other for tape data sets.

### **TITLE or T**

The system is to display the dump title and time of dump for each full direct access dump data set that you specify on the DSN parameter. The system does not display this information for tape dump data sets.

### **ERRDATA or ER or E**

The system is to display error data for the full direct access dump data sets that you specify in DSN=. The error data for each full data set includes:

- Dump title
- 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 system does not display error data for tape dump data sets.

The SDWA furnishes most of the data that appears in the display caused by the DISPLAY DUMP,ERRDATA command. This means that, if MVS/XA is not in recovery mode, the display contains only the title of the dump.

## DISPLAY Command

### DSN=

The system is to display dump information about specific defined direct access dump data sets or about all direct access dump data sets. You can specify for DSN= one or more single data sets, one or more ranges of data sets, one or more single data sets and one or more ranges of data sets, or all data sets. For any of these specifications, *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*.

The system does not display dump information for tape dump data sets.

### OPTIONS or O

The system is to display the dump mode and dump options in effect for the SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP dump types.

### L=a, cc, or cca

This operand specifies the display area (a), console (cc), or both (cca) where the display is to be presented.

*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, 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
```



## Displaying System Activity

Use the following form of the DISPLAY command to display information about current system activity. The descriptions of messages IEE104I and IEE105I in *Message Library: System Messages* show the resulting display.

|  |  |   |   |
|--|--|---|---|
| $\left\{ \begin{array}{l} \text{DISPLAY} \\ \text{D} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{JOBS} \\ \text{J} \\ \text{TS} \\ \text{A} \end{array} \right\}$ | $\left[ \begin{array}{l} \text{,LIST} \\ \text{,L} \\ \text{,ALL} \\ \text{,A} \\ \left\{ \begin{array}{l} \text{,name} \\ \text{,(name)} \end{array} \right\} \end{array} \right]$ | $\left[ \text{,L=} \left\{ \begin{array}{l} \text{a} \\ \text{cc} \\ \text{cca} \end{array} \right\} \right]$ |
|--|--|---|---|

### JOBS or J, TS, or A

The system is to display the following overview information about system activity:

- Number of active batch jobs
- Number of started tasks (MOUNT commands in execution are treated as started tasks)
- Number of logged-on time-sharing users
- Number of logged-on time-sharing users running under TSO/VTAM
- The maximum number of time-sharing users allowed to be logged on under TSO/VTAM
- Number of active system address spaces
- Number of active initiators

### 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), or both (A).

### ALL or A

The system is to display more detailed information for active jobs and started tasks, logged-on time-sharing users, and active system address spaces than that supplied by LIST.

# DISPLAY Command

## name or (name)

The system is to display detailed information for active jobs and started tasks, logged-on time-sharing users, and active system address spaces with the specified name. If the specified name is the same as a valid secondary operand, you must enclose it in parentheses (See Example 7).

### Notes:

1. The specified name can be the name of a job, started task, time-sharing user, or system address space. For a job or started task, the name can be 1 to 8 alphameric or national characters or 1 to 7 alphameric or national characters followed by an asterisk. For a time-sharing user, the name can be 1 to 7 alphameric or national characters or 1 to 6 alphameric or national characters followed by an asterisk. For a system address space, the name must be the complete address space name. When you use the asterisk format, all jobs, started tasks, or time-sharing users with names that begin with the specified characters are displayed. See Example 5.
2. The only time you may use an asterisk in the first position of a name is to specify the master address space: \*MASTER\*.

## L = a, cc, or cca

This operand specifies the display area (a), console (cc), or both (cca) where the display is to be presented.

The system displays, below the overview information, detailed information when you specify LIST or L. Figure 4-6 shows the operands that you can combine with LIST or L and the detailed information that results from each combination.

| Primary Operand | Information for the LIST Operand   |
|-----------------|--|
| JOBS or J       | List of active jobs, including, for each job: <ul style="list-style-type: none"><li>● Jobname</li><li>● Stepname</li><li>● Procedure stepname</li><li>● Type of job</li><li>● Address space status</li><li>● Real address range (V = R only)</li></ul>         |
| TS              | List of logged-on time-sharing users, including, for each user: <ul style="list-style-type: none"><li>● Userid</li><li>● Address space status</li></ul>  |
| A               | List of all active units of work, including: <ul style="list-style-type: none"><li>● For active jobs and started tasks, all of the information listed for JOBS or J</li><li>● For logged-on time-sharing users, all of the information listed for TS</li></ul> |

Figure 4-6. Displaying System Activity: Information for the LIST Operand

The system displays, below the overview information, more detailed information when you specify ALL or A than when you specify LIST or L. Figure 4-7 shows the operands that you can combine with ALL or A and the detailed information that results from each combination.

| Primary Operand | Information for the ALL Operand   |
|-----------------|---|
| JOBS or J       | <p>List of active jobs, including, for each job:</p> <ul style="list-style-type: none"> <li>● Jobname</li> <li>● Stepname</li> <li>● Procedure stepname</li> <li>● Type of job</li> <li>● Address space identifier</li> <li>● Address space status</li> <li>● Program event recording (PER) activity</li> <li>● Number of outstanding step-must-complete requests</li> <li>● Performance group number</li> <li>● Domain number</li> <li>● Processor affinity</li> <li>● Elapsed time since initiation</li> <li>● Accumulated processor time</li> <li>● Real address range (V = R only)</li> </ul> |
| TS              | <p>List of logged-on time-sharing users, including, for each user:</p> <ul style="list-style-type: none"> <li>● Address space status</li> <li>● Address space identifier</li> <li>● Program event recording (PER) activity</li> <li>● Number of outstanding step-must-complete requests</li> <li>● Performance group number</li> <li>● Domain number</li> <li>● Processor affinity</li> <li>● Elapsed time since LOGON</li> <li>● Accumulated processor time</li> </ul>   |
| A               | <p>List of all active units of work, including:</p> <ul style="list-style-type: none"> <li>● For each active job and started task: <ul style="list-style-type: none"> <li>– All of the information listed for JOBS or J</li> </ul> </li> <li>● For each time-sharing user: <ul style="list-style-type: none"> <li>– All of the information listed for TS</li> </ul> </li> <li>● For each active system address space: <ul style="list-style-type: none"> <li>– Name</li> <li>– Stepname</li> <li>– Procedure stepname</li> </ul> </li> </ul>  |

**Figure 4-7. Displaying System Activity: Information for the ALL Operand**

*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\*

## DISPLAY Command

*Example 7:* To display detailed information about any active time-sharing user with the name LIST, enter:

```
D TS,(LIST)
```

### Displaying System Requests

Use the following form of the DISPLAY command to display outstanding messages requiring operator action. 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 under messages IEE112I and IEE312I in *Message Library: System Messages*.

|                          |   |
|--------------------------|---|
| <pre>{ DISPLAY   D</pre> | <pre>       ,U       ,KEY [,SYS=sysname] [,CN=(ALL)] [,L= { a                                      cc                                      cca } ] R { ,I    ,E    ,CE    ,R    ,M    ,LIST    ,L    ,ALL    ,A    ,    { ,T      ,S } [,MSG=msgid] [,SYS=sysname] [,KEY=keyname]      ,J      ,M    [ ,CN= { xx             (ALL) } ] [ ,ROUT= { ALL                                 (rxx[,sss]...)                                 (rxx-sss[,rxx-sss]...) } ] [ ,L= { a   cc   cca } ] </pre> |
|--------------------------|---|

*Notes:*

1. If you supply all commas between DISPLAY R and the operands that have equal signs, you get default values. When you specify the location operand, however, supply only one comma before the L, even if you omit preceding operands. For example, DISPLAY R,I,L=2B.
2. Because TSO consoles do not route messages by console ids or routing codes, do not use the ROUT= or CN= operands on DISPLAY R commands issued from TSO terminals in OPERATOR mode.

### 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. **Unless you specify CN, the system displays information about only those messages that appeared at the console that issues the DISPLAY R command.**

### I

The texts and message identification numbers of all outstanding immediate action messages (descriptor codes 1 or 2) are to be displayed.

### E

The texts and message identification numbers of all outstanding eventual action messages (descriptor codes 3) are to be displayed.

### CE

The texts and message identification numbers of all outstanding critical eventual action messages (descriptor code 11) are to be displayed.

### R

The texts and message identification numbers of all messages awaiting replies are to be displayed.

### M

The texts and message identification numbers of all immediate action, eventual action, and critical eventual action messages, and messages awaiting replies are to be displayed.

### LIST, L, ALL, A, or *no operand*

The texts and message identification numbers of all immediate action, eventual action, and critical eventual action messages and messages awaiting replies are to be displayed. The device numbers of devices with unfulfilled mount requests and any units requiring operator intervention are also displayed.

### U

The device numbers of devices with unfulfilled mount requests and any units requiring operator intervention are to be displayed.

### T

The time the message was issued and the job name and system name of the issuer are to be displayed with the message text.

### S

The system name and job name of the issuer are to be displayed with the message text.

### J

The job id or job name of the issuer is to be displayed with the message text.

## DISPLAY Command

### M

Only the message text is to be displayed. This is the default if T, S, or J is not specified.

### 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*.

### 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 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 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.

### 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*.

### (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.

**(nn[,nn]...)**

Requests messages with one or more routing codes.

**(nn-nn)**

Requests messages within a range of routing codes. When you specify a range of routing codes, the first *nn* in the range must be less than or equal to the second *nn*.

**NONE**

Requests only those messages that the system directs to the console by console id.

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

This operand specifies the display area (a), console (cc), or both (cca) where the display is to be presented.

*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
```

## DISPLAY Command

*Example 7:* To display the system names, job ids 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, enter from a console on the JES3 global:

```
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 CON2), excluding messages that were directed to the master console by default, enter:

```
DISPLAY R,L,CN=CON2
```

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 CON24 by any routing code, enter:

```
DISPLAY R,ROUT=ALL      issued from console CON24  
or  
DISPLAY R,ROUT=ALL,CN=CON24  issued from another console
```

## Displaying SMF Data

Use the DISPLAY SMF command to display SMF data.

|  |     |  |   |
|--|-----|--|---|
| $\left\{ \begin{array}{l} \text{DISPLAY} \\ \text{D} \end{array} \right\}$ | SMF | $\left[ \begin{array}{l} ,S \\ ,O \end{array} \right]$ | $\left[ ,L= \left\{ \begin{array}{l} a \\ cc \\ cca \end{array} \right\} \right]$ |
|--|-----|--|---|

### SMF

Indicates that the status of SMF data sets or the SMF options in effect are to be displayed (message IEE967I).

### S

Indicates that SMF data set names and their status are to be displayed. S is the default parameter.

### O

Indicates that current SMF options are to be displayed.

### L = a, cc, or cca

This operand specifies the display area (a), console (cc), or both (cca) where the display is to be presented.



## 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.

|  |   |  |
|--|---|--|
| $\left\{ \begin{array}{l} \text{DISPLAY} \\ \text{D} \end{array} \right\}$ | PFK $\left[ \begin{array}{l} \left[ \begin{array}{l} , \text{TABLE} \\ , \text{T} \end{array} \right] [= \text{nnnnnnnn}] \\ , \text{CN} = \text{cc} \end{array} \right]$ | $\left[ \begin{array}{l} , \text{L} = \left\{ \begin{array}{l} \text{a} \\ \text{cc} \\ \text{cca} \end{array} \right\} \end{array} \right]$ |
|--|---|--|

### 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.

### 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, or cca

This operand specifies the display area (a), console (cc), or both (cca) where the display is to be presented.

*Example 1:* To display PFK definitions for the console on which you issue the command, enter:

```
D PFK
```

*Example 2:* To display all available PFK tables, enter:

```
D PFK,T
```

*Example 3:* To display the contents (PFK definitions) of the PFK table named MVSCMDS, enter:

```
D PFK,T=MVSCMDS
```

## DISPLAY Command

### Displaying the Local Time and Date and the 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.

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{DISPLAY} \\ \text{D} \end{array} \right\}$ | 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 SLIP Trap Information

Use the DISPLAY SLIP command to display information about SLIP traps.

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{DISPLAY} \\ \text{D} \end{array} \right\}$ | SLIP [=xxxx] $\left[ ,L= \left\{ \begin{array}{l} \text{a} \\ \text{cc} \\ \text{cca} \end{array} \right\} \right]$ |
|--|---|

**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 less 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. The SLIP MOD command allows you to enable and disable them as a group.

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

This operand specifies the display area (a), console (cc), or both (cca) where the display is to be presented.

*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 the Domain Description Table

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.

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{DISPLAY} \\ \text{D} \end{array} \right\}$ | $\text{DMN} [= \text{domainnum}] \left[ , \text{L} = \left\{ \begin{array}{l} \text{a} \\ \text{cc} \\ \text{cca} \end{array} \right\} \right]$ |
|--|---|

### 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, or cca

This operand specifies the display area (a), console (cc), or both (cca) where the display is to be presented.

*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
```

For each domain the following information is displayed:

- Domain number
- Minimum and maximum multiprogramming levels
- Current and target multiprogramming levels
- Average number of ready users
- Number of swappable users that are currently in storage
- Number of non-swappable users
- Number of users being swapped-out
- Time-weighted service rate
- Contention index
- Method of domain multiprogramming level target control
- Value for multiprogramming level target control
- Number of ready users that are currently swapped out of storage

# DISPLAY Command

## Displaying Message Suppression, Retention, Color, Intensity, and Highlighting Options

Use the DISPLAY MPF command to display information about message processing and presentation that is contained in the MPFLSTxx member of SYS1.PARMLIB currently in effect.

The MPF parameter on the INIT statement in the CONSOLxx member of SYS1.PARMLIB or the SET MPF=xx command activates and deactivates an MPFLSTxx member.

The syntax of the DISPLAY MPF command is:

|  |     |  |  |
|--|-----|--|--|
| $\left\{ \begin{array}{l} \text{DISPLAY} \\ \text{D} \end{array} \right\}$ | MPF | $\left[ \begin{array}{l} ,\text{MSG} \\ ,\text{M} \\ ,\text{COLOR} \\ ,\text{C} \end{array} \right]$ | $\left[ ,\text{L}=\left\{ \begin{array}{l} \text{a} \\ \text{cc} \\ \text{cca} \end{array} \right\} \right]$ |
|--|-----|--|--|

### 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 messages are **not** being retained by the action message retention facility
- Which user exits receive control for selected messages
- The status of the general WTO user exit IEAVMXIT
- What color, intensity, and highlighting capabilities are in effect

See message IEE189I in *Message Library: System Messages* for a description of the output.

### 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 user exits receive control for selected messages
- The status of the general WTO user exit IEAVMXIT

## COLOR or C

The system is to display the color, intensity, and highlighting options in effect.

## L = a, cc, or cca

This operand specifies the display area (a), console (cc), or both (cca) where the display is to be presented.

*Example 1:* To display information about the message IDs and WTOR user 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 on console 2 the color specifications, message processing information, and retention information for all consoles, enter:

```
D MPF ,L=2
```

## 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 and information about CTC adapters assigned to global resource serialization and attached to the system on which the command is issued.

- 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

If you do not code any keywords on the DISPLAY GRS command, the display includes configuration information.

## DISPLAY Command

The complete syntax for the DISPLAY GRS command is:

|                          |  |                                 |   |   |
|--------------------------|--|---------------------------------|---|---|
| <pre>{ DISPLAY D }</pre> | <pre>GRS [ ,SYSTEM       ,LINK       ,ALL       ,A ]</pre> | <pre>[ ,CONTENTION   ,C ]</pre> | <pre>,RNL= { CONVERSION         CON         C         EXCLUSION         EXCL         E         INCLUSION         INCL         I         ALL         A }</pre> | <pre>[ ,RES= ( { qname            *            } [ , { rname                  *                  } ] ) ] [ ,HEX ]</pre> |
|--------------------------|--|---------------------------------|---|---|

### 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 complex:
  - Active: The system is presently serializing global resources. It is a member of the global resource serialization ring.
  - Inactive: This system is not presently serializing global resources, but it can be used to restart the global resource serialization ring. The system suspends any of its tasks that try to obtain or release global resources.
  - Quiesced: This system is not presently serializing global resources, but it can be used to restart this system but not the global resource serialization ring. The system suspends any of its tasks that try to obtain or release global resources.
  - 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.
- The communication status of each system in the global resource serialization complex; that is, whether or not there is a functioning CTC adapter between this system and the system whose information is being displayed.

### LINK

CTC adapter information is to be displayed. The LINK operand produces a display only when a global resource serialization complex is active.

The display of CTC adapter information includes:

- The device number of each CTC adapter assigned to global resource serialization on this system
- The status of each CTC adapter, which can be any one of the following:
  - In-use: This system is using this CTC adapter to send messages to and receive messages from the system at the other end of the CTC adapter.
  - Alternate: This system is not presently using this CTC adapter to send or receive messages, but it has the ability to do so.
  - Disabled: This system cannot use this CTC adapter to send or receive messages.
  - Quiet: The system at the opposite end of the CTC adapter 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 CTC adapter.

### ALL or A

System information, CTC adapter information, resource contention information, and the contents of all RNLs for the current global resource serialization are to be displayed. Specify HEX if you want the resource names located in the RNLs displayed in EBCDIC and in hexadecimal. If the global resource serialization complex is inactive, the display does not contain system information, CTC adapter information, or contents of RNLs.

### CONTENTION or C

Resource contention information for the current global resource serialization complex is to be displayed. Specify HEX if you want resource names displayed in EBCDIC and in hexadecimal.

### RNL =

The contents of one or all 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.

### CONVERSION, CON, or C

The contents of the RESERVE conversion RNL are to be displayed.

### EXCLUSION, EXCL, or E

The contents of the SYSTEMS exclusion RNL are to be displayed.

### INCLUSION, 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.

## DISPLAY Command

### 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 Character Set Reference*.)

*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 characters that are alphanumeric (A-Z and 0-9), national (#, @, and \$), and/or 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 alphameric (A-Z and 0-9), national (#, @, and \$), and/or a period (.), specify either:

- 1-52 alphameric characters (a specific minor name)
- 1-51 alphameric 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.

*Example 1:* To display resource information about all resources that have requestors, enter:

```
D GRS,RES=(*,*)
```

*Note:* Because this command gives you data about every allocated ENQ/RESERVE resource, there may be a very large display.

## DISPLAY Command

*Example 2:* To display resource information about all resources whose major name is SYSDSN, enter:

```
D GRS,RES=(SYSDSN,*)
```

*Example 3:* CVOLs (non-VSAM user catalogs) are protected by the following RESERVE macro instruction:

```
RESERVE (SYSCTLG,vvvvvv.UCBADDR,E,rname-length
```

where:

SYSCTLG is the qname

vvvvvv.UCBADDR is the rname

vvvvvv - the volume serial of the DASD containing the CVOL

UCBADDR - the address of the UCB representing the DASD

(This address is in hexadecimal.)

E requests exclusive ownership of the resource

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  
EEEEEDC  
2823337
```

## DUMP Command

MVS/XA provides two types of dumps: system dumps and user dumps. System dumps (SVC dumps and SDUMPs) are taken, as required, by system recovery routines. User dumps are taken by the system during error termination processing of a user task or in response to a dump request made by a user with a system macro instruction. The DUMP command causes a system dump of virtual storage known as an SVC dump. The SVC dump is stored in a SYS1.DUMP data set, which can be either a tape device or a direct access data set.

You can define SYS1.DUMP data sets both during and after system initialization. During system initialization, you define SYS1.DUMP data sets in the DUMP system parameter. After system initialization, you define SYS1.DUMP data sets with the DUMPDS command. For more information about the DUMP system parameter, see *Initialization and Tuning*. For more information about the DUMPDS command, see the DUMPDS command.

The complete syntax for the DUMP command is:

|      |             |
|------|-------------|
| DUMP | COMM=(text) |
|------|-------------|

### COMM=(text)

The title (1-100 characters) you want the dump to have. This title becomes the first record in the dump data set.

In response to the DUMP command, the system prompts you for the dump options you want to specify with the following message:

```
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
```

## Specifying Dump Options

You must use the REPLY command to respond to message IEE094D. The REPLY command syntax for specifying dump options is:

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{REPLY} \\ \text{R} \end{array} \right\}$ | $\text{id} \left\{ \begin{array}{l} ,U \\ \left[ \begin{array}{l} ,ASID=(n[,n]...) \\ ,JOBNAME=(name[,name]...) \\ ,TSONAME=(name[,name]...) \end{array} \right] \left[ ,SDATA=[(option[,option]...)] \right] \left[ ,STOR=(beg,end[,beg,end]...) \right] \left[ \begin{array}{l} ,CONT \\ ,END \end{array} \right] \end{array} \right\}$ <p>Note: CONT or END, when specified, must be the last parameter on the input line.</p> |
|--|---|

### 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.

## DUMP Command

### ASID = (n[,n]...)

*n* is the hexadecimal address space identifier of an address space you want to dump. You can specify up to 15 address space identifiers. If you specify only one identifier, you do not need to enter the parentheses.

### JOBNAME = (name[,name]...)

*name* is the jobname of a background job whose address space you want to dump. You can specify a maximum of 15 jobnames (up to a total of 120 characters, including the keyword JOBNAME). If you specify only one name, you do not need to enter the parentheses.

### TSOname = (name[,name]...)

*name* is the user identifier of a TSO user whose address space you want to dump. You can specify a maximum of 15 names (up to a total of 120 characters, including the keyword TSOname). If you specify only one name, you do not need to enter the parentheses.

### 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 the system significantly. Dump only the address spaces that are likely to be involved in a problem. For example, if a TSO terminal is having a problem, dump only the TSO user's address space and the TCAM or VTAM address space.

### 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.

**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.

**NOPSA** — no prefixed storage area.

**RGN** — private area of address space being dumped, including LSQA and SWA.

**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 hardcopy buffer data.

If you do not specify SDATA or if you specify SDATA without any options, the system assumes the underlined options.

**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.

**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. The only parameters that you can specify on a continuation line are SDATA, STOR, CONT, and END.

**END**

Identifies the end of your reply. You need to specify END only when the reply line contains no other parameters. If you specify no parameters on the same line as the END parameter and the previous line contains no parameters other than CONT, the system dumps the master address space with the SDATA default options.

*Note:* You should make only one dump request at a time. Otherwise, you might have trouble determining the dump request that causes a particular appearance of message IEE094D. Also, the system takes only one SDUMP at a time, so it does not save anything to make several requests at once.

*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 don't want a summary dump and are sure of the address spaces 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),SDATA=(NOSUM),CONT
```

Because you specified CONT, the system reissues:

```
* id IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
```

## 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
```

## DUMPDS Command

Use the DUMPDS command to:

- Change the system's list of SYS1.DUMP data sets
- Clear full SYS1.DUMP data sets and make them available for dumps

Notes:

1. *SYS1.DUMP data sets are where SVC dump sends its dumps. A SYS1.DUMP data set can be either a direct access data set or a physical tape device. Each direct access SYS1.DUMP data set has a name of the form SYS1.DUMPnn, where nn can be any decimal number from 00-99.*
2. *The combined number of direct access data sets and tape units on the system's list of SYS1.DUMP data sets can not be more than 100 at any time.*
3. *You must issue the DUMPDS command from a console with SYSTEM authority.*
4. *A DUMPDS DEL or DUMPDS CLEAR command has no effect on any data set that is receiving an SVC dump when you issue the command.*

The complete syntax for the DUMPDS command is:

|  |  |  |
|--|--|--|
| $\left\{ \begin{array}{l} \text{DUMPDS} \\ \text{DD} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{ADD} \\ \text{DEL} \\ \text{CLEAR} \end{array} \right\}$ | $\left( \begin{array}{l} \text{,DSN=} \left\{ \begin{array}{l} \text{nn} \\ \text{(nn[,nn]...)} \\ \text{nn-nn} \\ \text{(nn-nn[,nn-nn]...)} \\ \text{(nn[,nn]... ,nn-nn[,nn-nn]...)} \\ \text{ALL} \\ \text{(ALL)} \end{array} \right\} \\ \text{,UNIT=devnum} \end{array} \right)$ |
|--|--|--|

### ADD,DSN=

The system is to add the specified direct access data sets to its list of SYS1.DUMP data sets. Before using ADD,DSN=, allocate, catalog, and protect the direct access data sets you specify.

*Note:* You don't have to protect the data sets, but if your installation does so, use the password or RACF procedures you normally use.

If any direct access data set you specify is empty, 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.

How you clear a full data set depends on whether or not you want to keep the contents of the data set. If you want to keep the contents of the data set, process the data set with the print dump service aid, AMDPRDMP. The print dump service aid clears the data set and marks it as available for a dump after it formats and prints the data set contents. If you don't want to keep the contents of the data set, issue DUMPDS CLEAR,DSN= for the data set. This command clears the data set and marks it as available for a dump without doing anything to save the data set contents.

## DUMPDS Command

A DUMPDS ADD,DSN= command does not process any direct access SYS1.DUMP data set that:

- Is already on the system's list of SYS1.DUMP data sets
- If added, would put over 100 the number of direct access data sets and tape units 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. For example, if you want the system to add the direct access data set SYS1.DUMP03 to its list of SYS1.DUMP data sets, specify DSN=03. You can specify one or more single identifiers, one or more ranges of identifiers, or one or more single identifiers and 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 add to its list of SYS1.DUMP data sets all the cataloged direct access SYS1.DUMP data sets that are not already on the list.

**ADD,UNIT = devnum**

The system is to add the specified tape unit to its list of SYS1.DUMP data sets. In response to this command, the system allocates the specified unit for dumps and requests that you mount an unlabeled tape on the unit. As soon as you mount this tape, the system marks the tape unit as available to receive a dump.

A DUMPDS ADD,UNIT = devnum command does not process any tape unit that:

- Is already on the system's list of SYS1.DUMP data sets
- If added, would put over 100 the number of direct access data sets and tape units on the system's list of SYS1.DUMP data sets
- Is managed by JES3

**devnum**

The three-digit hexadecimal device number (000-FFF) of a tape unit you want the system to add to its list of SYS1.DUMP data sets.

**DEL,DSN =**

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 direct access SYS1.DUMP data set you want the system to remove from its list of SYS1.DUMP data sets. For example, if you want the system to remove the direct access data set SYS1.DUMP03 from its list of SYS1.DUMP data sets, specify DSN=03. You can specify one or more single identifiers, one or more ranges of identifiers, or one or more single identifiers



## DUMPDS Command

and 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 direct access dump data sets from its list of SYS1.DUMP data sets.

### **DEL,UNIT = devnum**

The system is to remove the specified tape unit from its list of SYS1.DUMP data sets. In response to this command, the system unallocates the specified unit and unloads the tape on the unit.

A DUMPDS DEL,UNIT = devnum command does not process any tape unit that is not on the system's list of SYS1.DUMP data sets.

### **devnum**

The three-digit hexadecimal device number (000-FFF) of a tape unit you want the system to remove from its list of SYS1.DUMP data sets.

### **CLEAR,DSN =**

The system is to clear and mark as available for dumps the specified direct access dump data sets on its 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 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 clear and mark as available for a dump. For example, if you want the system to clear and mark as available for a dump the direct access data set SYS1.DUMP03 on the system's list of SYS1.DUMP data sets, specify DSN=03. You can specify one or more single identifiers, one or more ranges of identifiers, or one or more single identifiers and 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 clear and mark as available all full direct access dump data sets on its list of SYS1.DUMP data sets.

### **CLEAR,UNIT = devnum**

The system is to rewind and prepare for dumps the specified tape unit on its list of SYS1.DUMP data sets. The system clears the tape dump data set by going to the first file on the tape, writing an end-of-file mark, and going back to the first file.

A DUMPDS CLEAR,UNIT = devnum command does not process any tape unit that is not on the system's list of SYS1.DUMP data sets.

### **devnum**

The three-digit hexadecimal device number (000-FFF) of a tape unit you want the system to rewind and prepare for dumps.

## DUMPDS Command

*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 allocated and cataloged direct access data sets SYS1.DUMP00-SYS1.DUMP05, SYS1.DUMP08, and SYS1.DUMP10-SYS1.DUMP12 to the system's list of SYS1.DUMP data sets, 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 put tape unit 649 on the system's list of SYS1.DUMP data sets, enter:

```
dumpds add,unit=649
```

In response to this command, the system allocates tape unit 649 for dumps and requests that you mount an unlabeled tape on the unit.

*Example 5:* To remove SYS1.DUMP02 from the system's list of SYS1.DUMP data sets, enter:

```
dd del,dsn=02
```

*Example 6:* To delete all direct access data sets from the system's list of SYS1.DUMP data sets, enter:

```
dd del,dsn=all
```

*Example 7:* To remove tape unit 580 from the system's list of SYS1.DUMP data sets, enter:

```
DD del,unit=580
```

In response to this command, the system deallocates tape unit 580 and rewinds and unloads the tape on the unit.

*Example 8:* 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 9:* 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)
```

*Example 10:* To rewind the tape on unit 710 and prepare the tape to receive dumps, enter:

```
dumpds clear,unit=710
```

## 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. FORCE is not a substitute for CANCEL. Unless you issue CANCEL first for a cancellable job, the system issues error message IEE838I.

**CAUTION: 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 affected address space is deleted from the system and recovery is severely limited.
- If you need a dump, you must issue the DUMP command before you issue FORCE. Depending on the condition of the address space, a dump might not be possible.
- If your system was part of a global resource serialization ring (GRS=START or GRS=JOIN 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 DISPLAY GRS to determine GRS status.

You can enter FORCE only from the master console. With FORCE, you can immediately terminate:

- A job in execution
- A time-sharing user
- A cataloged procedure started by a START command
- A MOUNT command
- An external writer allocation
- The output processing for a job

The complete syntax for the FORCE command is:

|       |   |                  |
|-------|---|------------------|
| FORCE | $\left\{ \begin{array}{l} \{ \text{jobname} \} \\ \text{STARTING} \\ \{ \text{U=userid} \\ \text{U=*LOGON*} \} \\ \text{devnum} \\ \text{devicetype} \\ \text{identifier} \end{array} \right\}$ | [,ARM] [,A=asid] |
|-------|---|------------------|

### jobname or STARTING

The name of the job you want to terminate. Usually, you specify, for *jobname*, the unique name of the job in the system. However, if the job is just starting and does not yet have a unique name, or if other jobs in the system have the same name as the job you want to terminate, you must specify the job name in special ways.

If the job is just starting and does not yet have a name in the system, enter a DISPLAY JOBS,ALL command and note the address space identifiers of those tasks listed as

## FORCE Command

“STARTING.” Then, to terminate the starting job, enter `FORCE STARTING,A=asid`, specifying for *asid* the address space identifier of the job you want to terminate.

When two or more jobs or started tasks are running with the jobname you specify, the system rejects the command unless you specify `A=asid` along with the jobname, where *asid* is the address space identifier of the job you want to terminate. If the system rejects the command, reenter it with `A=asid`.

*Note:* When you terminate a job in execution with `FORCE`, 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.

### **U=userid or U=\*LOGON\***

The identifier of the time-sharing user you want to terminate. Usually, you specify, for *userid*, the unique identifier of the user. If, however, the user is just logging on and does not yet have a unique name, you must specify the identifier in a special way.

Enter `DISPLAY TS,ALL` to display the identifiers of the logged-on time-sharing users. If the user you want to terminate has a unique identifier in the system, enter `FORCE U=userid`, where *userid* is the unique identifier. If, however, the user is just logging on and does not yet have a unique name, note the address space identifiers of those tasks listed as “\*LOGON\*.” Then, to terminate the user, enter `FORCE U=*LOGON*,A=asid`, specifying for *asid* the address space identifier of the user you want to terminate.

### **devnum**

The device number specified when the `MOUNT` command that you want to terminate was issued.

### **devicetype**

The type of device specified when the `MOUNT` command that you want to terminate was issued.

### **identifier**

The identifier can be:

- The identifier specified on a `START` command for an external writer
- The device type (for example, 3410) or device number specified on a `START` command for an external writer
- The device type associated with the cataloged procedure specified on a `START` command for an external writer

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.

### ARM

The system is to terminate the specified job, time-sharing user, or started procedure if it is non-cancellable. 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 fails to terminate the address space within a reasonable time, reissue FORCE with the ARM parameter. If the command still fails after several attempts, try issuing FORCE without the ARM parameter.

### A = asid

The address space identifier, in hexadecimal, of the job, time-sharing user, or started procedure you want to terminate.

*Example 1:* To terminate an earlier MOUNT command for a 3330 device, enter:

```
FORCE 3330
```

*Example 2:* To remove job JOBXYZ from the system, enter:

```
FORCE JOBXYZ
```

*Example 3:* To stop device allocation for writer 1AF and terminate the writer itself, enter, during device allocation for writer 1AF:

```
FORCE 1AF
```

*Example 4:* To stop the output processing on writer 00E and terminate the writer itself, enter, during output processing on writer 00E:

```
FORCE 00E
```

*Example 5:* To log user A237 off the system, enter:

```
FORCE U=A237
```

*Example 6:* To terminate the non-cancellable job BIGTASK, enter:

```
FORCE bigtask,ARM
```

*Example 7:* To terminate the non-cancellable job SERVICE with the address space identifier of 1A8, enter:

```
FORCE service,arm,a=1a8
```

## HALT Command

## 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.

The complete syntax for the HALT command is:

|   |     |
|---|-----|
| $\left\{ \begin{array}{l} \text{HALT} \\ \text{Z} \end{array} \right\}$ | EOD |
|---|-----|

The HALT EOD command causes the system to take the following steps:

- Store the internal I/O device error counts in the SYS1.LOGREC data set.
- Empty the SMF buffers onto the active SMF data set in SYS1.MANx.
- 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.
```

## LOG Command

Use the LOG command to make an entry into the system log.

The complete syntax for the LOG command is:

|  |        |
|--|--------|
| $\left\{ \begin{array}{l} \text{LOG} \\ \text{L} \end{array} \right\}$ | 'text' |
|--|--------|

### 'text'

The entry (up to 122 characters) to be made in the system log. The system sends the entry to the master console if the system log is temporarily inactive; enter the command again when the log becomes active.

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, enter:

```
L 'DEVICE 235 OFFLINE FOR REPAIRS'
```

## MODE Command

## 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.
- The recording of system recovery or degradation machine check interruptions on the SYS1.LOGREC data set. For the procedure to print the SYS1.LOGREC data set, see the *EREP User's Guide*.
- The monitoring of hard machine checks, including machine checks that indicate timing facility damage.

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 defaulted to). 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, only the last (most recent) takes effect.

The complete syntax for the MODE command is:

|      |   |
|------|---|
| MODE | <p><u>STATUS</u></p> <p>{ SR } [ , QUIET [ =nnn ] [ , CPU={ x } ] ]</p> <p>{ DG } [ , RECORD [ =ALL [ , REPORT=nnn ] ] ] [ , CPU={ ALL } ]</p> <p>[ =4 ]</p> <p>{ PD } [ , INTERVAL={ nnnnn } ] [ , RECORD [ =nnn ] ] [ , CPU={ x } ]</p> <p>{ SD } [ =ALL ] [ , CPU={ ALL } ]</p> <p>{ IV } [ =5 ]</p> <p>{ TC }</p> <p>{ PT }</p> <p>{ CC }</p> <p>{ VS }</p> |
|------|---|



## MODE Command

Figure 4-8 shows the machine check interruption types you can specify and the parameters allowed for each type:

| Machine Check Interruption Type    | Command Parameters |            |                   |            |            |
|------------------------------------|--------------------|------------|-------------------|------------|------------|
|                                    | CPU =              | QUIET      | RECORD =          | REPORT =   | INTERVAL = |
| DG - degradation                   | X                  | X (note 1) | X (notes 1 and 2) | X (note 2) |            |
| SR - system recovery               | X                  | X (note 1) | X (notes 1 and 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) |

Note 1 - QUIET and RECORD = are mutually exclusive.

Note 2 - REPORT = can only be used with RECORD = ALL.

Note 3 - RECORD = ALL and INTERVAL are mutually exclusive.

**Figure 4-8. MODE Parameters Allowed for Machine Check Interruptions**

When you specify more than one option, the parameters can be entered in any order and must be separated by commas.

### Displaying Recording and Monitoring Status

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. STATUS is the default; if this parameter is specified, it must be the only parameter specified.

### Controlling the Recording of System Recovery and Degradation Machine Check Interruptions

Use the following form of the MODE command to control the recording of system recovery and degradation machine check interruptions.

|      |  |  |   |
|------|--|--|---|
| MODE | $\left\{ \begin{array}{l} \text{SR} \\ \text{DG} \end{array} \right\}$ | $\left[ \begin{array}{l} \text{,QUIET} \\ \text{,RECORD} \end{array} \left[ \begin{array}{l} =\text{nnn} \\ =\text{ALL} [ \text{,REPORT}=\text{nnn} ] \\ =\underline{4} \end{array} \right] \right]$ | $\left[ \text{,CPU}=\left\{ \begin{array}{l} \text{x} \\ \underline{\text{ALL}} \end{array} \right\} \right]$ |
|------|--|--|---|

## MODE Command

### 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 no number is specified or the RECORD parameter is omitted, RECORD = 4 is assumed.

### RECORD = ALL

All system recovery or degradation machine check interruptions occurring on the specified processor are to be recorded on the SYS1.LOGREC data set. You are notified each time a defined number of interruptions occur, but the system does not switch to QUIET mode for that type of interruption. The default number of interruptions is 4 unless changed by using the REPORT = nnn parameter along with the RECORD = ALL parameter.

### 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. This parameter can only be used in conjunction with the RECORD = ALL parameter.

### CPU = x

The address (0, 1, 2, or 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 (4) 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
```

## Controlling the Recording of Hard Machine Check Interruptions

Use the following form of the MODE command to control the recording or monitoring of hard machine check interruptions.

|      |   |  |   |  |
|------|---|--|---|--|
| MODE | $\left\{ \begin{array}{l} \text{PD} \\ \text{SD} \\ \text{IV} \\ \text{TC} \\ \text{PT} \\ \text{CC} \\ \text{VS} \end{array} \right\}$ | $\left[ , \text{INTERVAL} = \left\{ \begin{array}{l} \text{nnnnn} \\ \underline{300} \end{array} \right\} \right]$ | $\left[ , \text{RECORD} \left[ \begin{array}{l} =\text{nnn} \\ =\text{ALL} \\ =\underline{5} \end{array} \right] \right]$ | $\left[ , \text{CPU} = \left\{ \begin{array}{l} \text{x} \\ \underline{\text{ALL}} \end{array} \right\} \right]$ |
|------|---|--|---|--|

### 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.

### 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.

## MODE Command

### **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 SYS1.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, RECORD = 5 is assumed.

### **RECORD = ALL**

All specified hard machine check interruptions of the specified type occurring on the specified processor are to be recorded on the SYS1.LOGREC data set. The system no longer monitors the frequency of hard machine check interruptions of that type occurring on that processor.

### **CPU = x**

The address (0, 1, 2, or 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:* Instruction processing damage machine check interruptions are to be monitored on processor 0. If seven of these interruptions occur in 600 seconds on processor 0, the system invokes ACR to take processor 0 offline.

```
mode pd,record=7,interval=600,cpu=0
```

*Example 2:* All machine check interruptions indicating invalid PSW or registers are to be recorded on the SYS1.LOGREC data set but are not monitored for any processor in the system.

```
MODE IV,CPU=ALL,RECORD=ALL
```

*Example 3:* The frequency of system damage machine check interruptions is to be monitored 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), the system invokes ACR to take that processor offline.

```
mode sd
```

*Example 4:* The frequency of Vector Facility machine check interruptions is to be monitored on processor 1. If one of these interruptions occurs in 300 seconds, the processor disconnects the Vector Facility and issues message IGF970E.

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

**MODIFY Command****MODIFY Command**

Using the MODIFY system command, you can change the existing parameters of a job, start or stop TSO/TCAM, change the status of TSO/VTAM, change the status of an external writer, build a new LNKLST directory, display information about the catalog address space, or request the catalog address space to perform a specified service. The changes take effect immediately.

The complete syntax for the MODIFY command is:

|                             |  |
|-----------------------------|--|
| <pre>{   MODIFY   F }</pre> | <pre>jobname,parameters {   [procname.]identifier {     ,TS= { START } [ ,member ]         { STOP }     ,USERMAX=nnnnn     ,USER= { SIC }            { FSTOP }     ,CLASS= [classes]     ,JOBID= [jobid]     ,WRITER= { STDWTR               [user-writer-name]             }     ,FORMS= [forms-name]     ,DEST=  [ LOCAL               [remote-workstation-name]             ]     ,PAUSE= { FORMS               [DATASET]             }   }   LLA,REFRESH   CATALOG {     ,LIST[(xxxxxxxx)]     ,OPEN     ,CLOSE(catname)     ,VCLOSE(volser)     ,REPORT     ,ABEND(yyyyyyyy)     ,RESTART     ,ENTRY[(cname)]     ,END(ID) { ,REDRIVE               [ ,NOREDRIIVE ]             }   } }</pre> |
|-----------------------------|--|

**Modifying Job Parameters**

Use the MODIFY command to change the characteristics of a job. Note that you can modify the specified parameters only if the programmer has set the proper indicators and has given you the new parameters.

|                             |                               |
|-----------------------------|-------------------------------|
| <pre>{   MODIFY   F }</pre> | <pre>jobname,parameters</pre> |
|-----------------------------|-------------------------------|

**jobname**

The jobname used on the job card.

**parameters**

Parameters specified by a programmer that change the corresponding parameters in a job currently being processed.

## MODIFY Command

*Example:* To change the parameters in jobname TMASGX02 by specifying 00 as indicated by the programmer, enter:

```
f tmasgx02,00
```

### Starting TSO/TCAM Time-Sharing

Use the following form of the MODIFY command to start TSO/TCAM time-sharing once TCAM is active.

|                       |   |
|-----------------------|---|
| {<br>MODIFY<br>F<br>} | [procname.]identifier,TS=START[,member] |
|-----------------------|---|

**procname**

The name of the cataloged TCAM procedure in SYS1.PROCLIB used to start TCAM.

**identifier**

The user-determined identifier used on the START command to identify TCAM. If you do not specify an identifier on the START command, the system automatically uses the procname you supplied in the START command as the default identifier.

**TS=START**

Initiate TSO/TCAM time-sharing.

**member**

The SYS1.PARMLIB member containing TSO/TCAM time-sharing system parameters.

*Example:* To change the TCAM procedure TCAM2 used to start TSO/TCAM time-sharing by using the time-sharing system parameters in SYS1.PARMLIB member IKJPRM02 rather than the default parameters, enter:

```
s tcam2
f tcam2,ts=start,ikjprm02
```

### Stopping TSO/TCAM Time-Sharing

Use the following form of the MODIFY command to stop TSO/TCAM time-sharing.

|                       |                               |
|-----------------------|-------------------------------|
| {<br>MODIFY<br>F<br>} | [procname.]identifier,TS=STOP |
|-----------------------|-------------------------------|

**procname**

The cataloged TCAM procedure in SYS1.PROCLIB used to start TCAM.

**identifier**

The user-determined identifier used on the START command to identify TCAM. If you did not specify an identifier on the START command, the system automatically uses the procname you supplied in the START command as the default identifier.

## MODIFY Command

### 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 to allow terminal 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

Use the following form of the MODIFY command to control the number of users allowed to be logged on to TSO/VTAM and to terminate TSO user address spaces.

|                                     |  |
|-------------------------------------|--|
| <pre>{<br/>MODIFY<br/>F<br/>}</pre> | <pre>[procname.]identifier {<br/>,USERMAX=nnnnn<br/>,USER= {<br/>SIC<br/>FSTOP<br/>}<br/>}</pre> |
|-------------------------------------|--|

#### procname

The name of the cataloged procedure that was used in the START command to start TSO/VTAM time-sharing.

#### identifier

The identifier that was specified in the START command that started TSO/VTAM time-sharing. If you did not specify an identifier on the START command, the system automatically uses the procname you supplied in the START command as the default identifier.

#### 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.

# MODIFY Command

## Specifying Data Set Selection Criteria for an External Writer

Use the following form of the MODIFY command to specify the criteria that an external writer is to use in selecting data sets for processing.

|                                       |                                    |   |
|---------------------------------------|------------------------------------|---|
| <code>{<br/>MODIFY<br/>F<br/>}</code> | <code>[procname.]identifier</code> | <code>{<br/>,CLASS=[classes]<br/>,JOBID= [JOBnnnn<br/>          STCnnnn<br/>          TSUnnnn]<br/>,WRITER= [STDWTR<br/>          user-writer-name]<br/>,FORMS=[forms-name]<br/>,DEST= [LOCAL<br/>          remote-workstation-name]<br/>}</code> |
|---------------------------------------|------------------------------------|---|

### **procname**

The name of the cataloged procedure (for example, XWTR) used in the START writer command.

### **identifier**

The identifier, from the START command, of the writer to be modified. You can also use the device number associated with the writer you are modifying as the identifier. If you did not specify an identifier on the START command, the system automatically uses the procname you supplied in the START command as the default identifier.

### **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 hardcopy queue.

### **JOBID or J = JOBnnnn, STCnnnn, or TSUnnnn**

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 writer (that is, data sets that are to be processed by the 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.



**MODIFY Command****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**

Use the following form of the MODIFY command to cause an external writer to pause for operator intervention.

|   |                       |  |  |
|---|-----------------------|--|--|
| $\left\{ \begin{array}{l} \text{MODIFY} \\ \text{F} \end{array} \right\}$ | [procname.]identifier | $\left\{ \begin{array}{l} \text{,PAUSE} \\ \text{,P} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{=FORMS} \\ \text{=DATASET} \end{array} \right\}$ |
|---|-----------------------|--|--|

**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:* To stop writer 00E before it processes each new data set, enter:

F 00E,PAUSE=DATASET

## MODIFY Command

### Building and Replacing LNKLST Directories

Use the following form of the MODIFY command to cause the LNKLST lookaside (LLA) program to build a new copy of the LNKLST directories and then replace the old copy with this new copy.

|                         |                         |
|-------------------------|-------------------------|
| <pre>{ MODIFY F }</pre> | <pre>LLA, REFRESH</pre> |
|-------------------------|-------------------------|

#### LLA,REFRESH

Causes LLA to build a new copy of the LNKLST directory. This procedure is necessary when an installation makes changes to the LNKLST directories. Because system performance is degraded when LLA terminates, use this command rather than stopping and restarting LLA.

### Communicating with Catalog Address Space

Use the following form of the MODIFY command to display information about the catalog address space or request that the catalog address space perform a specified service.

|                         |   |
|-------------------------|---|
| <pre>{ MODIFY F }</pre> | <pre>CATALOG { ,LIST[(xxxxxxxx)] ,OPEN ,CLOSE(catname) ,VCLOSE(volser) ,REPORT ,ABEND(yyyyyyyyy) ,RESTART ,ENTRY[(cname)] ,END(ID) [ ,REDRIVE            ,NOREDRIVE ] }</pre> |
|-------------------------|---|

#### LIST[(xxxxxxxx)]

The catalog address space is to display the jobname of the user that the service task is working on behalf of, where *xxxxxxxx* is the full four byte address of the service task, in hexadecimal. If no address is specified, the system displays all service task addresses and user jobnames.

#### OPEN

Catalog address space displays a list of names of open catalogs and their volume serial numbers and use counts.

#### CLOSE(*catname*)

Catalog address space is to close the Integrated Catalog Facility (ICF) catalog, where *catname* is the name (up to 44 characters) of the ICF catalog.

#### VCLOSE(*volser*)

Catalog address space is to close the VSAM volume data set (VVDS), where *volser* is the volume serial number of the VVDS that is to be closed.

**MODIFY Command****REPORT**

Catalog address space is to display a summary of the catalog address space for diagnostic purposes.

**ABEND(yyyyyyyy)**

The catalog address space is to cause an abnormal termination of one service task with an abend code of 91A and take a SYS1.DUMP dump. *yyyyyyyy* is the full four-byte address of the TCB for the catalog address space service task, in hexadecimal. The termination is transparent to the user; the system honors outstanding requests to the catalog address space.

**RESTART**

The system terminates the catalog address space with an abend code of 81A and a SYS1.DUMP 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. (The name of the jobstep task is IGG0CLX0.)

**ENTRY[(cname)]**

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.

**END(ID)[,REDRIVE  
,NOREDRIVE]**

The catalog address space is to end the processing of the service task with the name specified by *ID*.

**REDRIVE**

The catalog request currently being processed will continue to be processed by a different task. REDRIVE will free up an ENQ lockout. REDRIVE is the default.

**NOREDRIVE**

Any catalog requests in process at the time will fail with a catalog return code of RC246.

*Example:* To list the names of catalogs that are open and their volume serial numbers and use counts, issue:

```
MODIFY CATALOG,OPEN
```

# MONITOR Command

## MONITOR Command

Use the MONITOR command to continually display jobnames, data set status, and time-sharing user sessions and to add certain information to mount and demount messages.

*Notes:*

1. To stop the MONITOR display, use the STOPMN command.
2. 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.
3. The SPACE and DSNAME operands are not routable.

The JOBNAMEs, SESS, and STATUS options on the MONITOR parameter on the CONSOLE statement in the CONSOLxx member of SYS1.PARMLIB also request jobnames, data set status, and TSO session information. The SPACE and DSNAME parameters on the MONITOR parameter on the INIT statement in the CONSOLxx member also determine whether additional information accompanies mount and demount messages. See "Displaying Jobname, Data Set Status, and TSO Information" in Chapter 3.

The complete syntax for the MONITOR command is:

|                               |  |
|-------------------------------|--|
| <pre>{ MONITOR } { MN }</pre> | <pre>{ JOBNAMEs [ ,T ] [ ,L=cc ]   DSNAME   SPACE   STATUS [ ,L=cc ]   SESS [ ,T ] [ ,L=cc ] }</pre> |
|-------------------------------|--|

### JOBNAMEs

The name of each job is to be displayed when the job starts and terminates, and unit record allocation is to be displayed 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 nontemporary 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 data set names and volume serial numbers of data sets with dispositions of KEEP, CATLG, or UNCATLG are to be displayed whenever they are freed.

### SESS

The user identifier for each TSO terminal is to be displayed 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.

**L=cc**

The console where the display is to be presented. If you omit this operand, the console on which the MONITOR command is entered is assumed.

*Example:* To display on console 3 the job name information and the time when any job starts or terminates, enter:

```
mn jobnames,t,l=3
```

## MOUNT Command

## 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 or MSS virtual unit, 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 MSS virtual unit, you must also enter a JES3 mount command (\*MODIFY,S,M=...). See *Operator's Library: JES3 Commands*.

If you are mounting an IBM 3348 Model 70F Data Module, you must make sure that the 3340 drive specified has the fixed-head feature installed.

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.

The complete syntax for the MOUNT command is:

|  |  |        |   |          |   |      |   |
|--|--|--------|---|----------|---|------|---|
| $\left\{ \begin{array}{l} \text{MOUNT} \\ \text{M} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{devnum} \\ \text{devicetype} \end{array} \right\}$ | ,VOL=( | $\left\{ \begin{array}{l} \text{NL} \\ \text{SL} \\ \text{AL} \end{array} \right\}$ | ,serial) | , | USE= | $\left[ \begin{array}{l} \text{STORAGE} \\ \text{PUBLIC} \\ \text{PRIVATE} \end{array} \right]$ |
|--|--|--------|---|----------|---|------|---|

### **devnum**

The device number of the input/output device to be mounted.

### **devicetype**

The type of device to be mounted. It can be any IBM-supplied name (for example, 3380). A list of IBM device types is included in *System Generation Reference*.

### **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.

### **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. The VOL parameter is optional when using SL tapes. 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 generation. The VOL parameter is optional when using AL tapes. 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 *SPL: System Modifications* for more information on USE attributes.

*Example:* To request that a volume with a standard label of 222222 be mounted on device 282, enter:

```
m 282,vol=(sl,222222),use=private
```

## MSGRT Command

## 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 member of SYS1.PARMLIB controls the display routing instructions at IPL.

The complete syntax for the MSGRT command is:

|                       |  |   |
|-----------------------|--|---|
| {<br>MSGRT<br>MR<br>} | {<br>D=(operand[,operand]...)<br>TR=A<br>K<br>CF<br>NONE<br><u>REF</u><br>CONTn<br>[MN] [,L=cc]<br>} | {<br>[,L={<br>a<br>cc<br>cca<br>}]<br>} |
|-----------------------|--|---|

### D = option

The system is to route DISPLAY command output to the specified MCS console. Any display operand that produces a status display is valid on this command. If you specify only one operand, the parentheses are not necessary. Note that the options on the MSGRT parameter on the CONSOLE statement in the CONSOLxx member are not always the same as the options on the D= operand of the MSGRT command.

The display options are:

#### A

Display the number of active batch jobs, logged-on time-sharing users, and active initiators.

#### ASM

Display information about any page and swap data sets the system is currently using.

#### C

Display a summary of the CONTROL command operands and the functions they perform.

#### CONSOLES

Display the status of the system console configuration.

#### D

Display the domain description table.



### DUMP

Display DUMP options and the status of the SYS1.DUMPxx data sets.

### M

Display the status of processors, channel paths, devices, and storage.

### R

Display the identification numbers of messages waiting for replies, unfulfilled mount requests, and any units requiring operator intervention.

### S

Display SLIP trap information

### U

Display unit information about all devices.

### 3850

See *Mass Storage System Extensions Operation* for information about MSS commands.

### SMF

Display SMF status or options.

### GRS

Display the state of the global resource serialization complex.

### MPF

Display the status of the message processing facility (MPF)

### PFK

Display the commands associated with program function keys (PFKs).

### TR=A

The system is to route the TRACK A command display and the action of the STOPTR command to the specified MCS console.

### K

Subsequent CONTROL command action is to affect the specified MCS console until the option is discontinued or you IPL the system. Only those CONTROL functions that you can explicitly route with the L=cca operand can be routed by default using MSGRT.

### CF

The system is to route the reply to the CONFIG command to the specified MCS console.

### L = a, cc, or cca

The display area (a), console identifier (cc), or both (cca) of the active MCS console where the specified display is to appear.

### 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. REF is the default.

# MSGRT Command

## CONT<sub>n</sub>

The system is to continue the display response to a previous MR REF command. When CONT<sub>n</sub> appears in the display response, all current routing instructions have not been displayed. Issue MR CONT<sub>n</sub> 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 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 member COMMNDxx of SYS1.PARMLIB, 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)
```

## 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

Use the PAGEADD command only at the request of your system programmer. PAGEADD adds auxiliary storage space (page and swap 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 or the CVIO (clear virtual I/O) option. The swap data sets are not preserved across an IPL. PAGEADD can also direct VIO data sets away from a page data set that is being added.

You might need to add auxiliary storage space if any of the following 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 following message is issued:

```
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.

Only one type of data set (page or swap) can be added to the system with a single PAGEADD command. Requested page or swap 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 swap or page data sets that can be in use by the system is limited by the number specified on the PAGTOTL system parameter (see *Initialization and Tuning*). 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 or swapping 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. For information about the commands that control subsystem caches, see *MVS/XA Cache Device Administration Guide*.

The page or swap 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. Information on defining data sets appears in *System Generation Reference*.

To identify the page and swap data sets the system is currently using, use the DISPLAY ASM command.

## PAGEADD Command

The complete syntax for the PAGEADD command is:

|                                  |  |
|----------------------------------|--|
| <pre>{PAGEADD } {PA      }</pre> | <pre>{ dsname[ ,dsname]...   SWAP=dsname[ ,dsname]...   NONVIO=dsname[ ,dsname]... }</pre> |
|----------------------------------|--|

### **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.

### **SWAP = dsname[,dsname]...**

The name of one or more swap data sets to be added. If *dsname* is not the name of a swap 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:

```
pageadd page3
```

*Example 2:* To add three page data sets, enter:

```
pa sys1.page01,sys1.page02,page3
```

*Example 3:* To add two swap data sets, enter:

```
pageadd swap=swp11,swap9
```

*Example 4:* To add SYS1.PAGE01 as a page data set and specify that it is not to be used for VIO paging, enter:

```
pageadd nonvio=SYS1.PAGE01
```

## QUIESCE Command

Use the QUIESCE command when you want 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 the master console. You can restart the system by performing the RESTART function.

```
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 0000CCC. See *Message Library: System Codes* for more information on wait state code CCC. You might receive the following message on the master console or its alternate:

```
IEE082I SYSTEM WAIT STATE CODE 'CCC'X - QUIESCE FUNCTION PERFORMED
```

When you perform the restart function and the system restarts successfully, you receive the following message and the system continues normal processing:

```
IEE752I QUIESCE WAS SUCCESSFUL
```

*Notes:*

1. *If this system is actively using global resource serialization to share global resources, 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

## 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.

Using the REPLY command, you can:

- Reply to system requests for information
- Reply to system requests during recovery processing
- Specify system parameters
- Set the time of day clock and specify IPS (installation performance specification) parameters
- Specify SMF options
- Specify dump options after issuing the DUMP command

The complete syntax for the REPLY command is:

|  |    |   |
|--|----|---|
| $\left\{ \begin{array}{l} \text{REPLY} \\ \text{R} \end{array} \right\}$ | id | $\left( \begin{array}{l} \text{'text'} \\ \text{'parm=,'} \\ \text{'parm=,parm'} \\ \text{'parm=value'} \\ \text{'parm=(value[,value]...[,L]'} \\ \text{'parm=(value,value),parm=value'} \\ \text{'DATE=yy,ddd,[CLOCK=hh.mm.ss] [,GMT] [,IPS=nn]} \\ \\ \text{,U} \\ \left[ \begin{array}{l} \text{,ASID=(n[,n]...)} \\ \text{,JOBNAME=(name[,name]...)} \\ \text{,TSONAME=(name[,name]...)} \end{array} \right] \left[ \text{,SDATA=(option[,option]...)} \right] \\ \left[ \text{,STOR=(beg,end[,beg,end]...)} \right] \left[ \begin{array}{l} \text{,CONT} \\ \text{,END} \end{array} \right] \end{array} \right)$ |
|--|----|---|

### Replying to Disabled-Console Communication Messages

When disabled console communication routines issue messages that require your response, the reply line is pre-formatted on the warning line of the master console (or first alternate console). The data on that line is in the following format:

```
R 00, _
```

There is a blank (space) between the comma and the cursor (\_). Do not move the cursor before you reply to the message. If you do, input will be inhibited.

If the disabled console communication facility issues a message on the system console, "R 00, " will not appear on the reply line. Do not enter these characters with your reply.

## Replying to System Information Requests

Use this form of the REPLY command to respond to system requests. To review outstanding requests before replying, issue DISPLAY R.

|                        |   |
|------------------------|---|
| <pre>{REPLY} {R}</pre> | <pre>id [ , { 'text' } ]       {       text       }</pre> |
|------------------------|---|

### id

The identification number (0-99), as specified in the message requesting a response. A leading zero can be omitted.

### 'text'

The response to the message. The apostrophes are optional and need only be included if your answer contains uppercase and lowercase characters or a comment. If you include the apostrophes and your answer contains an apostrophe, use two apostrophes in the message text.

A blank entered immediately after the identification number indicates a null reply; you can comment following the blank in a null reply. Once JES2 or JES3 is active, you can use an abbreviated format for the REPLY command that does not require the entry of either REPLY or R (refer to the appropriate operations publication).

*Example 1:* To reply 'U' to system message 03, enter:

```
r 03,u
```

*Example 2:* To add a comment after the text of the reply to communicate additional information to the programmer issuing a WTOR (see "Command Syntax" at the beginning of this chapter for information about the use of the comment field), enter:

```
r xx,please call the operator
```

## Replying to System Requests During Recovery Processing

To respond to system requests during recovery processing, use this form of the REPLY command.

|                        |   |
|------------------------|---|
| <pre>{REPLY} {R}</pre> | <pre>id [ , { 'text' } ]       {       text       }</pre> |
|------------------------|---|

During recovery processing, normal console operations are suspended and the system uses the master console or its alternate to communicate with you. If recovery processing issues a message that requires a response, it places R 0 in the entry area of the screen and waits for your reply. Do not include the command name or the id in your response.

# REPLY Command

## Specifying System Parameters

During system initialization, you receive the following message:

```
IEA101A SPECIFY SYSTEM PARAMETERS FOR RELEASE xx.yy.zzz
```

where xx is the release number, yy is the release level, and zzz is the system type.

You must reply to this message with the REPLY command. The abbreviated JES replies are invalid because JES is not yet active. 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.

|   |  |
|---|--|
| $\left. \begin{array}{l} \text{REPLY} \\ \text{R} \end{array} \right\}$ | $\text{id, } \left\{ \begin{array}{l} \text{U} \\ \text{parm=,} \\ \text{parm=,parm,} \\ \text{parm=value} \\ \text{parm=(value[,value]...[,L])} \\ \text{parm=(value,value),parm=value} \end{array} \right\} \left[ \text{,CONT} \right] \left. \right\}$ |
|---|--|

### id

The identification number (0-99) as specified in the message requesting information. A leading zero can be omitted.

### 'U'

No parameters are to be changed. The system uses the default list of system parameters in SYS1.PARMLIB.

### 'parm =,'

The parameter, as specified in SYS1.PARMLIB, 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 SYS1.PARMLIB, 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 SYS1.PARMLIB. 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 no greater than 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; for 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 *Initialization and Tuning* or ask your system programmer.

After you have specified the system parameters, the system asks you to state the reason for the IPL:

```
* id IFB010D ENTER 'IPL REASON, SUBSYSTEM ID,' OR 'U'
```

You must reply to this message with the REPLY command, as described earlier under “Replying to System Information Requests.”

## 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=yy.ddd[,CLOCK=hh.mm.ss][,GMT]'
```

where yy is the year (00-99), 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:* The apostrophes in the above reply are optional.

If you included GMT in your reply, the time and date are Greenwich mean time. If you omitted GMT, the system assumes the values are the local time and date, converts them to Greenwich mean time values, and sets the clock(s) with the Greenwich mean time. You can also specify the SET IPS command at this time. See the SET command later in this chapter.

When you have entered a valid reply to message IEA886A, the system issues message IEA903A, requesting you to press the TOD clock security switch. Reply U to message IEA903A and, at the exact time that matches the TOD clock setting, depress the security switch. 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=yy.ddd,CLOCK=hh.mm.ss  
IEA888A LOCAL DATE=yy.ddd,CLOCK=hh.mm.ss REPLY U, OR GMT/LOCAL TIME
```

If the values are acceptable, reply ‘U’. If you want to change the value of the TOD clock, enter a new date, time, or both as follows:

```
R id, '[DATE=yy.ddd][,CLOCK=hh.mm.ss][,GMT]'
```

*Note:* The system automatically issues message IEA888A at IPL time if the OPERATOR parameter is included in the active CLOCKxx member of SYS1.PARMLIB. (See *Initialization and Tuning* for details.)

## REPLY Command

If you specified a different TOD clock setting, message IEA903A is issued asking you to press the switch as described above. If you omitted GMT, the local date and/or time are assumed. If you want to specify a high local date, specify 99.365 (not 99.366). Once you have set the new time and/or date, the system re-issues message IEA888A with new values. Reply to the message as previously described.

*Note:* The TOD clock should be set 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 C.E. servicing 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 SMF Options

If PROMPT (LIST or ALL) is specified in the SMFPRMxx member of SYS1.PARMLIB, you can change the SMF options, and you receive the following message:

```
IEE354I SMF PARAMETERS
```

The system lists the SMF parameters specified at system generation time and issues the following message:

```
* id IEE357A REPLY WITH SMF VALUES OR U
```

If you do not want to change the SMF values, enter:

```
R id,U
```

If you want to change the SMF values, use the following form of the REPLY command:

```
R id,options
```

where options is the list of SMF parameters to be changed. These options must be separated by commas. The SMF options are generally provided by your system programmer. A description of these parameters can be found in *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 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:

|                              |  |
|------------------------------|--|
| <pre> REPLY R         </pre> | <pre> id {   ,U   ,ASID=(n[,n]...)   ,JOBNAME=(name[,name]...)   ,TSOname=(name[,name]...) } [,SDATA[(option[,option]...)]                                      [,STOR=(beg,end[,beg,end]...)]                                      [,CONT]                                      [,END] }         </pre> <p>Note: CONT or END, when specified, must be the last parameter on the input line.</p> |
|------------------------------|--|

For a detailed explanation of the DUMP options, see “Specifying Dump Options” under the DUMP command in this chapter.

## RESET Command

## RESET Command

Use the RESET command to change the performance group of a job currently in execution. This command affects the current job step and all subsequent job steps in this execution. This command should be used only at the direction of the system programmer.

The complete syntax for the RESET command is:

|  |                   |
|--|-------------------|
| $\left\{ \begin{array}{l} \text{RESET} \\ \text{E} \end{array} \right\}$ | jobname,PERFORM=n |
|--|-------------------|

### **jobname**

The name of the job whose performance group is to be changed.

### **PERFORM = n**

The number, between 1 and 999, to be assigned as the new performance group value.

*Example:* To change the performance group value of job TMASGJ02 to 1, enter:

```
e tmasgj02,perform=1
```

## SEND Command

Use the SEND command to communicate with other operators in a multiple-console support (MCS) 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.BROADCAST data set.

The SEND command enables you to:

- Communicate with other operators in an MCS environment
- Send messages to specific users or to all time-sharing users currently receiving messages
- Send messages to specific time-sharing users or to all time-sharing users logging on the system
- Save messages in the SYS1.BROADCAST data set
- List messages in the SYS1.BROADCAST data set
- Delete messages from the SYS1.BROADCAST data set

The complete syntax for the SEND command is:

|  |  |
|--|--|
| $\left\{ \begin{array}{l} \text{SEND} \\ \text{SE} \end{array} \right\}$ | $\left( \begin{array}{l} \left[ \begin{array}{l} \text{'message'} \\ \text{msgno} \end{array} \right] \left[ \begin{array}{l} \text{,BRDCST} \\ \text{,OPERATOR=routecode} \\ \text{,CN=console} \\ \left[ \begin{array}{l} \text{,USER=(userid[,userid]...)} \\ \text{,ALL} \end{array} \right] \left[ \begin{array}{l} \text{,LOGON} \\ \text{,SAVE} \\ \text{,NOW} \end{array} \right] \end{array} \right] \\ \text{msgno} \left\{ \begin{array}{l} \text{,DELETE} \\ \text{,LIST} \end{array} \right\} \\ \text{LIST} \end{array} \right)$ |
|--|--|

# SEND Command

## Communicating with Other Operators

Use the SEND command to communicate with other operators at MCS consoles.

|                                |  |  |
|--------------------------------|--|--|
| <pre>{ SEND }<br/>{ SE }</pre> | <pre>{ 'message' }<br/>{ msgno }</pre> | <pre>{ ,BRDCST<br/>{ ,OPERATOR=routecode<br/>{ ,CN=console }</pre> |
|--------------------------------|--|--|

### 'message'

The message to be sent.

### msgno

The number of the message in SYS1.BROADCAST to be sent. (See "Saving Messages in the Broadcast Data Set" under this command 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.

### OPERATOR = routecode

The installation area (such as tape library) to receive the message, specified as a one or two digit number between 1 and 15 (see Figure 3-8 earlier in this manual).

### 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.

When you are uncertain of the routing codes and console identifiers in effect, enter the DISPLAY CONSOLES command. See "Displaying Console Configuration Information" under the DISPLAY command in this chapter.

*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 following form of the SEND command to communicate with specific time-sharing users.

|  |  |
|--|--|
| $\left\{ \begin{array}{l} \text{SEND} \\ \text{SE} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{'message' } \\ \text{msgno} \end{array} \right\} , \text{USER}=(\text{userid}[\text{,userid}] \dots) \left[ \begin{array}{l} \text{,LOGON} \\ \text{,NOW} \end{array} \right]$ |
|--|--|

### **'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" under this command for information on 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.

### **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.

### **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. NOW is the default value when NOW and LOGON are omitted.

*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)
```

# SEND Command

## Communicating with All Terminal Users

Use the following form of the SEND command to send a message to all terminal users currently logged on the system.

|  |   |  |
|--|---|--|
| $\left\{ \begin{array}{l} \text{SEND} \\ \text{SE} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{'message' } \\ \text{msgno} \end{array} \right\}$ | $\left[ \begin{array}{l} \text{,LOGON} \\ \text{,NOW} \end{array} \right]$ |
|--|---|--|

### **'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" under this command for information on how to save and later send messages by message number.)

### **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 mail 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.

### **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. NOW is the default value if NOW and LOGON are omitted.

*Note:* When possible, use the LOGON parameter so you do not interrupt the user's terminal session 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 following form of the SEND command to store messages in the broadcast data set to be issued only at LOGON time, or when requested.

|                            |   |                  |
|----------------------------|---|------------------|
| <pre>{ SEND } { SE }</pre> | <pre>{ 'message' } [ ,USER=(userid[,userid]...) ] { msgno } [ ,<u>ALL</u> ]</pre> | <pre>,SAVE</pre> |
|----------------------------|---|------------------|

### '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 is printed when the message is stored and can be used as *msgno* in this and other forms of the SEND command. ALL is the default value if both USER = and ALL are omitted.

### 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.

*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.

## SEND Command

### Listing the Notices Section of the Broadcast Data Set

Use the following form of the SEND 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 present.

|                                |                         |
|--------------------------------|-------------------------|
| <pre>{ SEND }<br/>{ SE }</pre> | <pre>[msgno,]LIST</pre> |
|--------------------------------|-------------------------|

#### **msgno**

The number of the message to be listed. If this operand is omitted, all messages in the notices section are listed, along with the message numbers assigned to them.

#### **LIST**

The requested message or all messages in the notices section of the SYS1.BROADCAST data set are to be listed on the console.

*Example 1:* To list all messages in the notices section of the SYS1.BROADCAST data set, enter:

```
se list
```

*Example 2:* To list message number 21, enter:

```
SEND 21,LIST
```

### Deleting a Message from the Notices Section of the Broadcast Data Set

If you find, after listing the notices section of the broadcast data set, that a message is no longer needed, use the following form of the SEND command to delete it.

|                                |                         |
|--------------------------------|-------------------------|
| <pre>{ SEND }<br/>{ SE }</pre> | <pre>msgno,DELETE</pre> |
|--------------------------------|-------------------------|

#### **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 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 set of available PFK tables

The complete syntax for the SET command is:

|   |  |
|---|--|
| $\left. \begin{array}{c} \text{SET} \\ \text{T} \end{array} \right\}$ | $\left[ \begin{array}{l} [\text{DATE}=\text{yy}.\text{ddd}] [\text{,CLOCK}=\text{hh}.\text{mm}.\text{ss}] \\ \text{RESET} \end{array} \right] [\text{,IPS}=\text{xx}] [\text{,OPT}=\text{xx}] [\text{,ICS}=\text{xx}] [\text{,SMF}=\text{xx}]$<br><br>$[\text{,DAE}=\text{xx}] \left[ \text{,MPF} = \left\{ \begin{array}{c} \text{xx} \\ \text{NO} \end{array} \right\} \right] [\text{SLIP}=\text{xx}] [\text{PFK}=\text{xx}]$ |
|---|--|

*Note:* Operands can be specified in any order. 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.

### DATE = yy.ddd

The year (00-99) and the day (001-366). If you specify a new time that implies a change of date, you must explicitly specify the new date. Specify 99.365 when a high date is desired, because the system does not recognize 99.366 as a high date.

### CLOCK = hh.mm.ss

The time in hours (00-23), minutes (00-59), and seconds (00-59). (The time value is local time; you can specify the GMT time value only from the REPLY command.) The system does not change the date when the new time implies change of date; if you want a new date, use the DATE parameter or wait for the time to pass midnight.

### RESET

The local date and time are set to the values that are in the CLOCKxx member of SYS1.PARMLIB. If you specify RESET, do not specify DATE and CLOCK.

The following parameters determine which members of SYS1.PARMLIB the system is to use. Use them only at the direction of the system programmer. For more information on members of SYS1.PARMLIB, see *Initialization and Tuning*.

## SET Command

### **IPS = xx**

The two alphanumeric characters indicating the IEAIPSxx member of SYS1.PARMLIB that contains the new parameters SRM is to use.

### **OPT = xx**

The two alphanumeric characters indicating the IEAOPTxx member of SYS1.PARMLIB that contains the new parameters SRM is to use.

### **ICS = xx**

The two alphanumeric characters indicating the IEAICSxx member of SYS1.PARMLIB that contains the new installation control specifications SRM is to use.

### **SMF = xx**

The two alphanumeric characters indicating the SMFPRMxx member of SYS1.PARMLIB that contains the parameters MVS/XA is to use when it restarts SMF.

### **DAE = xx**

The two alphanumeric characters indicating the ADSETxx member of SYS1.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**

The two alphanumeric characters indicating the MPFLSTxx member of SYS1.PARMLIB that contains:

- The message IDs or prefixes of messages to be processed
- The color, intensity, and highlighting options

The MPF(xx) parameter on the INIT statement in the CONSOLxx member of SYS1.PARMLIB controls which, if any, MPFLSTxx member is active at IPL.

### **MPF = NO**

Ends MPF processing (message suppression and presentation).

### **SLIP = xx**

The two alphanumeric characters indicating the IEASLPxx member of SYS1.PARMLIB that contains the commands SLIP processing is to use.

### **PFK = xx**

The two alphanumeric characters indicating the PFKTABxx member of SYS1.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.

*Example 1:* When the displayed local time and date are 19.00.00 and 82.141, respectively, to set the local time ahead to 1:00 a.m., enter:

```
T DATE=82.142,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 or in a previous SET command, 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 SYS1.PARMLIB, enter:

```
t ics=00,ips=22
```

*Example 4:* To restart SMF with the parameters found in the SMFPRMAA member of SYS1.PARMLIB, enter:

```
t smf=aa
```

## SETDMN Command

### SETDMN Command

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).

Use the following syntax for the SETDMN command if the **WT** subparameter was specified or assumed by default in the current IPS.

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{SETDMN} \\ \text{SD} \end{array} \right\}$ | $\text{domainnum} \left\{ \begin{array}{l} ,\text{MIN}=\text{n1} \quad [ ,\text{MAX}=\text{n2}] \quad [ ,\text{WT}=\text{n3}] \\ ,\text{MAX}=\text{n2} \quad [ ,\text{MIN}=\text{n1}] \quad [ ,\text{WT}=\text{n3}] \\ ,\text{WT}=\text{n3} \quad [ ,\text{MAX}=\text{n2}] \quad [ ,\text{MIN}=\text{n1}] \end{array} \right\}$ |
|--|---|

Use the following syntax for the SETDMN command if the **WT** subparameter was not specified or assumed by default in the current IPS.

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{SETDMN} \\ \text{SD} \end{array} \right\}$ | $\text{domainnum} \left\{ \begin{array}{l} ,\text{MIN}=\text{n1} \quad [ ,\text{MAX}=\text{n2}] \quad \left[ \begin{array}{l} ,\text{AOBJ}=\text{n4} \\ ,\text{DOBJ}=\text{n5} \\ ,\text{FWKL}=\text{n6} \end{array} \right] \\ \left[ \begin{array}{l} ,\text{AOBJ}=\text{n4} \\ ,\text{DOBJ}=\text{n5} \\ ,\text{FWKL}=\text{n6} \end{array} \right] \quad [ ,\text{MIN}=\text{n1}] \quad [ ,\text{MAX}=\text{n2}] \\ ,\text{MAX}=\text{n2} \quad \left[ \begin{array}{l} ,\text{AOBJ}=\text{n4} \\ ,\text{DOBJ}=\text{n5} \\ ,\text{FWKL}=\text{n6} \end{array} \right] \quad [ ,\text{MIN}=\text{n1}] \end{array} \right\}$ |
|--|---|

*Note:* The AOBJ, DOBJ, and FWKL keywords can be specified only if they have been specified or assumed by default in the current IPS. Specifying the AOBJ, DOBJ, or FWKL keywords overrides any previous value set for them either in the current IPS or in another SETDMN command.

#### SETDMN Command Restrictions:

1. The value for *n2* in the MAX=*n2* parameter must be greater than or equal to the value of *n1*.
2. At least one keyword must be specified.
3. Duplicate keywords cannot be specified.
4. Keywords can be specified in any order.

**SETDMN Parameter Descriptions:****domainnum**

The domain table entry (1-128) to be modified.

**MIN = n1**

The minimum multiprogramming level (0-255).

**MAX = n2**

The maximum multiprogramming level (0-255).

**WT = n3**

The weighting factor (1-255).

**AOBJ = n4**

Specifies the number of a performance objective (1-64) for controlling the average service rate per ready user in a domain.

**DOBJ = n5**

Specifies the number of a performance objective (1-64) for controlling the total service rate for a domain.

**FWKL = n6**

Specifies a fixed workload level (1-128) for this domain.

*Note:* For more details on the parameters used in the SETDMN command, see *Initialization and Tuning*.

*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, the maximum MPL to 4, and the weighting factor to 1 in domain 6, enter:

```
SD 6,MIN=3,MAX=4,WT=1
```

*Example 3:* For domain 2, to set the minimum MPL to 0, the maximum MPL to 255, and the performance objective to 5 to control the average service rate per ready user in the domain, enter:

```
SD 2,MIN=0,MAX=255,AOBJ=5
```

Specifying **AOBJ = 5** overrides any previous values set for the **AOBJ**, **DOBJ**, or **FWKL** keywords in domain 2.

## SETSMF Command

## SETSMF Command

In contrast to the SET command, which allows an installation to specify a different SMFPRMxx member of SYS1.PARMLIB or restart SMF, the SETSMF command allows an installation to add a SUBPARM parameter or replace any previously-specified parameter in the active SMF member of SYS1.PARMLIB except the ACTIVE, PROMPT, SID, or EXITS parameters. 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.

The complete syntax for the SETSMF command is:

|                                  |   |
|----------------------------------|---|
| <pre>{ SETSMF }<br/>{ SS }</pre> | <pre>parameter (value[,value]...)</pre> |
|----------------------------------|---|

The parameters are:

**parameter**

specifies any SMF parameter in the SMFPRMxx member except ACTIVE, PROMPT, SID, or EXITS.

**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:* 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))
```



## SLIP Command

Use the SLIP command only at the direction of the system programmer. The command can be entered from a console or from a TSO terminal. Because the SLIP command is complex, prompting is provided to help you enter it correctly.

The SLIP command controls SLIP (serviceability level indication processing), a diagnostic aid designed to intercept or trap certain system events. You can indicate what kinds of events you want trapped and what the system is to do when these events occur.

The kinds of events you can intercept are:

- Program event recording (PER) events
  - Instruction fetch PER interruption
  - Successful branch PER interruption
  - Storage alteration PER interruption
- Error events
  - Paging error
  - Dynamic address translation error
  - Machine check associated software error
  - Address space termination error
  - SVC 13 issued by a task
  - SVC error
  - Program check interruption
  - Restart interruption

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 GTF trace 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 process to get control
- Ignore the event
- Cause a system trace table record to be written
- Cause a SYS1.LOGREC record to be written

The PER and error 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 by qualifying the event, you can specify exactly what state 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 you are not interested in. When conditions specified are the same as those in the system, this is called a *match*. When conditions specified are not the same as those in the system, this is called a

## SLIP Command

*no-match.* Only when all the conditions you specify match will your action be taken. Among the conditions you can specify are:

- The type of error being processed
- 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 interruption
- The type of dump that would result from an abend

If you do not specify a particular condition, then the system does not check for that condition.

When you define more than one SLIP trap, SLIP processing first examines the last defined trap. If it does not find a match condition, it proceeds to the previously defined trap.

If you issue a multiline SLIP command from a TSO CLIST, use the line continuation character at the end of text on each line and the END keyword on the last line.

There are three types of SLIP commands:

- SLIP SET commands define SLIP traps.
- SLIP MOD commands enable or disable previously defined SLIP traps.
- SLIP DEL commands delete previously defined SLIP traps.

See *SPL: System Macros and Facilities* for more information about the use of SLIP to intercept software errors. For more information about designing an effective SLIP trap, see *Diagnostic Techniques*.

You can place SLIP commands in IEACMD00, COMMNDxx, or IEASLPyy members of SYS1.PARMLIB. It is recommended that you place your SLIP commands in IEASLPyy and issue SET SLIP=yy. IEASLPyy is the only member that does not require that you restrict a command to one line. In addition, unlike commands placed in IEACMD00 or COMMNDxx, SLIP processes commands in IEASLPyy in the order in which they are specified. If you move SLIP commands that you currently have in other members to IEASLPyy, add the following command to IEACMD00 or COMMNDxx:

```
COM 'SET SLIP=yy'
```

Because the SLIP command is such a powerful diagnostic tool, it has several forms, each with many keywords. The following introductory syntax gives you an overview of the entire command.

|  |   |
|--|---|
| <pre> { SLIP } { SL   {     { [,options]       {         { [,IF[,options]]           {             { [,SB[,options]]               {                 { [,SA[,options]]               }             }           }         }       }     }   } } </pre> <p>SET { } ,END</p> <p>MOD { ,ENABLE } { ,ALL }<br/> { ,DISABLE } { ,ID=xxxx }</p> <p>DEL { ,ALL }<br/> { ,ID=xxxx }</p> | <p>Defines an error event (non-PER) trap. The required syntax is fully described under "Setting a Non-PER SLIP Trap."</p>   |
|  | <p>Defines an instruction fetch PER trap. The required syntax is fully described under "Setting an Instruction Fetch or Successful Branch PER Trap."</p>                        |
|  | <p>Defines a successful branch PER trap. The required syntax is fully described under "Setting an Instruction Fetch or Successful Branch PER Trap."</p>                         |
|  | <p>Defines a storage alteration PER trap. The required syntax is fully described under "Setting a Storage Alteration PER Trap."</p>   |
|  | <p>Makes previously defined SLIP traps eligible for checking (ENABLE) or ineligible for checking (DISABLE). The complete syntax is described under "Modifying a SLIP Trap."</p> |
|  | <p>Deletes a previously defined SLIP trap. The complete syntax is described under "Deleting a SLIP Trap."</p>   |

The keywords that can be used with any form of the SLIP SET command are defined after the syntaxes of the command. "Keyword Relationships" presents a brief description of the functional groups of SLIP SET keywords. "Keywords for SLIP SET Commands" first describes the positional keywords (IF, SB, SA, and END), then the remaining keywords, some of which appear on all types of SLIP SET commands and some of which are unique to one form of the command, in alphabetical order.

# SLIP Command

## Setting a Non-PER SLIP Trap

Use this form of the SLIP SET command to set an error event (non-PER) trap.

|                          |  |
|--------------------------|--|
| <pre>{ SLIP   SL }</pre> | <pre>SET [ ,ACTION= {   IGNORE [ ,RECORD]   NODUMP [ ,RECORD]   (nodumptype [ ,nodumptype] ...) [ ,RECORD]   NOSUP [ ,RECORD]   RECORD   SVCD [ ,options] [ ,RECORD]   TRACE [ ,TRDATA=(options) ] [ ,RECORD]   TRDUMP [ ,options] [ ,RECORD]   WAIT [ ,RECORD] } ]  [ ,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=(id [ ,id] ...) ] [ ,COMP=code1 [ ,REASON=code2 ] ]  [ ,DATA=(comparisons) ]  [ ,DEBUG] [ ,ENABLE ] [ ,ERRTYP=(options) ] [ ,ID=xxxx ] [ ,DISABLE ]  [ ,JOBNAME=name ] [ ,JSPGM=jobstep-program-name ]  [ ,MATCHLIM=m ] [ ,MODE=(options) ] [ ,RBLEVEL= {   ERROR   PREVIOUS   NOTSSVRB } ] ,END</pre> |
|--------------------------|--|

### Notes:

1. Do not enter a SLIP trap that uses all the defaults (SLIP SET,END). The defaults taken (ENABLE, RBLEVEL=ERROR, and ACTION=SVCD) cause the system to issue message IEA412I and take multiple SVC dumps for non-error conditions.
2. If you specify only one nodumptype for the ACTION= parameter, you do not need to enclose it in parentheses.

Setting an Instruction Fetch or Successful Branch PER Trap

Use this form of the SLIP SET command to set an instruction fetch or successful branch PER trap.

|                      |  |
|----------------------|--|
| <p>{SLIP<br/>{SL</p> | <pre> SET { ,IF } [ ,ACTION= {     ( IGNORE [ ,RECOVERY ] )     RECOVERY     STRACE     STDUMP [ ,options ]     ( SVCD [ ,RECOVERY ] ) [ ,options ]     ( TRACE [ ,RECOVERY ] ) [ ,TRDATA=(options) ]     ( TRDUMP [ ,RECOVERY ] ) [ ,options ]     ( WAIT [ ,RECOVERY ] ) } ] [ ,ASID=(id [ ,id ] ...) ] [ ,DATA=(comparisons) ] [ ,DEBUG ] { ,ENABLE               ,DISABLE } [ ,ID=xxxx ] [ ,JOBNAME=name ] [ ,JSPGM=jobstep-program-name ] { ,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 ] ) } [ ,MATCHLIM=m ] [ ,MODE=(options) ] [ ,PRCNTLIM=p ] ,END </pre> |
|----------------------|--|

# SLIP Command

## Setting a Storage Alteration PER Trap

Use this form of the SLIP SET command to set a PER storage alteration trap. The RANGE keyword is required unless ACTION = IGNORE is specified, in which case RANGE cannot be specified.

|                |  |
|----------------|--|
| { SLIP<br>SL } | <pre> SET, SA [ , ACTION= {   ( IGNORE [ , RECOVERY ] )   RECOVERY   STRACE   STDUMP [ , options ]   ( SVCD [ , RECOVERY ] ) [ , options ]   ( TRACE [ , RECOVERY ] ) [ , TRDATA = ( options ) ]   ( TRDUMP [ , RECOVERY ] ) [ , options ]   ( WAIT [ , RECOVERY ] ) } ]  [ , 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 = ( id [ , id ] ... ) ]  [ , ASIDSA = ( id [ , id ] ... ) ] [ , DATA = ( comparisons ) ]  [ , DEBUG ] {   , ENABLE   , DISABLE } [ , ID = xxxx ] [ , JOBNAME = name ]  [ , JSPGM = jobstep-program-name ] [ , MATCHLIM = m ] [ , MODE = ( options ) ]  [ , PRCNTLIM = p ] , RANGE = ( start [ , end ] ) , END </pre> |
|----------------|--|

## Keyword Relationships

SLIP SET keywords fall into six functional groups: trap type keywords, event filter keywords, action-related keywords, trap control keywords, dump and trace tailoring keywords, and specialized keywords.

The trap-type keywords are IF, SA, and SB. Each indicates a specific type of PER interruption trap. Omitting all trap-type keywords also has meaning; it indicates an error detection trap, which is called a non-PER trap.

The event filter keywords are ADDRESS, ASID, ASIDSA, COMP, DATA, ERRTYP, JOBNAME, JSPGM, LPAEP, LPAMOD, MODE, NUCEP, NUCMOD, PVTEP, PVTMOD, RANGE, and REASON. Event filter keywords define the type of event or events the trap is to monitor; all other events are filtered out. For example, specifying ERRTYP = DAT indicates that, for this particular trap, you are interested only in dynamic address translation errors. You can, of course, specify various combinations of these event filter keywords.

The action-related keyword is ACTION. The operands of ACTION (IGNORE, NODUMP, NOSUP, NOSVCD, NOSYSA, NOSYSM, NOSYSU, RECORD, RECOVERY, STDUMP, STRACE, SVCD, TRACE, TRDUMP, and WAIT) specify what you want the system to do when the trap matches. For example, specifying ACTION = SVCD means that you want an SVC dump.

The four trap control keywords are **DISABLE**, **ENABLE**, **MATCHLIM**, and **PRCNTLIM**. They control the operation of the trap by indicating whether the trap is active or inactive, how many times the trap should match and produce the desired action before it is automatically made inactive, and what percentage of the system time can be used by **SLIP PER** processing.

The dump and trace tailoring keywords are **ASIDLST**, **LIST**, **SDATA**, **SUMLIST**, and **TRDATA**. The first four keywords are used to tailor a dump, and **TRDATA** is used to tailor trace records. When you request a dump or trace using **ACTION=SVCD**, **ACTION=STDUMP**, **ACTION=TRDUMP**, or **ACTION=TRACE**, the dump and trace tailoring keywords enable you to tailor the contents of the dump or trace record.

The specialized keywords are **END**, **ID**, **DEBUG**, and **RBLEVEL**. **END** marks the end of the **SLIP SET** command. **ID** assigns an identifier to the trap. **DEBUG** is used to diagnose a **SLIP** trap that is not working as you expected; it indicates that you want some trap information recorded each time the trap is checked rather than each time it matches. **RBLEVEL** indicates which request block the system is to use for error detection traps.

### Keywords for SLIP SET Commands

The following keywords, if specified, must immediately follow **SET**. Each indicates a particular type of **PER** trap.

**IF**  
Indicates an instruction fetch **PER** trap

**SA**  
Indicates a storage alteration **PER** trap

**SB**  
Indicates a successful branch **PER** trap

*Note:* If you do not specify one of these three keywords on the **SLIP SET** command, you are setting an error detection (non-**PER**) trap.

**{END}**  
**{E}**

Marks the end of the **SLIP SET** command. If you do not specify this keyword, the system prompts you for additional keywords.

**{ACTION}** = [options]  
**{A}**

Indicates what happens when the trap matches system conditions. The possible actions are: **IGNORE**, **NODUMP**, **NOSUP**, **NOSVCD**, **NOSYSA**, **NOSYSM**, **NOSYSU**, **RECORD**, **RECOVERY**, **STDUMP**, **STRACE**, **SVCD**, **TRACE**, **TRDUMP**, and **WAIT**. Only one **PER** trap with an action other than **IGNORE** can be eligible for checking at any one time.

If you have only one option on the **ACTION=** keyword, you can omit the parentheses.

## SLIP Command

### IGNORE

Indicates that you want the system to resume normal processing when the event occurs. You can use IGNORE to exclude a subset of events from being trapped by a different, more general trap.

#### *Example*

```
ACTION=IGNORE
```

### NODUMP

For error event traps, indicates that you want to suppress any SVC, SYSABEND, SYSUDUMP, or SYSMDUMP dumps requested by the system or by a user program. NODUMP is invalid for PER traps.

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 jobname, other jobs that abend with that completion code still produce dumps. The following example shows a specific SLIP definition with a completion code and jobname:

```
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 one of these contains more than one abend, a second error has occurred. If you need a dump for the second error, disable the SLIP trap that specifies ACTION=NODUMP and rerun the failing job.

#### *Example*

```
ACTION=NODUMP
```

### (nodumptype[,nodumptype]...)

For error event traps, indicates that you want to suppress specific dump types requested by the system or by a user program. You can specify any combination of NOSVCD, NOSYSA, NOSYSU, or NOSYSM. If you specify only one of these options, you do not need the parentheses.

#### NOSVCD

The system is to suppress all SVC dumps requested while processing this error.

#### NOSYSA

The system is to suppress all SYSABEND dumps requested while processing this error.

#### NOSYSU

The system is to suppress all SYSUDUMP dumps requested while processing this error.



**NOSYSM**

The system is to suppress all SYSMDUMP dumps requested while processing this error.

*Example*

```
ACTION=(NOSYSA,NOSYSM)
```

**NOSUP**

For error event traps, indicates that regardless of any dump suppression specifications, you do not want the system to suppress any dumps requested by the system or by a user program (such as suppression of the dump by dump analysis and elimination, DAE). Thus, the NOSUP option ensures that a dump is taken for the trapped event. NOSUP is invalid for PER traps.

*Example*

```
ACTION=NOSUP
```

**RECORD**

For error event traps, indicates that you want to force SYS1.LOGREC recordings for every recovery routine, regardless of what the recovery routine specifies.

*Example* To force a LOGREC recording for an abend 0C6 (FRRs and ESTAEs), enter:

```
SLIP SET,C=0C6,ACTION=RECORD,END
```

**RECOVERY**

Forces PER traps to initiate recovery processing for the interrupted process. If another action is specified, it is done before the recovery processing begins. The recovery processing appears initially as an 06F system completion code, but the recovery routines may change it. You can specify RECOVERY by itself or with IGNORE, SVCD, TRACE, TRDUMP, and WAIT.

For non-PER traps, RECOVERY is invalid.

*Note:* Use the RECOVERY keyword carefully to avoid unexpected results. Before using RECOVERY, you should be thoroughly familiar with the principles of MVS recovery. In particular, make sure that recovery procedures have been established at the point where you are forcing recovery processing. Know what the recovery routines will do under the circumstances in which you are forcing recovery processing.

```
STDUMP [ ,ASIDLST=(id[,id]...)
        ,LIST=(start,end[,start,end]...)
        ,SDATA=(option[,option]...)
        ,SUMLIST=(start,end[,start,end]...) ]
```

Indicates that SLIP system trace records are to be created while the trap is enabled and that an SVC dump is to be scheduled when the trap is disabled or deleted. The SVC dump the system schedules includes the registers and PSW for the current task. ASIDLST, LIST, SDATA, and SUMLIST are used to tailor the dump. Refer to the descriptions of these keywords under ACTION=SVCD.

The STDUMP keyword overrides DAE suppression.

## SLIP Command

*Note:* Although all keywords that are valid on PER traps are allowed on a SLIP trap with ACTION=STDUMP, some keywords require less SLIP processing to find a matching trap. They are:

For IF and SB keywords    ENABLE/DISABLE, ID, MATCHLIM,  
   RANGE/NUCEP/NUCMOD/LPAEP/LPAMOD,  
   ASIDLST, LIST, SDATA, SUMLIST,  
   and END

For SA keywords            ENABLE/DISABLE, ID, RANGE,  
   MATCHLIM, ASIDLST, LIST,  
   SDATA, SUMLIST, and END

If you do not specify MATCHLIM, the trap is disabled after 50 matches.

If you use keywords other than these on the same command with ACTION=STDUMP, SLIP processing uses more system resources. Also, if you do not specify MATCHLIM, there is no limit to the number of matches.

### STRACE

Indicates that at least one SLIP system trace record is to be written when the trap matches. In order for the record to be written, system trace must be active.

*Note:* Although all keywords that are valid on PER traps are allowed on a SLIP trap with ACTION=STRACE, some keywords require less SLIP processing. They are:

For IF and SB keywords    ENABLE/DISABLE, ID, MATCHLIM,  
   RANGE/NUCEP/NUCMOD/LPAEP/LPAMOD,  
   and END

For SA keywords            ENABLE/DISABLE, ID, RANGE,  
   MATCHLIM and END

In this case, if you do not use the MATCHLIM keyword, the trap is disabled after 50 matches.

If you use keywords other than these on the same command with ACTION=STRACE, SLIP processing uses more system resources. Also, if you do not use the MATCHLIM keyword, there is no limit to the number of matches.

SVCD    [ ,ASIDLST = (id[,id]...)  
          ,LIST = (start,end[,start,end]...)  
          ,SDATA = (option[,option]...)  
          ,SUMLIST = (start,end[,start,end]...)

Indicates that the system is to schedule an SVC dump for the current or failing address space when the trap matches. ASIDLST, LIST, SDATA, and SUMLIST 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. The SVCD keyword overrides DAE suppression.

**Caution:** Avoid using a general trap, such as:

```
SLIP SET,COMP=0C4,ACTION=SVCD,END
```

SVC dump processing normally encounters many expected program interrupts, each resulting in a 0C4 completion code. Make sure that your trap does not match for these program interrupts as well, since SLIP then attempts to schedule an SVC dump for each interrupt. Qualify the SLIP trap by using other parameters, such as DATA, PVTMOD, and MATCHLIM. These additional parameters prevent unwanted matches.

**Example**

ACTION=SVCD,SDATA=(SQA,PSA,SUM)

**{ ASIDLST } = (id[,id]...)**  
**AL**

Specifies the hexadecimal address space identifiers of the address space or spaces to be dumped. Up to 15 identifiers can be specified.

You can also specify an address space identifier symbolically. A symbolic address space identifier consists of one of the following:

|               |                                      |
|---------------|--------------------------------------|
| HASID or H    | to indicate home address space       |
| PASID or P    | to indicate primary address space    |
| SASID or S    | to indicate secondary address space  |
| CURRENT or CU | to indicate current address space    |
| LLOC          | to indicate the locked address space |

*Note:* Zero indicates the current address space.

**Example**

ACTION=SVCD,ASIDLST=(0,C)

**{ LIST } = (start,end[,start,end]...)**  
**LS**

Specifies the starting and ending hexadecimal addresses of one or more areas of storage to be included in an SVC dump. The starting and ending addresses can be either direct or indirect. (Refer to “Indirect Addressing Used with SLIP” and “Qualifying Direct or Indirect Addresses to Address Spaces” later in this section.) The starting address must be less than or equal to the ending address.

LIST is similar to SUMLIST (described later) except that the information specified by LIST is collected as part of the scheduled SVC dump while the information specified by SUMLIST is collected as part of the disabled summary dump.

**Example**

ACTION=SVCD,LIST=(152,155,10%+2C%%,+3F)

**{ SDATA } = (option[,option]...)**  
**SD**

Specifies the kind of system control information to be dumped. You can specify any combination of the following, separated by commas: ALLNUC, ALLPSA, CSA, GRSQ, LPA, LSQA, NOALLPSA/NOALL, NOSQA, NOSUMDUMP/NOSUM, NUC, PSA, RGN, SQA, SUMDUMP/SUM, SWA, and TRT.

## SLIP Command

If you don't specify SDATA:

- The default dump options for ACTION=SVCD are ALLPSA, CSA, LPA, NUC, RGN, SQA, SUM, and TRT.
- The default dump options for ACTION=TRDUMP are TRT, NOALLPSA, NOSQA, and NOSUM.
- The default dump options for ACTION=STDUMP are TRT, NOALLPSA, NOSQA, and NOSUM.

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 keyword.

If you specify SDATA, there are no defaults.

### *Example*

```
ACTION=SVCD,SDATA=(SQA,TRT,SUM)
```

**SUMLIST** = (start,end[,start,end]...)  
**SL**

Specifies the starting and ending addresses of one or more areas to be included in a summary dump. The addresses can be either direct or indirect. (Refer to "Indirect Addressing Used with SLIP" and "Qualifying Direct or Indirect Addresses to Address Spaces" later in this section.) SUMLIST is used with ACTION=SVCD or with ACTION=TRDUMP when the SDATA SUM option is specified or is the default. The starting address must be less than or equal to the ending address.

SUMLIST is similar to LIST (described previously) except that the information specified by SUMLIST is collected as part of the disabled summary dump while the information specified by LIST is collected as part of the scheduled SVC dump.

### *Example*

```
ACTION=SVCD,SUMLIST=(152,155,2R%,+3)
```

### **TRACE[,TRDATA = (options)]**

Indicates that GTF is to write at least one SLIP trace record when the trap matches. In order for the record to be built and recorded, GTF with the SLIP option chosen must be active. With TRDATA, you can specify the type and content of the SLIP GTF trace record.

### *Example*

```
ACTION=TRACE
```

**TRDATA** = ( **REGS**[,list] )  
**TD** = ( **STD**[,REGS[,list] ] )  
**list**

Indicates the type and contents of GTF records for ACTION=TRACE or ACTION=TRDUMP.

**STD**

Indicates that GTF is to write a SLIP standard trace record, as described in *MVS/XA Debugging Handbook, Volume I*.

**REGS**

Indicates that the SLIP trace records are to contain the contents of the sixteen general purpose registers at the time of the error or PER interruption.

**list**

Specifies one or more pairs of addresses that indicate one or more areas of storage to be dumped. Each range cannot exceed 65535 bytes. The addresses can be either direct or indirect. For each address pair, the starting address must be less than or equal to the ending address. (Refer to “Indirect Addressing Used with SLIP” and “Qualifying Direct or Indirect Addresses to Address Spaces” later in this section.)

**Example**

```
ACTION=TRACE, TRDATA=( STD, REGS, 152, 155 )
```

```
TRDUMP [ ,ASIDLST = (id|,id|...)
        ,LIST = (start,end|,start,end|...)
        ,SDATA = (option|,option|...)
        ,SUMLIST = (start,end|,start,end|...)
        ,TRDATA = (options) ]
```

Indicates that trace records are to be created while the trap is enabled and that an SVC dump is to be scheduled when the trap is disabled or deleted. The SVC dump the system schedules includes the registers and PSW for the current or failing task. SDATA, LIST, ASIDLST, and SUMLIST are used to tailor the dump and TRDATA is used to tailor the trace records. Refer to the descriptions of these keywords under ACTION=SVCD and ACTION=TRACE.

The TRDUMP keyword overrides DAE suppression processing.

**Example**

```
ACTION=TRDUMP, TRDATA=( STD, REGS ), SDATA=( TRT, SQA )
```

**WAIT**

Indicates the system is to stop and display the following information in message IEE837I:

- SLIP id
- Type of trap and related information:
  - RTM1 - the SDWA address
  - RTM2 - the RTM2WA address
  - MEMTERM - the ASCB address
  - PER - the PER code and address
- PSW at the time of error or PER event
- Contents of control registers 3 and 4 that indicate which address spaces are primary (PASID) and secondary (SASID)
- Contents of general registers 0 to 15

## SLIP Command

The system instructs you to enter "U" 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. (Taking a stand-alone dump, for example, prevents a restart.)

If the system cannot display the message (for example, if the console is unavailable), the system enters a restartable wait state (code 01B). You can then restart the system, provided that a restart is possible, as mentioned in the preceding paragraph.

*Note:* When the SLIP command is entered from a TSO terminal, the system does not accept this parameter.

### *Example*

```
ACTION=WAIT
```

**{ADDRESS } = (start[,end])**  
**{AD }**

For an error detection trap, ADDRESS specifies the virtual address at which or range of addresses within which the error must occur. Each address is 1 to 8 hexadecimal digits.

For a storage alteration PER trap, ADDRESS specifies the virtual address at which or range of addresses within which the instruction that causes the storage alteration must reside.

ADDRESS is invalid for an instruction fetch or successful branch PER trap.

The starting address must be less than or equal to the ending address.

### *Example*

```
ADDRESS=(CD300,CD400)
```

**{ASID } = (id[,id]...)**  
**{AS }**

ASID specifies the address space that must be in control when the error or PER interruption occurs. Each address space identifier is 1 to 4 hexadecimal digits, and you can specify up to 16 identifiers.

For non-IGNORE PER traps, specifying ASID indicates the address spaces in which PER monitoring is to be active.

If you do not specify ASID on non-IGNORE PER traps, PER monitoring is to be active in all address spaces unless you limit it by specifying JOBNAME.

If you specify both ASID and JOBNAME, one of the address space identifiers must be the one in which the job is running or the trap will not match.

Specifying HOME on the MODE keyword also limits PER monitoring. For non-IGNORE PER traps, 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.

This keyword is useful when monitoring storage alteration of an address range that falls in the private area.

**Example**

ASID=( 1 , 7 , 1A )

**{ASIDSA}** = (id[,id]...)  
**{ASA}**

For storage alteration PER traps, ASIDSA specifies the address space of the storage that is being altered. Each address space identifier is 1 to 4 hexadecimal digits, and you can specify up to 16 identifiers. Note that the ASID of the address space where the storage is being altered can differ from the ASID of the address space from which instructions are fetched.

You can also specify an ASID symbolically. A symbolic ASID consists of one of the following:

|               |                                     |
|---------------|-------------------------------------|
| HASID or H    | to indicate home address space      |
| PASID or P    | to indicate primary address space   |
| SASID or S    | to indicate secondary address space |
| CURRENT or CU | to indicate current address space   |

**Example**

ASIDSA=( 5 , 3A , 17B , 24E )

**{COMP}** = code1    [ **{REASON}** = code2 ]  
**{C}**                    [ **{RE}** ]

Specifies a system or user completion code and, optionally, a reason code to be associated with the error.

**code1**

Specifies a system or user completion code to be associated with the error.

For a system completion code, the form must be *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 *x*'s in any position.

For a user completion code, the form must be *Uxxxx*, where *U* indicates a user code and the *x*'s represent decimal numbers. By leaving one or more of the *x*'s in, you can specify a set of codes. For example, *U102x* means U1021, U1022, U1023, and so forth.

# SLIP Command

## 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, 57D, 6FC, 700, 72A, A00, B00, and E00.*

*Each of these codes was originally a program check (code 0C4) that has been converted to a new value by the system. 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 such abend completion codes as 201, 202, 402, and 702, the SLIP action might not be taken. In certain paths, each of these codes was originally a program check, as described in Note 1. In other paths, the abend was 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 processing 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 instruction, 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 keyword.*
5. *See ACTION=SVCD for a cautionary note about specifying COMP=0C4 with ACTION=SVCD.*
6. *SLIP processing cannot trap the 822 completion code because RTM does not get control when the event that causes the code occurs.*

## code2

Specifies a user or system reason code to be associated with the error. (You cannot specify the REASON keyword without the COMP keyword.)

For the reason code, enter one to eight hexadecimal digits. If the code is less than eight 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 seven of the eight digits. For example, REASON=44XXXX means any reason code that begins with the digits 0044; the last four digits can be any hexadecimal value. You can use x's in any position.

## Note:

To match the REASON keyword, the reason code must have been specified via the REASON parameter of the ABEND, SETRP, or CALLRTM macro.

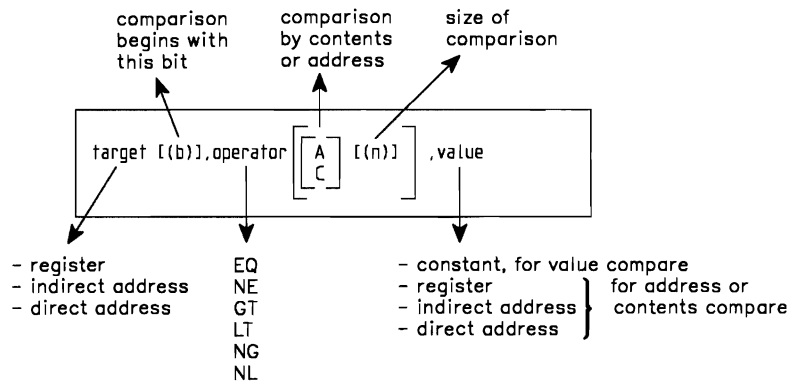
## Example

```
COMP=U123x, REASON=8
```



**DATA = (triplet) or (group of triplets)**

where each **triplet** consists of a target, operator, and value, as follows:



Each triplet specifies that the contents of a target location are to be logically compared to a value. 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 (they 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.

### 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 (virtual address) of 1 to 8 hexadecimal digits. ( See "Qualifying Direct or Indirect Addresses to Address Spaces" later in this section.)
- A GPR in the form xR where x is 0 – 15.
- An indirect address (See "Indirect Addressing Used with SLIP" and "Qualifying Direct or Indirect Addresses to Address Spaces" later in this section.)

### b – binary compare

If specified, *b* modifies the target address by indicating the bit position where a binary comparison is to start. For registers, *b* can be 0 – 31. The starting bit position plus the bit size for comparison must not exceed 31.

# SLIP Command

## 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 an invalid combination of binary compare (1R(0)) and address compare (EQA). In a four-byte compare, bit 0 is ignored; bits 1 - 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 - 8 bits, with the default being one bit. Otherwise, the range is 1 - 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 is to be compared. If you do not specify *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 eight 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 four bytes. The value can be one to eight 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 four bytes, and the value is right-justified (for example, 2R,EQ,4 and 2R,EQ,00000004 are equivalent).

If you specify A or C as part of the operator, the value can be:

- A direct address (virtual address) of 1 to 8 hexadecimal digits
- A GPR in the form xR where x is 0 – 15
- An indirect address

**Example 1:** The following expression is true if general purpose register 2 is 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 that register 3 is pointing to.

**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)
```

## SLIP Command

**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?)
```

### DEBUG

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. GTF and its trace option for SLIP records must be active.

DEBUG indicates that trace records are to be created each time the SLIP trap is tested. Each DEBUG record contains SLIP information plus two bytes of match/no match bit indicators. Each bit corresponds to a possible test made to determine a match for a trap. If the bit is 1, the match failed. If the bit is 0, the match was successful or not performed. (After one failure, no further tests are made.)

For a description of the DEBUG trace record and the bit indicators, see *Debugging Handbook*.

#### **Example**

```
DEBUG
```

**[DISABLE ]**  
**[D**

Indicates that the trap set is to be initially inactive (ineligible for checking). When neither DISABLE nor ENABLE is specified, ENABLE is the default.

#### **Example**

```
DISABLE
```

**[ENABLE ]**  
**[EN**

Indicates that the trap defined is to be initially active (eligible for checking). When neither ENABLE nor DISABLE is specified, ENABLE is the default.

#### **Example**

```
ENABLE
```

**{ERRTYP}** = (type[,type]...)  
**{ER}**

Specifies one or more error events. If you specify ALL or more than one error type, the occurrence of any one of them satisfies the match test for this keyword. You can specify one or more of the following error events:

|         |   |
|---------|---|
| PROG    | Program check interruption  |
| REST    | Restart interruption  |
| SVCERR  | SVC error (issuing an SVC while locked, disabled, or in SRB mode) |
| PGIO    | Paging I/O error  |
| DAT     | Dynamic address translation error                                 |
| MACH    | Software error caused by machine check                            |
| MEMTERM | Abnormal address space termination                                |
| ABEND   | Task-issued SVC 13  |
| ALL     | A short method of entering all of the above error conditions      |

**Example**

ERRTYP=(MACH, DAT)

**ID = xxxx**

Specifies 1 to 4 alphanumeric characters as a trap identifier. If ID is not indicated in SLIP SET commands, the system assigns a unique id.

**Example**

ID=PER1

**{JOBNAME}** = jobname  
**{J}**

Specifies the 1 to 7 character name of the time-sharing user or the 1 to 8 character name of the job or started task that must be in control for the JOBNAME match test to be satisfied.

For non-IGNORE PER traps, JOBNAME limits PER monitoring to the address spaces in which the specified job runs. If the unit of work executes 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. If neither JOBNAME nor ASID is specified for a non-IGNORE PER trap, PER monitoring is active in all address spaces in the system. If both JOBNAME and ASID are specified, one of the address space identifiers must be the one in which the job is running, otherwise the trap will not match.

For non-IGNORE PER traps, 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.

**Example**

JOBNAME=D10AXX1

## SLIP Command

**{JSPGM } = jobstep-program-name**  
**{JS }**

Specifies the 1 to 8 character name of the job step program that must be in control to satisfy the JSPGM match test. On non-PER traps, JSPGM causes a no-match if the error type being processed is MEMTERM.

### *Example*

```
JSPGM=IFOX00
```

**LPAEP = (name[,start[,end]])**

Monitors modules in the link pack area, given an entry point within the module.

*name* is the 1 to 8 character entry point name or alias. If you use an asterisk as the last character of the module name, SLIP interprets the asterisk as X'C0'. (Certain SVC module names end with the character X'C0'.) If *name* is used alone, the range of monitoring is from the entry point or alias you specify to the end of the module. *start* and *end* are offsets from the entry point or alias; they indicate the range where the monitoring is to occur. The starting offset must be less than or equal to the ending offset. For more information, see the notes under the RANGE keyword. If just a starting offset is specified, the range consists of that single address.

For a non-PER trap and an IF or SB PER trap, LPAEP establishes the range of addresses to be monitored.

For an SA PER trap, LPAEP indicates that the storage alteration must be caused by an instruction within the specified range.

### *Example*

```
LPAEP=IEECB907, ...
```

**{LPAMOD } = (name[,start[,end]])**  
**{L }**

Monitors modules in the link pack area, given the name of the module.

*name* is the 1 to 8 character name that identifies the beginning of the module. If you use an asterisk as the last character of the module name, SLIP interprets the asterisk as X'C0'. If the name is the alias of a load module, SLIP processing uses the load module name. If *name* is used alone, the range of monitoring is the entire module. *start* and *end* are the offsets within the module that specifically indicate the range where the monitoring is to occur. The starting offset must be less than or equal to the ending offset. For more information, see the notes under the RANGE keyword. If just a starting offset is specified, the range consists of that single address.

For a non-PER trap and an IF or SB PER trap, LPAMOD establishes the range of addresses to be monitored.

For an SA PER trap, LPAMOD indicates that the storage alteration must be caused by an instruction within the specified range.

### *Example 1*

```
LPAMOD=( IEAVTXYZ, 2C)
```

**Example 2:** To monitor events in a load module for occurrences of SVC60, enter:

```
LPAMOD=IGC0006*
```

**{MATCHLIM}** = *m*  
**{ML}**

Specifies that the SLIP trap is to be disabled after *m* matches, where *m* is an integer from 1 to 65535. If you don't specify MATCHLIM for a PER trap with ACTION=SVCD, the trap is disabled after one match. If you don't specify MATCHLIM for a PER trap with ACTION=STRACE, the trap is disabled after 50 matches. If you don't specify MATCHLIM for any other type of trap, the trap can match any number of times.

**Example**

```
MATCHLIM=50
```

**{MODE}** = (mode[,mode]... [ANY ])  
**{M}** [EVERY ])

Specifies the mode the system must be in for the trap to match. You can specify more than one mode and you can indicate that any of them (ANY) will be sufficient to cause a match or you can indicate that every one specified (EVERY) must be present. The modes you can specify are:

|            |  |
|------------|--|
| 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                                  |
| <u>ALL</u> | All of the above except HOME                           |

**Notes:**

1. Specifying the LLOC, LOCK, or ALL option of the MODE keyword, 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. 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.

## SLIP Command

**Invalid Modes:** The following mode specifications are invalid and cause an error message to be issued:

MODE=(EVERY) – not possible because some modes are mutually exclusive. (ALL is assumed).  
MODE=(ANY) – Must specify one or more modes to go with ANY.  
MODE=(ANY,EVERY) – Contradictory.

The RECV mode cannot be chosen when MODE is specified on a PER trap.

### Example

```
MODE=(LLOC,SRB,EVERY)
```

**{NUCMOD}** = (name[,start[,end]])  
**{NUCEP}**  
**{N}**

Monitors modules in the nucleus. There is no difference between the NUCMOD and NUCEP keywords.

*name* is the 1 to 8 character name that identifies the beginning of the area to be monitored. If *name* is used alone, the range of monitoring is from *name* to the end of the module. *start* and *end* are the offsets within the module that specifically indicate the range where monitoring is to occur. The starting offset must be less than or equal to the ending offset. For more information, see the notes under the RANGE keyword. If just a start offset is specified, the error must occur at that offset. If offsets are not specified, the error can occur anywhere within the nucleus module.

For an IF or SB PER trap, NUCMOD (NUCEP, or N) establishes the range of addresses to be monitored.

For an SA PER trap, NUCMOD (NUCEP, or N) indicates that the storage alteration must be caused by an instruction within the nucleus module or within the module in the range of offsets specified by start and end.

If you use an asterisk as the last character of the module name, SLIP interprets it as X'C0'.

### Example

```
NUCMOD=(IEAVTRTS,C4)
```

**{PRCNTLIM}** = p  
**{PL}**

Specifies, for PER traps, 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. The value computed to test PRCNTLIM is an approximation. If you don't specify PRCNTLIM, the default is 10.

### Example

```
PRCNTLIM=20
```



## **PVTEP = (name[,start[,end]])**

Monitors modules in the private area, given an entry point or alias within the module.

*name* is a 1 to 8 character entry point within the module. If you use an asterisk as the last character of the module name, SLIP interprets the asterisk as X'C0'. If *name* is used alone, the range of monitoring is from the entry point or alias you specify to the end of the module. *start* and *end* are the offsets from the entry point or alias; they indicate the range where the monitoring is to occur. The starting offset must be less than or equal to the ending offset. For more information, see the notes under the RANGE keyword. If just a starting offset is specified, the range consists of that single address.

For a non-PER trap and an IF or SB PER trap, PVTEP establishes the range of addresses to be monitored.

For an SA PER trap, PVTEP indicates that the storage alteration must be caused by an instruction within the specified range.

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.

### **Example:**

```
PVTEP=(MYEP,10,40)
```

## **PVTMOD } = (name[,start[,end]])** **P**

Monitors modules in the private area, given the name of the module.

*name* is the 1 to 8 character name that identifies the beginning of the module. If you use an asterisk as the last character of the module name, SLIP interprets the asterisk as X'C0'. If the name is the alias of a load module, SLIP processing uses the load module name. If *name* is used alone, the range of monitoring is the entire module. *start* and *end* are the offsets within the module that specifically indicate the range where the monitoring is to occur. The starting offset must be less than or equal to the ending offset. If just a starting offset is specified, the error must occur at that offset.

For a non-PER trap and an IF or SB PER trap, PVTMOD establishes the range of addresses to be monitored.

For an SA PER trap, PVTMOD indicates that the storage alteration must be caused by an instruction within the specified range.

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.

### **Example**

```
PVTMOD=(MYMOD,1C,1F)
```

## SLIP Command

**{RANGE } = (start[,end])**  
**{RA }**

For PER traps, specifies the starting and ending hexadecimal addresses of the virtual storage area to be monitored. If only one address is specified, a one byte range is assumed. If the starting address is greater than the ending address, the addresses wrap around. RANGE is invalid for non-PER traps. RANGE cannot be specified on an ACTION=IGNORE storage alteration PER trap.

The following notes apply to all SLIP keywords that have *start[,end]* options to indicate a range of offsets within a module.

### Notes:

1. *Consider the range carefully on any PER trap. A wide range could cause performance degradation 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 (which 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 RANGE, LPAMOD|EP, NUCMOD|EP, and PVTMOD|EP keywords. 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, then, 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 keywords, 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.*

*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.*

### Example

RANGE=(600,700)

$$\left. \begin{array}{l} \text{RBLEVEL} \\ \text{RB} \end{array} \right\} = \left\{ \begin{array}{l} \text{ERROR} \\ \text{PREVIOUS} \\ \text{NOTSVRB} \end{array} \right\}$$

Indicates which RB contains the registers for use in resolving indirect addresses and the PSW for use by LPAMOD, PVTMOD, ADDRESS, and MODE. This keyword applies only to unlocked task mode errors and is not valid for PER traps.

## ERROR

The PSW used is obtained from the RB prior to the SVC 13 (ABEND) RB. The registers are obtained from the SVC 13 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.

### *Example*

```
RBLEVEL=NOTSVRB
```

## 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, A=NOSYSA,END
```

*Example 3:* The following 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.

```
SL 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,SB,ENABLE, ID=PER1,LPAMOD=(MOD01,108,4FC), JOBNAME=JOBX,
ACTION=TRACE, MATCHLIM=20,END
```

*Example 5:* The following 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.

```
SL SET,SA,ENABLE, ACTION=SVCD, RANGE=CD3010,END
```

## SLIP Command

*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,EN,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.

## Modifying a SLIP Trap

Use the following form of the SLIP command to modify an existing SLIP trap.

|  |     |  |  |
|--|-----|--|--|
| $\left\{ \begin{array}{l} \text{SLIP} \\ \text{SL} \end{array} \right\}$ | MOD | $\left\{ \begin{array}{l} ,\text{ENABLE} \\ ,\text{EN} \\ ,\text{DISABLE} \\ ,\text{D} \end{array} \right\}$ | $\left\{ \begin{array}{l} ,\text{ALL} \\ ,\text{ID=xxxx} \end{array} \right\}$ |
|--|-----|--|--|

### MOD

This SLIP command modifies one or all existing SLIP traps.

#### ENABLE or EN

The specified SLIP trap is to be made active.

#### DISABLE or D

The specified SLIP trap is to be made inactive.

#### 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.

**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 the consequences for your system.**

#### ID = xxxx

Only the SLIP trap associated with the identifier xxxx is to be modified.

Where asterisks replace any or all of the four characters of the identifier, all SLIP traps whose IDs match the non-asterisk characters in the identifiers are to be modified. If you specify less than four 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 and enable and disable them as a group.

*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 a SLIP Trap

Use the following form of the SLIP command to delete a SLIP trap.

|  |  |
|--|--|
| $\left\{ \begin{array}{l} \text{SLIP} \\ \text{SL} \end{array} \right\}$ | DEL $\left\{ \begin{array}{l} ,\text{ALL} \\ ,\text{ID=xxxx} \end{array} \right\}$ |
|--|--|

### DEL

This SLIP command removes one or all SLIP traps from the system.

### ALL

Every SLIP trap in the system is to be deleted. To find out what SLIP traps are in the system, issue DISPLAY SLIP.

**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 the consequences for your system.

### ID = xxxx

Only the SLIP trap associated with the identifier *xxxx* 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.

*Example:* The following SLIP command is used to delete the SLIP trap associated with identifier 0008. This SLIP trap cannot be reactivated by the SLIP MOD command.

```
SLIP DEL, ID=0008
```

# SLIP Command

## SLIP Keyword Summary

### The required keywords are:

For all SLIP commands:

- SET, MOD, or DEL

For all SLIP SET commands:

- END

For SLIP SET commands with IF or SB:

- RANGE, LPAEP, LPAMOD, NUCEP, NUCMOD, PVTEP, or PVTMOD (except when ACTION=IGNORE is specified)

For SLIP SET commands with SA:

- RANGE (except when ACTION=IGNORE is specified)

For SLIP MOD or SLIP DEL commands:

- ID or ALL

For SLIP MOD commands only:

- ENABLE or DISABLE

### The positional keyword restrictions are:

- SET, MOD, or DEL must immediately follow SLIP.
- IF, SB, or SA, if specified, must immediately follow SET.

### The default keywords are:

For SLIP SET commands with IF, SB, or SA:

- ENABLE
- ACTION=SVCD
- PRCNTLIM=10
- MATCHLIM=1 (only for ACTION=SVCD)
- MATCHLIM=50 (only for ACTION=STRACE and ACTION=STDUMP, as described in the notes that accompany the descriptions of the STRACE and STDUMP operands)

For SLIP SET commands without IF, SB, or SA:

- ENABLE
- ACTION=SVCD
- RBLEVEL=ERROR

For ACTION = SVCD:

- SDATA = (ALLPSA, CSA, LPA, NUC, RGN, SQA, SUM, TRT)
- MATCHLIM = 1 (only for SLIP SET commands with IF, SB, or SA)

For ACTION = TRACE:

- TRDATA = STD trace record

For ACTION = TRDUMP:

- TRDATA = STD trace record
- SDATA = TRT, NOALLPSA, NOALLSQA, NOSUM

For ACTION = STRACE and ACTION = STDUMP:

- MATCHLIM = 50 (For more information, see the notes that accompany the description of the STRACE and STDUMP operand.)
- SDATA = TRT, NOALLPSA, NOALLSQA, NOSUM

### Indirect Addressing Used with SLIP

An indirect address is the address of a location or general purpose register that contains another address. You can use indirect addressing with the following SLIP command keywords: DATA, LIST, 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 **register**, in the form nR, where *n* is a decimal number from 0 to 15.
3. An **indirection indicator**, which can be a percent sign (%) or a question mark (?). 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.
4. A **displacement**, which begins with a plus or minus sign and consists of 1 to 4 hexadecimal digits. The maximum displacement allowed is 7FFF.

# SLIP Command

In the expression

12C%+4%+8%+C

12C is the 24-bit direct address,  
% are the indirection indicators,  
and +4, +8, and +C are displacements.

In the expression

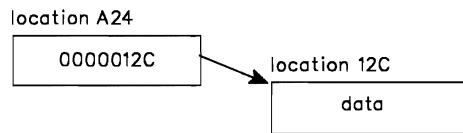
2R??+4?

2R is the register (general purpose register 2),  
?? and ? are the indirection indicators (as many as  
necessary can be used), and +4 is the displacement.

To refer to data when the address of the data is located at A24, specify:

A24%

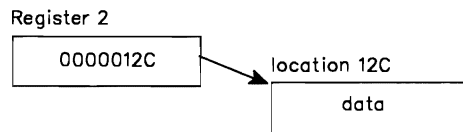
Graphically:



To refer to data when the address of the data is in general purpose register 2, specify:

2R%

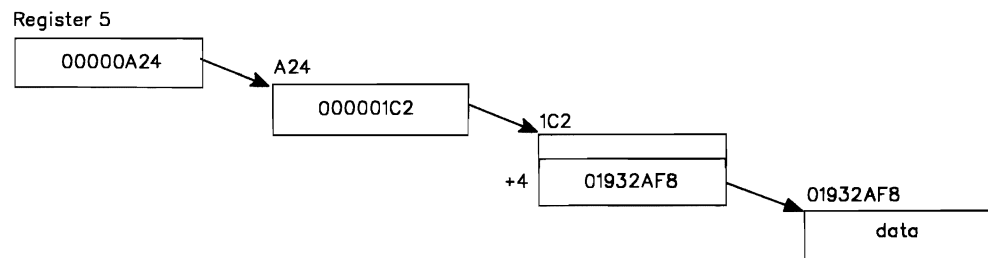
Graphically:



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:



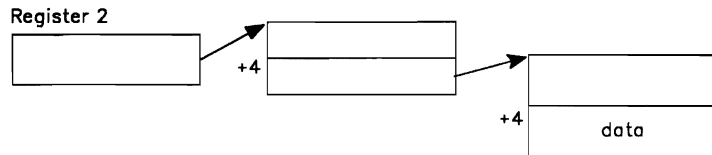


## Base/Displacement Shorthand

Instead of entering the complete direct or indirect address, you can use a form of shorthand for any address after the first. The first address establishes a base address and is written out completely. Subsequent addresses are written as plus or minus displacements from the base.

$2R\%+4,+4$   
 base displacement

Graphically:



The base consists of everything in the indirect address **except the last displacement**. (If no displacement is specified, SLIP assumes a displacement of zero.)

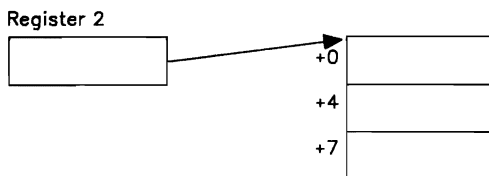
Short Method

Regular Method

$2R\%+4,+7$   
 base displacement 1 displacement 2

$2R\%+4, 2R\%+7$

Graphically:



The preceding example defines a four-byte range, from  $2R\% + 4$  through  $2R\% + 7$ . The base is  $2R\%$  and SLIP adds each displacement to that base.

When you are using shorthand to determine 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 define a four-byte range as:

$2R\%+4,+3$

because SLIP interprets the range as  $2R\% + 4$  to  $2R\% + 3$ . An error condition exists because the ending address is less than the starting address.

# SLIP Command

## Indirect Addressing and the DATA Parameter

The following discussion applies to the DATA parameter 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,

$\underbrace{2R+4}_{\text{target}}$ , EQ, A24,  $\underbrace{2R+8}_{\text{target}}$ , NE, B66  
operator value target operator value

is written using shorthand form as

$\underbrace{2R+4}_{\text{target}}$ , EQ, A24, +8, NE, B66  
operator value target operator value

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 and the LIST, SUMLIST, and TRDATA Parameters

When indirect addressing is used with the LIST, SUMLIST, 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:

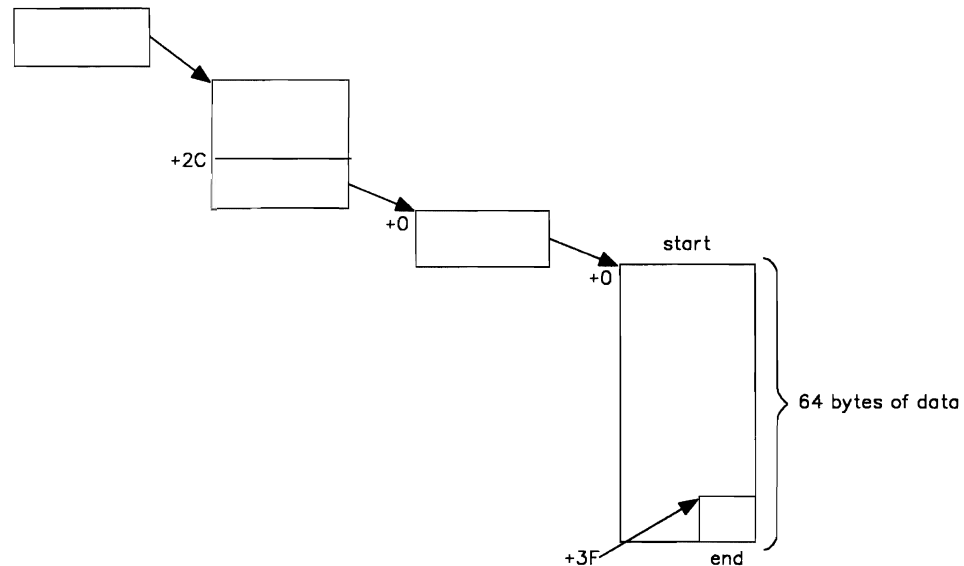
$\underbrace{10+2C\%}_{\text{start}}$ ,  $\underbrace{10+2C\%+3F}_{\text{end}}$

The same example in shorthand is:

$\underbrace{10+2C\%}_{\text{start}}$ ,  $\underbrace{+3F}_{\text{end}}$

Graphically:

Location 10 (hexadecimal)



## Qualifying Direct or Indirect Addresses to Address Spaces

You can explicitly or symbolically qualify each direct or indirect address to an address space associated with the DATA, LIST, SUMLIST, and TRDATA keywords of the SLIP command. The format of the address space qualifier is:

`asid.addr`

where

### **asid**

Is an explicit or symbolic address space qualifier. An explicit address space qualifier consists of a hexadecimal ASID number with a maximum of four digits. A symbolic address space qualifier consists of one of the following:

|               |                                     |
|---------------|-------------------------------------|
| HASID or H    | to indicate home address space      |
| PASID or P    | to indicate primary address space   |
| SASID or S    | to indicate secondary address space |
| CURRENT or CU | to indicate current address space   |

If you do not specify an address space qualifier, SLIP processing uses the current address space.

The period separates the ASID qualifier from the direct or indirect address.

### **addr**

Is either a direct address of one to eight 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)
```



**SUB= subsystemname**

The name of the subsystem that is to select the job for processing. If the parameter is not specified, the system uses the primary job entry subsystem. The subsystem specified must be active and must support job selection.

**keyword = option**

Any appropriate keyword specified to override the corresponding parameter in the cataloged procedure.

*Notes:*

1. *If you specify an option within apostrophes, for example, "DUMMY," use uppercase characters within the apostrophes. START command processing does not convert the lower case characters to uppercase. Thus, entering lower case letters might cause a JCL error or an abend.*
2. *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 terminates and returns message IEF640I.*
3. *If you are overriding a symbolic parameter, do not use any of the DD keywords. For example, do not use UNIT= to override the devicename positional parameter in the cataloged procedure.*

*Example 1:* To start the job named SCRATCH, enter:

```
s scratch
```

*Example 2:* To start JES2, enter:

```
s jes2
```

## Reading a Job from a DASD Device or Tape

Using the same form of the START command that starts a job from the console, you can read a JES2 job or a JES3 internal reader job from a magnetic tape or direct access device. The cataloged procedure named procname must be an internal reader procedure. For a description of the internal reader and how to catalog an internal reader procedure, see the subsection on the internal reader in the appropriate JES2 or JES3 publication.

## Starting a Writer

Use the following form of the START command to start an external writer.

|                            |   |
|----------------------------|---|
| <pre>{ START } { S }</pre> | <pre>procname[.identifier][,devicename][,volumeserial][,classes] [ ,keyword=option[,keyword=option]...]</pre> |
|----------------------------|---|

**procname**

The name of a cataloged procedure, residing in SYS1.PROCLIB, that defines the writer to be started. The name can be either the IBM-assigned name (XWTR) or a user-assigned name.

# START Command

## 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.

## devicename

The device number, such as 280, or the device type, such as 3211, of the writer to be started.

## 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. If you are overriding a symbolic parameter, do not use any of the DD keywords. For example, do not use UNIT= to override the devicename positional parameter in the cataloged procedure.

The commas preceding devicename, volumeserial, and classes can be omitted when the positional parameters following them are not entered.

*Example:* To start an external writer, with the identifier A, enter:

```
S XWTR.A,282
```

## Starting GTF

To start the generalized trace facility, use the following form of the START command and the following parameters.

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{START} \\ \text{S} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{GTF} \\ \text{procname} \end{array} \right\}$ [.identifier][,devicename][,volumeserial]<br><br>[ , ( [ MODE= $\left\{ \begin{array}{l} \text{INT} \\ \text{EXT} \end{array} \right\}$ ] [,BUF=nnn][,TIME=YES][,DEBUG=YES] ) ]<br><br>[ ,MEMBER=xxxxxxxx ][,REGION=nnnnK]<br><br>[ ,keyword=option[,keyword=option]... ] |
|--|---|

## GTF

The name of the IBM-supplied cataloged procedure that invokes GTF.

## procname

The name of the user-written cataloged procedure that invokes GTF.

**identifier**

The user-determined name identifying this specific GTF session. If you do not specify an identifier, the system automatically uses the procname as the identifier.

**devicename**

The device number, such as 280, or 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.

**volumesimal**

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=EXT**

Trace data is to be maintained in an external data set.

**BUF=nnn**

The number (10-255) of buffers to be formatted if an abnormal termination occurs or if a dump is taken.

**TIME=YES**

Each logical trace record is to be time-stamped. If this parameter is omitted, individual records are not to be time-stamped.

**DEBUG=YES**

GTF terminates 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=xxxxxxx**

The member of SYS1.PARMLIB to be accessed by this invocation of GTF. If this parameter is omitted, GTFPARM is used.

**REGION=nnnnK**

The maximum size of the GTF address space in bytes. You can specify any number from 832K to 2880K.

**keyword = option**

Any appropriate keyword specified to override the corresponding keyword in the cataloged procedure. If you are overriding a symbolic parameter, do not use any of the DD keywords. For example, do not use UNIT= to override the devicename positional parameter in the cataloged procedure.

*Example 1:* This example starts a GTF session with the identifier EXAMPLE and with trace data maintained in the GTF address space. The zyx keyword is entered to override a symbolic parameter defined in the cataloged procedure.

```
S GTF.EXAMPLE , , , (MODE=INT) , zyx=NULLFILE
```

## START Command

*Example 2:* This example starts a GTF session with trace data recorded on the non-labeled tape on device 282. Each trace record is time-stamped. Twenty buffers are formatted if a dump is taken.

```
S GTF,282,,(TIME=YES,BUF=20),LABEL=(,NL)
```

*Example 3:* 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 DA,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 TSO/VTAM Time-Sharing

Use the following form of 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.

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{START} \\ \text{S} \end{array} \right\}$ | <pre>procname[.identifier][,devicename][,volumeserial]       [, ( [ MEMBER= { name } ] [,USERMAX=nnnnn] ) ]       [,keyword=option[,keyword=option]...]</pre> |
|--|---|

### **procname**

The name of the cataloged procedure that starts TSO/VTAM time-sharing.

### **identifier**

The user-determined name identifying this specific time-sharing session. If you do not specify an identifier, the system automatically uses the procname as the identifier.

### **devicename**

The device number or 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=IEFRDRE. If the PRINTOUT DD statement specifies a SYSOUT device, the parameter values are listed on that device.

### **volumeserial**

If devicename 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 SYS1.PARMLIB that contains TSO/VTAM time-sharing parameters.

#### **nn**

A two-digit decimal number that forms the suffix for specifying the name of the member in SYS1.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 SYS1.PARMLIB member that contains TSO/VTAM time-sharing parameters, a value of 40 is used.

**keyword = option**

Any appropriate keyword specified to override the corresponding keyword in the cataloged procedure. If you are overriding a symbolic parameter, do not use any of the DD keywords. For example, do not use UNIT= to override the devicename positional parameter in the cataloged procedure.

If more than one SYS1.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 SYS1.PARMLIB (that contains TSOKEY00) is.
5. The default values in the TCAS program if neither a member nor a SYS1.PARMLIB is specified.

## Starting the LNKLST Lookaside (LLA) Address Space

Use the following form of the START command to start the LLA address space. LLA processing enables the system to obtain LNKLST directory entries from the LLA directory in storage instead of making time-consuming searches of PDS directories.

|                            |               |
|----------------------------|---------------|
| <pre>{ START }<br/>S</pre> | LLA, SUB=MSTR |
|----------------------------|---------------|

**LLA**

Invokes the LLA procedure and creates the LLA address space.

The START LLA command is initially issued by the system from the SYS1.PARMLIB member IEACMD00. Use this command if LLA terminates, either because of an error or as a result of a STOP LLA command.

**SUB = MSTR**

Indicates that the system is to start the LLA address space from master's address space. If you omit this parameter, the system starts the LLA from the JES address space. The resulting dependency on JES causes the LLA to terminate when JES terminates.

*Note:* System performance is degraded if LLA terminates.

## STOP Command

## STOP Command

Use the STOP command to:

- Stop a job in execution
- Stop a writer
- Stop the generalized trace facility (GTF)
- Stop TSO/VTAM
- Terminate the LNKLST lookaside (LLA) address space

The complete syntax for the STOP command is:

|   |  |
|---|--|
| $\left\{ \begin{array}{l} \text{STOP} \\ \text{P} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{jobname} \\ \text{[procname.]identifier} \\ \text{LLA} \end{array} \right\}$ |
|---|--|

### **jobname**

The name of the job to be stopped. The STOP command stops a job only if the programmer has coded a stop routine in his program.

### **procname**

The name of the cataloged procedure that was used in the START command.

### **identifier**

One of the following:

- The identifier of the writer to be stopped. You can also use the device number of the writer. If the writer is waiting, it stops immediately; if it is working, it stops after finishing the current data set.
- The identifier that was specified when GTF was started or the device number of the GTF trace data set. If GTF was started in internal mode and no identifier was specified, GTF is the identifier. If you are uncertain of the identifier or device number, use the DISPLAY A,LIST command to display active jobs.
- The identifier that was specified in the START command that started TSO/VTAM time-sharing. (Note that, if you did not use an identifier with the START command, the system automatically used the procname you supplied in the START command as the identifier.)

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 identifier to the START command procname; use that same unique identifier on the STOP command.

If you did not use an identifier with the START command, the system automatically uses as the identifier the procname you supplied in the START command.

## 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 IKTO10D 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 terminates. Otherwise, the following command has no effect.

```
p sysda
```

*Example 2:* To stop 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 EXAMPLE, enter:

```
P EXAMPLE
```

### Terminating the LNKLST Lookaside (LLA) Address Space

Use the following form of the STOP command to terminate the LLA address space.

|   |     |
|---|-----|
| $\left\{ \begin{array}{l} \text{STOP} \\ \text{P} \end{array} \right\}$ | LLA |
|---|-----|

#### **LLA**

Terminate the LLA address space.

Stopping LLA might be done when two or more systems have shared access to the same LNKLST directories and modifications are to be made to these shared LNKLST directories. After stopping LLA and modifying the shared LNKLST 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 guarantees that the updates will be made simultaneously on all of the sharing systems.

## STOPMN Command

## STOPMN Command

Use the STOPMN command to stop the continual display of job status, data set status, or time-sharing user session activity initiated in response to the MONITOR command or MONITOR parameters on the CONSOLE and INIT statements in the CONSOLxx member of SYS1.PARMLIB.

The complete syntax for the STOPMN command is:

|  |   |
|--|---|
| $\left\{ \begin{array}{l} \text{STOPMN} \\ \text{PM} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{JOBNAMES} [ , \text{L}=\text{cc} ] \\ \text{DSNAME} \\ \text{SPACE} \\ \text{STATUS} [ , \text{L}=\text{cc} ] \\ \text{SESS} [ , \text{L}=\text{cc} ] \end{array} \right\}$ |
|--|---|

### JOBNAMES

The jobname display specified in the MONITOR JOBNAMES command is to be stopped.

### DSNAME

The display of nontemporary data set names specified in the MONITOR DSNAME command is to be stopped.

### SPACE

The display of available space on direct access volumes specified in the MONITOR SPACE command is to be stopped.

### STATUS

The display of data set names, volume serial numbers, and status specified in the MONITOR STATUS command is to be stopped.

### SESS

The display of time-sharing user identifiers is to be stopped.

### L=cc

The console where the display is to appear unless routing instructions are in effect; see the MSGRT command in this chapter. If you omit the keyword, the console on which you enter the PM command is assumed.

*Example:* To discontinue the display of job name information that appears when a job is initiated or terminated, enter:

```
pm jobnames
```

## STOPTR Command

Use the STOPTR command to halt or reduce the information periodically displayed as a result of the TRACK command.

The complete syntax for the STOPTR command is:

|  |  |  |
|--|--|--|
| $\left\{ \begin{array}{l} \text{STOPTR} \\ \text{PT} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{TS} \\ \text{JOBS} \\ \text{J} \\ \text{A} \end{array} \right\}$ | $\left[ ,L= \left\{ \begin{array}{l} \text{a} \\ \text{cc} \\ \text{cca} \end{array} \right\} \right]$ |
|--|--|--|

### TS

The display of active time-sharing users is to be terminated.

### JOBS or J

The display of active jobs, MOUNT commands in execution, and active tasks is to be terminated.

### A

The display of active jobs, active time-sharing users, MOUNT commands in execution, and active tasks is to be terminated.

### L = a, cc, or cca

The display area (a), console identifier (cc), or both (cca) of the active MCS console where the display appears. 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 (but not time-sharing users), enter:

```
TR A,L
PT TS
```

## SWAP Command

## SWAP Command

Use the SWAP command to initiate an operator request for dynamic device reconfiguration (DDR) and to activate or deactivate system-initiated DDR.

The complete syntax for the SWAP command is:

|   |   |
|---|---|
| $\left\{ \begin{array}{l} \text{SWAP} \\ \text{G} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{OFF} \\ \text{ON} \\ \text{xxx,yyy} \end{array} \right\}$ |
|---|---|

### 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.

|   |         |
|---|---------|
| $\left\{ \begin{array}{l} \text{SWAP} \\ \text{G} \end{array} \right\}$ | xxx,yyy |
|---|---------|

**xxx**

The device number of the device from which the volume is to be swapped.

**yyy**

The device number of the device to which the volume is to be swapped.

xxx and yyy must be the same device type and must have compatible features.

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.
- zzz — an alternate “to” device is to be used.

After replying YES or zzz, wait for the following message before proceeding with the swap:

```
* id IGF502E PROCEED WITH SWAP OF xxx TO yyy
```

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.

DDR does not manage all the swapping of 3851 devices. Information on the switching from a failing 3851 to the backup 3851 is described in *Mass Storage System Extensions Operation*.

If you are using a 3348 Model 70F Data Module, make sure that the “to” 3340 device has the fixed-head feature installed.

*Example:* To move a volume from device 183 to device 283, enter:

```
swap 183,283
```

**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.

|   |   |
|---|---|
| $\left\{ \begin{array}{c} \text{SWAP} \\ \text{G} \end{array} \right\}$ | $\left\{ \begin{array}{c} \text{OFF} \\ \text{ON} \end{array} \right\}$ |
|---|---|

**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 must be satisfied before system-initiated DDR can be deactivated. If you cannot satisfy the requirements, you might have to re-IPL.

*Example:* To deactivate system-initiated DDR, enter:

```
g off
```

## SWITCH Command

## SWITCH Command

Use the SWITCH command to manually switch the recording of SMF data from one data set to another.

The complete syntax for the SWITCH command is:

|   |     |
|---|-----|
| $\left\{ \begin{array}{l} \text{SWITCH} \\ \text{I} \end{array} \right\}$ | SMF |
|---|-----|

### 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.



## TRACE Command

Use the TRACE command to:

- Start, stop, or modify system trace
- Start, stop, or modify master trace
- Display the current status of system trace and master trace

During system initialization, or whenever you reactivate system trace after a system trace failure, the system creates a TRACE address space. The TRACE 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 **16K**.

You can issue TRACE ST or TRACE MT from the master console only. You can issue TRACE STATUS from any console.

The complete syntax for the TRACE command is:

|       |   |
|-------|---|
| TRACE | $\left[ \begin{array}{l} \text{ST } \left[ \begin{array}{l} , \text{nnnK} \\ , \text{OFF} \end{array} \right] \left[ \begin{array}{l} , \text{BR} = \left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\} \end{array} \right] \\ \\ \text{MT } \left[ \begin{array}{l} , \text{nnnK} \\ , \text{OFF} \end{array} \right] \\ \\ \text{STATUS} \end{array} \right]$ |
|-------|---|

### ST

The system is to change the on or off status of system trace, the size of the system trace table, or the on or off status of the branch tracing function of system trace. Unless you specify **ST,OFF**, the system assumes you want to recreate the TRACE address space if it has terminated and turn system trace on if it is not on already.

### nnnK

The amount of preferred, real 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 16K for each processor, or the size established by the last TRACE command during the IPL that specified a table size.

### 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.

## TRACE Command

### **BR = ON or OFF**

The system is to turn on or turn off the branch tracing function 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.

### **MT**

The system is to change the on or off status of master trace or the size of the master trace table. Unless you specify **MT,OFF**, the system assumes you want to turn master trace on if it is not on already.

### **nnnK**

The master trace table size you want the system to use. You can specify for *nnn* any decimal number from 16 to 999. If a master trace table already exists, this new table replaces it. If you omit this parameter, the system assumes a table size of 24K, unless you specified a different table size in a previous TRACE command or, during system initialization, in a SCHEDxx member of SYS1.PARMLIB.

### **OFF**

The system is to stop master trace.

### **STATUS**

The system is to display the current status of system trace and master trace. Status information includes both the current sizes of the system and master trace tables and the on or off status of the system trace functions.

The system assumes the STATUS option when you enter the TRACE command without operands.

*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
```

## 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.

The complete syntax for the TRACK command is:

|   |  |   |  |
|---|--|---|--|
| $\left\{ \begin{array}{l} \text{TRACK} \\ \text{TR} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{TS} \\ \text{JOBS} \\ \text{J} \\ \text{A} \end{array} \right\}$ | $\left[ \begin{array}{l} , \text{LIST} \\ , \text{L} \end{array} \right]$ | $\left[ \begin{array}{l} , \text{L} = \left\{ \begin{array}{l} \text{a} \\ \text{cc} \\ \text{cca} \end{array} \right\} \end{array} \right]$ |
|---|--|---|--|

### TS, JOBS, J, 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

### 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” under the DISPLAY command earlier in this chapter for a description of the detailed information you get from LIST.

### L = a, cc, or cca

The display area (a), console identifier (cc), or both (cca) of the active MCS console where the requested display is to appear. 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).

## TRACK Command

*Example:* To periodically display 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 or MSS virtual unit, 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 MSS virtual unit, you must also enter a JES3 unload command (\*MODIFY,S,U=...). See *Operator's Library: JES3 Commands*.

The complete syntax for the UNLOAD command is:

|  |        |
|--|--------|
| $\left. \begin{array}{l} \text{UNLOAD} \\ \text{U} \end{array} \right\}$ | devnum |
|--|--------|

### devnum

The device number of the input/output device containing the volume to be unloaded. The volume to be unloaded cannot be a system residence or permanently resident volume. If the volume is in use, it is unloaded when the steps using it have terminated. If the volume is a tape volume, it is rewound and unloaded.

## VARY Command

## VARY Command

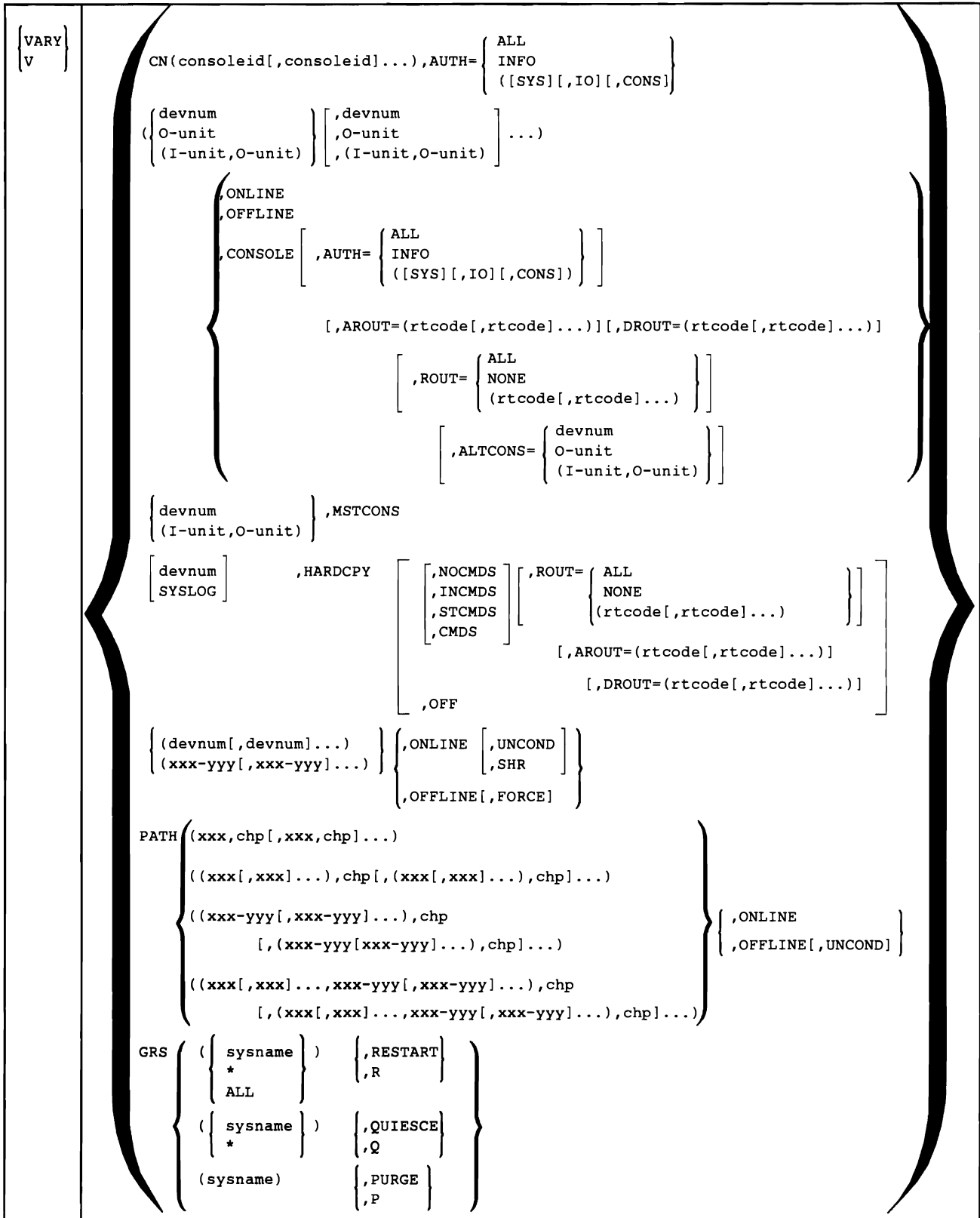
Use the VARY command to:

- Assign and control MCS consoles:
  - Define a device as an MCS console
  - Change the types of system commands a console can issue
  - Define or change the routing codes that send system messages to a console
  - Define an alternate MCS console
- Change the master console
- Control the hardcopy log:
  - Discontinue the hardcopy log function
  - Change the routing of messages or commands to the hardcopy log
- Change the online or offline status of resources:
  - Place a secondary console online or offline
  - Place an I/O device (or a range of I/O devices) online or offline
  - Place an I/O path (or paths) online or offline
  - Assign the 3480 tape device as sharable among more than one system.

*Note:* To change the online or offline status of processors, channel paths, and real storage sections, see the CONFIG command.

- Control a global resource serialization complex:
  - Temporarily remove a system from the global resource serialization ring
  - Restore a quiesced system to the global resource serialization ring
  - Remove a system from the global resource serialization complex
  - Rebuild a disrupted global resource serialization ring (that is, a group of inactive systems)

The complete syntax for the VARY command is:



# VARY Command

## Assigning and Controlling MCS Consoles

Use the following forms of the VARY command to:

- Define a device as an MCS console
- Change the types of system commands a console can issue
- Define or change the routing codes that send system messages to a console
- Define an alternate MCS console

The devices you specify in this command must have been defined as devices during system generation and consoles during initialization. See *MVS Configuration Program: Guide and Reference* for more information about specifying devices.

The ALTERNATE, AUTH, and ROUTCODE parameters on the CONSOLE statement in the CONSOLxx member of SYS1.PARMLIB define the console characteristics that operands on the VARY CN command can change.

For details about allocating JES3 consoles, and changing JES3 console authority and JES3 message routing, see *Operations: JES3 Commands*.

The syntax of the command is:

|               |   |
|---------------|---|
| { VARY<br>V } | {<br>CN(consoleid[,consoleid]...), AUTH= { ALL<br>INFO<br>([SYS][,IO][,CONS]) }<br><br>{ devnum<br>{ O-unit<br>{ (I-unit,O-unit) } } [,devnum<br>{ O-unit<br>{ (I-unit,O-unit) } } } ... )<br><br>,CONSOLE [ ,AUTH= { ALL<br>INFO<br>([SYS][,IO][,CONS]) } ] ]<br><br>[,AROUT=(rtcode[,rtcode]...)] [,DROUT=(rtcode[,rtcode]...)]<br><br>[ ,ROUT= { ALL<br>NONE<br>(rtcode[,rtcode]...) } ]<br><br>[ ,ALTCONS= { devnum<br>O-unit<br>(I-unit,O-unit) } ]<br>} |
|---------------|---|

*Note:* In general, you do not have to put a single device number, a single AUTH= operand, or a single routing code in parentheses. However, when you specify CN, you must put a single console in parentheses.

### CN(consoleid[,consoleid]...)

Change the indicated authority for the specified consoles. When you are uncertain of the current console identifiers, enter a DISPLAY CONSOLES command; see “Displaying Console Configuration Information,” under the DISPLAY command, earlier in this chapter, for further information on this command.



**devnum**

The device number of an input/output console device.

**O-unit**

The device number of a device with output capability.

**(I-unit,O-unit)**

The device numbers of the input and output devices that make up a composite console.

**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 the master console. Figure 3-4 on page 3-10 lists the commands and their associated groups.

**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 those other types specified. These parameters can be entered 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 Figure 3-8 on page 3-25 for a list of routing codes.

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 or ranges of routing codes on the same ROUT operand.

**rtcode-rtcode**

The console receives all routing codes in the specified range or ranges.

## VARY Command

### AROUT

The system is to add the specified routing codes to the routing codes already defined for the console.

#### **rtcode**

The system is to add the specified routing code(s) to the routing codes already defined for the console.

#### **rtcode - rtcode**

The system is to add the routing codes in the specified range(s) to the routing codes already defined for the console. You can specify single routing codes or ranges of routing codes on the same AROUT operand.

### DROUT

The system is to remove the specified routing code from the routing codes already defined for the console.

#### **rtcode**

The system is to remove the specified routing code(s) from the routing codes already defined for the console.

#### **rtcode - rtcode**

The system is to remove the routing codes in the specified range(s) from the routing codes already defined for the console. You can specify single routing codes or ranges of routing codes on the same DROUT operand.

### ALTCONS =

The device number of the alternate console. If this parameter is omitted, the alternate console is unchanged.

#### **devnum**

The console specified is to be assigned as an alternate to the console specified in the first operand.

#### **O-unit**

The output device specified is to be assigned as the alternate to the output-only console specified in the first operand.

#### **(I-unit, O-unit)**

The composite console specified is to be assigned as the alternate to the console specified in the first operand.

*Example 1:* The composite console, consisting of 00C and 00E, and console 009 are to become active consoles with 01F as the alternate console for both.

```
v ((i-00c,o-00e),009),console,altcons=01f
```

*Example 2:* 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 3:* To remove routing codes 8, 31, 32, 33, 34, and 35 from the existing routing codes for console 2B, enter:

```
v 2b,console,drout=(8,31-35)
```

## Changing the Master Console

Use the following form of the VARY command to switch the master console to another device. If you are not sure of the console assignments currently in effect, enter the DISPLAY CONSOLES command (see “Displaying Console Configuration Information” under the DISPLAY command, earlier in this chapter).

The syntax of the VARY ,MSTCONS command is:

|   |   |          |
|---|---|----------|
| $\left\{ \begin{array}{l} \text{VARY} \\ \text{V} \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{devnum} \\ \text{(I-unit,O-unit)} \end{array} \right\}$ | ,MSTCONS |
|---|---|----------|

### devnum

The device number of the device the system is to assign as the master console. This device must have input and output capabilities.

### (I-unit,O-unit)

The device numbers of the input (I-unit) and output (O-unit) devices forming a composite console that the system is to assign as the master console.

### MSTCONS

The system is to switch the master console.

*Example:* To assign the composite console consisting of devices 00C and 00E as the new master console, enter:

```
V (I-00C,O-00E),MSTCONS
```

## Controlling the Hardcopy Log

Use the following form of the VARY command to:

- Assign a unit as the hardcopy log device
- Discontinue the hardcopy log function
- Change the routing of messages or commands to the hardcopy log

The DEVNUM, ROUTCODE, and CMDLEVEL parameters on the HARDCOPY statement in the CONSOLxx member of SYS1.PARMLIB define characteristics that the operands on the VARY ,HARDCPY command can change.

# VARY Command

The syntax of the VARY ,HARDCPY command is:

|   |  |
|---|--|
| $\left\{ \begin{array}{l} \text{VARY} \\ \text{V} \end{array} \right\}$ | $\left[ \begin{array}{l} \text{devnum} \\ \text{SYSLOG} \end{array} \right]$ ,HARDCPY $\left[ \begin{array}{l} \left[ \begin{array}{l} \text{,NOCMDS} \\ \text{,INCMDS} \\ \text{,STCMDS} \\ \text{,CMDS} \end{array} \right] \left[ \begin{array}{l} \text{,ROUT=} \left\{ \begin{array}{l} \text{ALL} \\ \text{NONE} \\ \text{(rtcode[,rtcode]...)} \end{array} \right\} \end{array} \right] \\ \text{[,AROUT=(rtcode[,rtcode]...)]} \\ \text{[,DROUT=(rtcode[,rtcode]...)]} \\ \text{,OFF} \end{array} \right]$ |
|---|--|

## devnum

The device number of the device the system is to use for the hardcopy log. *Devnum* must be the device number of an active, nondisplay console. If you want to make a composite console the hardcopy log device, specify for *devnum* the device number of the output device.

## SYSLOG

The system log is to be the hardcopy log device.

When you omit the *devnum* and SYSLOG operands, the system changes what goes into the hardcopy log, rather than the hardcopy log device. When the system log is the hardcopy log device, you might want to see “Controlling System Information Recording” in Chapter 1 for information about scheduling system log output.

## HARDCPY

The system is to change the status of the hardcopy log.

## NOCMDS

The system is to record no operator commands or responses in the hardcopy log. Unless very few consoles are active and none of these consoles is a display console, you must have a hardcopy log. The system, therefore, usually ignores this operand and assumes CMDS.

## INCMDS

The system is to record operator commands and responses in the hardcopy log. The system does not record status displays in the hardcopy log.

## STCMDS

The system is to record operator commands, responses, and status displays (except dynamic status displays) in the hardcopy log.

## CMDS

The system is to record all operator commands, responses, and status displays (both static and dynamic status displays) in the hardcopy log.

*Note:* The system records JES2 commands you enter from a console in the hardcopy log if CMDS or INCMDS is in effect. The system does not record JES2 commands entered through the input stream in the hardcopy log.

## **OFF**

The system is to discontinue the hardcopy log. Unless very few consoles are active and none of these consoles is a display console, the system rejects this operand because you must have a hardcopy log. The system also rejects this operand when JES3 is active. If you do specify OFF, however, it must be the last parameter.

## **ROUT=**

The routing codes that the system is to send to the hardcopy log.

### **ALL**

The system is to send all system and operator communications to the hardcopy log.

### **NONE**

The system is to send no system or operator communications to the hardcopy log. This operand only works if your system has no active display consoles. If any display consoles are active, the system sends minimum routing codes to the hardcopy log, as described next under "rtcode."

### **rtcode**

The system is to send all messages with the specified routing codes to the hardcopy log. The minimum routing codes for the hardcopy log are 1, 2, 3, 4, 7, 8, 10, and 42. See Figure 3-8 on page 3-25 for a list of routing codes.

### **rtcode-rtcode**

The system is to send all messages with routing codes in the specified range (or ranges) to the hardcopy log. You can specify single routing codes or ranges of routing codes on the same ROUT operand.

## **AROUT**

The system is to add the specified routing codes to the routing codes already defined for the hardcopy log.

### **rtcode**

The system is to add the specified routing code(s) to the routing codes to the routing codes already defined for the hardcopy log.

### **rtcode - rtcode**

The system is to add the routing codes in the specified range(s) to the routing codes already defined for the hardcopy log. You can specify single routing codes or ranges of routing codes on the same AROUT operand.

## **DROUT**

The system is to remove the specified routing code from the routing codes already defined for the hardcopy log.

### **rtcode**

The system is to remove the specified routing code(s) from the routing codes to the routing codes already defined for the hardcopy log.

### **rtcode - rtcode**

The system is to remove the routing codes in the specified range(s) from the routing codes already defined for the hardcopy log. You can specify single routing codes or ranges of routing codes on the same DROUT operand.

## VARY Command

The system processes the ROUT, AROUT, and DROUT operands in the order that you specify them.

*Example 1:* To record all operator commands, responses, and status displays (except dynamic status displays) in the hardcopy log, enter:

```
v ,hardcpy,stcmds
```

*Example 2:* To have the hardcopy log 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 log, enter:

```
v ,hardcpy,arout=(11-13,17,44)
```

## Changing the Online or Offline Status of Resources

Use the VARY command to move some system resources online or offline, that is, to make them available or unavailable for system use. The following forms of the VARY command can:

- Place a secondary console online or offline
- Place an I/O device (or a range of I/O devices) online or offline
- Place an I/O path (or paths) online or offline

To move available processors, storage sections or elements, and channel paths online or offline, see the CONFIG command.

### Placing a Secondary Console Online or Offline

Use the following form of the VARY command to move a secondary console online or offline. You cannot move the master console or the hardcopy log online or offline without first assigning these functions to other consoles, as described earlier in this command.

|   |  |   |
|---|--|---|
| $\left\{ \begin{array}{l} \text{VARY} \\ \text{V} \end{array} \right\}$ | $\left\{ \begin{array}{l} (\text{devnum} ,\text{devnum} \dots) \\ (\text{O-unit}[\text{,O-unit}]\dots) \\ ((\text{I-unit},\text{O-unit})[\text{,(I-unit},\text{O-unit})]\dots) \end{array} \right\}$ | $\left\{ \begin{array}{l} \text{,OFFLINE} \\ \text{,ONLINE} \end{array} \right\}$ |
|---|--|---|

#### **devnum**

The device number(s) of the console(s) the system is to move online or offline. If you specify only one device number, you do not need to enter the parentheses.

#### **O-unit**

The device number of an output device. If you specify only one device number, you do not need to enter the parentheses.

**(I-unit,O-unit)**

The device numbers of the input and output devices that make up a composite console. If you specify only one composite console, you do not need the outside parentheses.

**ONLINE**

The system is to bring the specified device(s) online.

**OFFLINE**

The system is to take the specified device(s) offline.

*Example:* To take consoles 003 and 001 offline, enter:

```
v (003,001),offline
```

**Placing an I/O Device or a Range of I/O Devices Online or Offline**

Use the following form of the VARY command to move I/O devices online or offline. In a JES2 environment, use this command to assign the 3480 tape device as sharable 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 *MVS/XA Operations: JES3 Commands*.

If your installation requires more than two or three JES3-managed consoles, the device can be varied online after MVS initialization. To vary the device online, include an MVS VARY devnum,ONLINE command for each device that should be a JES3 console in the SYS1.COMMNDxx member of SYS1.PARMLIB.

|   |  |  |
|---|--|--|
| $\left\{ \begin{array}{l} \text{VARY} \\ \text{V} \end{array} \right\}$ | $\left\{ \begin{array}{l} (\text{devnum} [ , \text{devnum} ] \dots) \\ (\text{xxx-yyy} [ , \text{xxx-yyy} ] \dots) \end{array} \right\}$ | $\left\{ \begin{array}{l} , \text{ONLINE} \left[ \begin{array}{l} , \text{UNCOND} \\ , \text{SHR} \end{array} \right] \\ , \text{OFFLINE} [ , \text{FORCE} ] \end{array} \right\}$ |
|---|--|--|

**devnum**

The device number of a specific I/O device that the system is to move online or offline. You do not have to enter the parentheses when specifying only one device.

**xxx-yyy**

The lower device number (xxx) and the upper device number (yyy) of a range of devices the system is to move online or offline. You do not have to enter the parentheses when specifying only one range of devices.

**ONLINE**

The system is to bring the specified device(s) or range(s) of devices online, that is, make the device(s) or range(s) of devices available for allocation to problem programs and system tasks, if there is an online path to the device(s). 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.

# VARY Command

*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 the devices are not valid, you receive message IEE313I indicating the device numbers that are not valid.*

## UNCOND

The system is to bring the specified devices or range(s) of devices online, even if there are no paths to the devices. The system ignores this operand if you specify it for a tape or direct access device. Use the UNCOND operand carefully because it causes inaccessible devices to appear accessible to some system components.

## SHR

The system permits the 3480 you bring online to be shared among other processors. If you specify VARY 3480,ONLINE without the SHR keyword, the system assigns the 3480 to the system that issued the VARY command. If you specify SHR for a device other than the 3480, the system ignores the SHR keyword.

Do not use this keyword for devices managed by JES3. JES3-managed devices are automatically assigned sharable within the JES3 complex.

## 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 problem 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.

*Note:* When you issue the VARY OFFLINE command, the device immediately assumes a pending-offline status. The system does not allocate a device in pending-offline status to any job unless the job specifically requests the volume mounted on the device.

The VARY device OFFLINE command takes effect 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. This action causes a system task to start, allowing the system to take device(s) 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.DUMP data sets unless you first remove the SYS1.DUMP data sets from the system's list of SYS1.DUMP data sets with DUMPDS DEL,DSN = or DUMPDS DEL,UNIT = devnum commands.**

## FORCE

The system is to put the specified device(s) or range(s) of devices immediately in pending-offline status, even if they are currently active, allocated, or reserved. 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 bring 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.*

## VARY Command

- 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.

*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:

```
vary (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).

### Placing an I/O Path or Paths Online or Offline

Use the following form of the VARY command to move a path to a device or a range of devices online or offline. A path is the logical route between a processor and a device, consisting of the processor, a channel path, and a control unit. A path can be offline if:

- One or more of the path components is offline
- A VARY PATH OFFLINE command has been entered

*Notes:*

- You can bring paths taken offline with a VARY PATH command back online only with another VARY PATH command.
- When MVS takes the last path to a device offline, it also takes the device offline.

|  |   |
|--|---|
| $\left. \begin{array}{c} \text{VARY} \\ \text{V} \end{array} \right\}$ | $\text{PATH} \left\{ \begin{array}{l} (\text{xxx}, \text{chp} [, \text{xxx}, \text{chp}] \dots) \\ ((\text{xxx} [, \text{xxx}] \dots), \text{chp} [, (\text{xxx} [, \text{xxx}] \dots), \text{chp}] \dots) \\ ((\text{xxx-yyy} [, \text{xxx-yyy}] \dots), \text{chp} [, (\text{xxx-yyy} [, \text{xxx-yyy}] \dots), \text{chp}] \dots) \\ ((\text{xxx} [, \text{xxx}] \dots, \text{xxx-yyy} [, \text{xxx-yyy}] \dots), \text{chp} \\ \quad [, (\text{xxx} [, \text{xxx}] \dots, \text{xxx-yyy} [, \text{xxx-yyy}] \dots), \text{chp}] \dots) \end{array} \right\} \left\{ \begin{array}{l} , \text{ONLINE} \\ , \text{OFFLINE} [, \text{UNCOND}] \end{array} \right\}$ |
|--|---|

## PATH

The system is to move the specified path(s) online or offline. If the specified path is the last available path to an allocated device, the system does not take the device offline with the path.

### xxx

A single device associated with the path(s) the system is to move online or offline. You may specify for *xxx* (the device number) any number from 000 to FFF.

### (xxx[,xxx]...)

One or more single devices associated with the path(s) the system is to move online or offline. You may specify for *xxx* (the device number) any number from 000 to FFF.

### (xxx-yyy[,xxx-yyy]...)

One or more ranges of devices associated with the path(s) the system is to move online or offline. The lower device number of each range is *xxx* and the higher device number of each range is *yyy*. You may specify for *xxx* and *yyy* any numbers from 000 to FFF, provided that *xxx* is less than *yyy* in each range of devices.

### (xxx[,xxx]...,xxx-yyy[,xxx-yyy]...)

One or more single devices and one or more ranges of devices associated with the path(s) the system is to move online or offline. You may specify for *xxx* and *yyy* any numbers from 000 to FFF, provided that *xxx* is less than *yyy* in each range of devices.

### 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.

## ONLINE

The system is to bring the path(s) online.

## OFFLINE

The system is to take the specified path(s) offline. The system rejects this command if the specified path is the last available path to a device that is any one of the following:

- Online (either allocated or unallocated)
- In use by the system
- A console
- Assigned to JES3

## 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:

- Online and allocated
- In use by the system
- A console
- Assigned to JES3

## VARY Command

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.

*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.

### Controlling a Global Resource Serialization Complex

Use the VARY GRS command to:

- Temporarily remove a system from the global resource serialization ring
- Restore a quiesced system to the global resource serialization ring
- Remove a system from the global resource serialization complex
- Rebuild a disrupted global resource serialization ring (that is, a group of inactive systems)

Use the VARY GRS command mainly for recovery situations.

|   |   |
|---|---|
| $\left\{ \begin{array}{l} \text{VARY} \\ \text{V} \end{array} \right\}$ | GRS $\left( \left( \begin{array}{l} \text{sysname} \\ * \\ \text{ALL} \end{array} \right) \left( \begin{array}{l} ,\text{RESTART} \\ ,\text{R} \end{array} \right) \right.$<br>$\left. \left( \begin{array}{l} \text{sysname} \\ * \end{array} \right) \left( \begin{array}{l} ,\text{QUIESCE} \\ ,\text{Q} \end{array} \right) \right.$<br>$\left. \left( \text{sysname} \right) \left( \begin{array}{l} ,\text{PURGE} \\ ,\text{P} \end{array} \right) \right)$ |
|---|---|

#### 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, all previously-suspended requests for global resources are processed.

### **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. 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. Notify the system programmer if the system you are purging holds any resources.

## WRITELOG Command

## 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.

The complete syntax for the WRITELOG command is:

|                                 |  |
|---------------------------------|--|
| <code>{ WRITELOG<br/>W }</code> | <code>[ class<br/>CLOSE<br/>START ]</code> |
|---------------------------------|--|

*Note:* Specifying WRITELOG without any operands schedules the system log to be printed with the default output class specified at system generation. If no default output class was specified at system generation, specifying WRITELOG without any operands schedules the system log to be printed with output class A.

### **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 log and no other hardcopy device is available.

### **START**

The system log is to be restarted.

*Example:* To schedule the system log to the class D output queue, enter:

```
writelog d
```

## Glossary

This glossary includes definitions developed by the American National Standards Institute (ANSI). An asterisk (\*) identifies these definitions, which are reproduced from the *American National Dictionary for Information Processing*, copyright 1977 by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased from the American National Standards Institute, 1430 Broadway, New York, N.Y. 10018.

If the term you are seeking does not appear in this glossary, refer to *Vocabulary for Data Processing, Telecommunications, and Office Systems*, GC20-1699.

**abend.** Abnormal end of task.

**abnormal end of task (abend).** Termination of a task prior to its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

**ACR.** Alternate CPU recovery.

**action message retention facility.** A facility that, when active, retains all action messages except those specified by the installation in the MPFLSTxx member in effect.

**active.** A state of a system in a global resource serialization complex. The system is currently serializing global resources. It is a member of the global resource serialization ring.

**address space.** The complete range of addresses that is available to a programmer. See also *virtual address space*.

**address space identifier (ASID).** A unique system-assigned identifier for an address space.

**address stop.** A capability to specify at the system console an address which, when encountered, causes a halt in processing.

**allocate.** To assign a resource for use in performing a specific task.

**\*allocation.** See *storage allocation*.

**alternate.** A state of a CTC adapter in a global resource serialization complex. A system is not currently using this CTC adapter to send or receive messages, but it has the ability to do so.

**alternate CPU recovery (ACR).** A facility that attempts system recovery when a processing unit fails by transferring work to another processing unit.

**American National Standard Labels.** Magnetic tape labels that conform to the conventions established by the American National Standards Institute. Same as *ASCII label*.

**APAR.** Authorized program analysis report. A request for correction of a problem caused by a defect in a current unaltered release of a program. A program temporary fix or corrected code is issued to the customer and a corresponding correction is incorporated into subsequent releases of the program.

**ASID.** Address space identifier.

**automatic restart.** A restart that takes place during the current run, that is, without resubmitting the job. An automatic restart can occur within a job step or at the beginning of a job step. Contrast with *deferred restart*.

**automatic volume recognition (AVR).** A feature that allows the operator to mount labeled volumes on available I/O devices before the volumes are needed by a job step.

**auxiliary storage.** (1) Data storage other than main storage; for example, storage on magnetic tape or direct access devices. Same as *external storage, secondary storage*. (2) \* A storage that supplements another storage. Contrast with *main storage*.

**availability.** The degree to which a system or resource is ready when needed to process data.

**AVR.** Automatic volume recognition.

**background.** (1) In multiprogramming, the environment in which low-priority programs are executed. (2) Under TSO, the environment in which jobs submitted through the SUBMIT command or SYSIN are executed. One job step at a time is assigned to a region of main storage, and remains in main storage to completion. Contrast with *foreground*.

**background job.** (1) A low-priority job, usually a batched or noninteractive job. (2) Under TSO, a job entered through the SUBMIT command or through SYSIN. Contrast with *foreground job*.

**\*backspace character.** A format effector that causes the location of the printing or display position to be moved backward one printing or display space.

**batch processing.** (1) \* Pertaining to the technique of executing a set of computer programs such that each is completed before the next program of the set is started. (2) \* Pertaining to the sequential input of computer programs or data. (3) \* Loosely, the execution of computer programs serially. (4) Under TSO, the processing of one job step in a region, so called because jobs are submitted in a group or batch. (5) See also *execution batch processing*.

**batched job.** (1) A job that is grouped with other jobs as input to a computing system. (2) A job whose job control statements are grouped with job control statements of other jobs as input to a computing system.

**broadcast data set.** Under TSO, a system data set containing messages and notices from the system operator, administrators, and other users. Its contents are displayed to each terminal user when he logs on the system, unless suppressed by the user.

**buffer.** (1) \* A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transmitting data from one device to another. (2) An area of storage that is temporarily reserved for use in performing an input/output operation, into which data is read or from which data is written.

**cataloged data set.** A data set that is represented in an index, or hierarchy of indexes, that provide the means for locating it.

**cataloged procedure.** A set of job control statements that has been placed in a partitioned data set called the procedure library, and can be retrieved by naming it in an execute (EXEC) statement or started by the START command.

**cathode ray tube.** An electronic vacuum tube, such as a television picture tube, that can be used to display graphic images.

**central processing unit (CPU).** Same as *processor*.

**channel.** A hardware device that connects the processor and main storage with the I/O control units.

**channel path.** A connection between the channel subsystem and a control unit along which signals and data can be sent to perform I/O operations. Related to *channel*.

**channel-to-channel (CTC) adapter.** A hardware device that can be used to connect two channels on the same computing system or on different systems.

**checkpoint.** (1) \* A place in a routine where a check, or a recording of data for restart purposes, is performed. (2) A point at which information about the status of a job and the system can be recorded so that the job step can be later restarted. (3) To record such information.

**checkpoint data set.** A sequential or partitioned data set containing a collection of checkpoint entries. If a checkpoint data set is a partitioned data set, each checkpoint entry is a member.

**checkpoint restart.** The process of resuming a job at a checkpoint within the job step that caused abnormal termination. The restart may be automatic or deferred, where deferred restart involves resubmitting the job. See also *automatic restart*, *deferred restart*. Contrast with *step restart*.

**checkpoint/restart facility.** (1) A facility for restarting execution of a program at some point other than at the beginning, after the program was terminated due to a program or system failure. A restart can begin at a checkpoint or from the beginning of a job step, and uses checkpoint records to reinitialize the system. (2) Under TCAM, a facility that records the status of the teleprocessing network at designated intervals or following certain events. Following system failure, the system can be restarted and continue without loss of messages.

**\*clear.** To place one or more storage locations into a prescribed state, usually zero or the space character. Contrast with *set*.

**\*clock.** (1) A device that generates periodic signals used for synchronization. (2) A device that measures and indicates time. (3) A register or storage area whose contents change at regular intervals in such a way as to measure time.

**composite console.** A console consisting of two different devices that are considered as one unit. One device is used for input and the other for output, such as a reader and printer.



**configuration.** The group of machines, devices, and programs that make up a data processing system.

**\*console.** That part of a computer used for communication between the operator or maintenance engineer and the computer.

**\*control panel.** A part of a computer console that contains manual controls.

**control program.** A program that is designed to schedule and supervise the performance of data processing work by a computing system.

**control unit.** A device that controls input/output operations at one or more devices.

**conversational.** Pertaining to a program or a system that carries on a dialog with a terminal user, alternately accepting input and then responding to the input quickly enough for the user to maintain his train of thought.

**\*CPU.** Central processing unit. See *processor*.

**\*CRT display.** Cathode ray tube display.

**cursor.** A movable spot of light on a cathode ray tube unit that indicates where the next character will be entered.

**DASD.** Direct access storage device.

**data definition name.** The name of a data definition (DD) statement, which corresponds to a data control block that contains the same name. Abbreviated as *ddname*.

**data definition (DD) statement.** A job control statement that describes a data set associated with a particular job step.

**data set.** The major unit of data storage and retrieval in the operating system, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

**data set label.** (1) A collection of information that describes the attributes of a data set and is normally stored on the same volume as the data set. (2) A general term for data set control blocks and tape data set labels.

**data set name.** The term or phrase used to identify a data set.

**ddname.** Data definition name.

**DD statement.** Data definition statement.

**DDR.** Dynamic device reconfiguration.

**deallocate.** To release a resource that is assigned to a specific task.

**default value.** The choice among exclusive alternatives made by the system when no explicit choice is specified by the user.

**deferred restart.** A restart performed by the system on resubmission of a job by the programmer. The operator submits the restart deck to the system through a system input reader. Contrast with *automatic restart*.

**degradation machine check (DG).** Indicates a continuous degradation of system performance. Degradation machine checks can be reported when system-recovery conditions exceed a machine pre-defined threshold or when unit deletion occurs.

**demount.** To remove a volume from a tape unit or a direct access device.

**device number.** The unique number assigned to an external device.

**device type.** The general name for a kind of device; for example, 3330.

**\*diagnostic.** Pertaining to the detection and isolation of a malfunction or mistake.

**direct access.** (1) Retrieval or storage of data by a reference to its location on a volume, rather than relative to the previously retrieved or stored data. (2) \* Pertaining to the process of obtaining data from, or placing data into, storage where the time required for such access is independent of the location of the data most recently obtained or placed in storage. (3) \* Pertaining to a storage device in which the access time is effectively independent of the location of the data.

**direct access storage device (DASD).** A device in which the access time is effectively independent of the location of the data.

**direct address.** (1) An address that designates the storage location of an item of data to be treated as an operand. (2) Same as *one-level address* (3) Contrast with *indirect address*.

**disabled.** A state of a CTC adapter in a global resource serialization complex. A system cannot use this CTC adapter to send or receive messages.

**disk.** Loosely, a *disk storage device*.

**disk pack.** A removable direct access storage volume containing magnetic disks on which data is stored. Disk packs are mounted on a disk storage drive, such as the IBM 3330 Disk Storage Drive.

**disk storage.** Storage on direct access devices that record data magnetically on rotating disks.

**dismount.** See *demount*.

**\*display.** A visual presentation of data.

**display line.** On a display device, the series of character locations that constitute a horizontal line on the display surface. Same as *character row*.

**display position.** On a display device, the series of character locations that constitute a vertical line on the display surface.

**\*display tube.** A tube, usually a cathode ray tube, used to display data.

**domain.** A group of users whose characteristics are defined by the IPS and for which the SRM controls the multiprogramming level (the number of swappable users within the group that are in main storage). There may be 1 to 128 domains in the system.

**drum storage.** A direct access storage device that records data magnetically on a rotating cylinder. A type of addressable auxiliary storage associated with some computers.

**\*dump.** (1) To copy the contents of all or part of a storage, usually from an internal storage into an external storage. (2) A process as in (1). (3) The data resulting from the process as in (1).

**dynamic allocation.** Assignment of system resources to a program at the time the program is executed, rather than at the time it is loaded into main storage.

**dynamic data set definition.** The process of defining a data set and allocating auxiliary storage space for it during job step execution rather than before job step execution.

**dynamic device reconfiguration (DDR).** A facility that allows a demountable volume to be moved, and repositioned if necessary, without abnormally terminating the job or repeating the initial program load procedure.

**EBCDIC.** Extended binary coded decimal interchange code.

**end of block (EOB).** A code that marks the end of a block of data.

**end-of-file mark (EOF).** A code that signals that the last record of a file has been read.

**\*end of tape marker.** A marker on a magnetic tape used to indicate the end of the permissible recording area, for example, a photo-reflective strip, a transparent section of tape, or a particular bit pattern.

**environmental recording, editing, and printing.** See *EREP*.

**EOB.** End of block.

**EOF.** End-of-file mark.

**EOV.** End of volume.

**\*erase.** To obliterate information from a storage medium, for example, to clear, to overwrite.

**EREP.** The environmental recording, editing, and printing program that makes the data contained on the system recorder file available for further analysis.

**\*error message.** An indication that an error has been detected.

**execute (EXEC) statement.** A job control language (JCL) statement that marks the beginning of a job step and identifies the program to be executed or the cataloged or in-stream procedure to be used.

**execution batch processing.** The concatenation of jobs or transactions that come from possibly different input sources and have possibly different output routings, and the execution of these jobs or transactions as if they were one continuous input stream. The purpose is to improve system performance by reducing the overhead of bringing different batch processors in and out of the system.

**execution priority.** A rank assigned to a task that determines its precedence in being selected for execution.

**extended binary coded decimal interchange code (EBCDIC).** A set of 256 characters, each represented by eight bits.

**external interruption.** An interruption caused by a signal from the interruption key on the system console panel, from the timer, or from another computing system.

**external writer.** A program that supports the ability to write SYSOUT data in ways and to devices not supported by the job entry subsystem.

**fixed page.** A page in real storage that is not to be paged-out.

**foreground.** (1) In multiprogramming, the environment in which high-priority programs are executed. (2) Under TSO, the environment in which programs are swapped in and out of main storage to allow CPU time to be shared among terminal users. All command processor programs execute in the foreground. Contrast with *background*.

**foreground job.** (1) A high-priority job, usually a real-time job. (2) A teleprocessing or graphic display job that has an indefinite running time during which communication is established with one or more users at local or remote terminals. (3) Under TSO, any job executing in a swapped region of main storage, such as a command processor or a terminal user's program. Contrast with *background job*.

**generalized trace facility (GTF).** A service program that records significant system events (such as supervisor calls and start I/O operations) for the purpose of problem determination.

**general register.** A register used for operations such as binary addition, subtraction, multiplication, and division. General registers are used primarily to compute and modify addresses in a program.

**global resource serialization.** A facility used to serialize access to system resources.

**global resource serialization complex.** The set of all systems that can potentially share global resources. That is, systems in the quiesced, inactive, joining, restarting, and active states.

**global resource serialization ring.** All systems in the global resource serialization complex that are currently sharing global resources. That is, all active systems in the complex.

**Greenwich mean time (GMT).** The mean solar time of the meridian of Greenwich used as the prime basis of standard time throughout the world. See also *TOD clock*.

**GTF.** Generalized trace facility.

**hardcopy.** A printed copy of machine output in a visually readable form; for example, printed reports, listings, documents, and summaries. See also *display*.

**hardcopy log.** In systems with multiple console support or a graphic console, a permanent record of system activity.

**\*hardware.** Physical equipment, as opposed to the computer program or method of use; for example, mechanical, magnetic, electrical, or electronic devices. Contrast with *software*.

**Hardware system area (HSA).** A part of real storage reserved for hardware use, not accessible for programming use. HSA storage appears as offline when you issue the DISPLAY M (STOR) command. The area contains microcode, hardware control blocks, and hardware communication areas.

**high local date.** The highest valid local date that can be specified. (On most systems, 99.365, that is, December 31 1999.)

**HSA.** Hardware system area.

**inactive.** A state of a system in a global resource serialization complex. The system is not currently serializing global resources, but it can be used to restart the global resource serialization ring. The system suspends any of its tasks that try to obtain or release global resources.

**indirect address.** (1) An address that designates the storage location of an item of data to be treated as the address of an operand, but not necessarily as its direct address. (2) Same as *multi-level address*. (3) Contrast with *direct address*.

**initial program load (IPL).** The initialization procedure that causes an operating system to commence operation.

**\*initial program loader (IPL).** The procedure that causes the initial part of an operating system or other program to be loaded such that the program can then proceed under its own control.

**\*initialize.** To set counters, switches, and addresses to zero or other starting values at the beginning of, or at prescribed points in, a computer routine. Same as *prestore*.

**\*input.** (1) Pertaining to a device, process, or channel involved in the insertion of data or states, or to the data or states involved. (2) One, or a sequence of, input states. (3) Same as *input device*.

**\*input device.** The device or collective set of devices used for conveying data into another device. Same as *input (3)*.

**input job queue.** Same as *input work queue*.

**input/output.** (1) \* Pertaining to either input or output, or both. (2) A general term for the equipment used to communicate with a computer, commonly called I/O. (3) The data involved in such communication. (4) The media carrying the data for input/output.

**input/output interruption.** See *I/O interruption*.

**input queue.** Same as *input work queue*.

**input reader.** See *reader (2)*.

**input stream.** The sequence of job control statements and data submitted to an operating system on an input unit especially activated for this purpose by the operator. Same as *input job stream, job input stream*.

**input stream control.** Same as *JES reader*.

**input work queue.** A queue (waiting list) of job definitions in direct access storage assigned to a job class and arranged in order of assigned priority. Same as *input queue, input job queue, job queue*.

**\*instruction counter.** A counter that indicates the location of the next computer instruction to be interpreted.

**internal reader.** A facility that allows JES2 or JES3 to accept commands, control cards, and jobs from a program as though they were submitted through a card reader supported by JES.

**interrupt.** (1) \* To stop a process in such a way that it can be resumed. (2) In data transmission, to take an action at a receiving station that causes the transmitting station to terminate a transmission.

**interruption.** A break in the normal sequence of instruction execution. It causes an automatic transfer to a preset storage location where appropriate action is taken. See also *external interruption, I/O interruption, machine check interruption, program check interruption, SVC interruption*.

**in-use.** A state of a CTC adapter in a global resource serialization complex. The system is using this CTC adapter to send messages to and receive messages from the system at the other end of the CTC adapter.

**\*I/O.** Input/output.

**I/O engine.** An integrated unit of the processor complex that performs the data transfer functions.

**I/O interruption.** An interruption caused by the termination of an I/O operation or by operator intervention at the I/O device.

**IPL.** (1) \* Initial program loader. (2) Initial program load.

**JCL.** Job control language.

**JES.** Job entry subsystem.

**JES reader.** The part of the job entry subsystem that controls the input stream and its associated job control statements. Same as *input stream control*.

**JES writer.** The part of the job entry subsystem that controls the output of specified data sets. Same as *output stream control*.

**job.** (1) \* A specified group of tasks prescribed as a unit of work for a computer. By extension, a job usually includes all necessary computer programs, linkages, files, and instructions to the operating system. (2) A collection of related problem programs, identified in the input stream by a JOB statement followed by one or more EXEC and DD statements. (3) See also *batched job, background job, foreground job*.

**job class.** Any one of a number of job categories that can be defined under an MVS/XA control program configuration. By classifying jobs and directing initiators to initiate specific classes of jobs, it is possible to control the mixture of jobs that are performed concurrently.

**job control language (JCL).** A programming language used to code job control statements.

**job entry subsystem (JES).** A system facility for spooling, job queuing, and managing the scheduler work area.

**jobname.** The name assigned to a JOB statement; it identifies the job to the system.

**job priority.** A value assigned to a job that is used as a measure of the job's relative importance while the job contends with other jobs for system resources.

**job queue.** Same as *input work queue*.

**job (JOB) statement.** The job control statement that identifies the beginning of a job. It contains such information as the name of the job, an account number, and the class and priority assigned to the job.

**job step.** (1) \* The execution of a computer program explicitly identified by a job control statement. A job may specify that several job steps be executed. (2) A unit of work associated with one processing program or one cataloged procedure and related data. A job consists of one or more job steps.

**job step task.** A task that is initiated by an initiator/terminator in the job scheduler in accordance with specifications in an execute (EXEC) statement.

**job stream.** See *input stream, output stream*.

**joining.** A state of a system in a global resource serialization complex. The system is in the process of joining the global resource serialization ring.

**K.** 1024 bytes; used in referring to storage capacity.

**keyboard.** A device for the encoding of data by key depression, which causes the generation of the selected code element.

**keyword.** A part of a command operand that consists of a specific character string (such as *DSNAME =*).

**keyword parameter.** A parameter that consists of a keyword, followed by one or more values. Contrast with *positional parameter*.

**label.** (1) \* One or more characters used to identify a statement or an item of data in a computer program. (2) An identification record for a tape or disk file.

**library.** (1) \* A collection of organized information used for study and reference. (2) \* A collection of related files. For example, one line of an invoice may form an item, a complete invoice may form a record, the complete set of such records may form a file, the collection of inventory control files may form a library, and the libraries used by an organization are known as its data bank. (3) Any partitioned data set.

**light pen attention.** An interruption generated by a light pen when it senses light on the screen of a CRT display device. Same as *selector pen attention*.

**line number.** (1) A number associated with a line in a printout or display. (2) In systems with time-sharing, a number associated with a line in a line data set.

**listing.** A printout, usually prepared by a language translator, that lists the source language statements and contents of a program.

**\*load.** In programming, to enter data into storage or working registers.

**logoff.** The procedure by which a user ends a terminal session.

**logon.** The procedure by which a user begins a terminal session.

**\*loop.** A sequence of instructions that is executed repeatedly until a terminal condition prevails.

**M.** 1,048,576 bytes; used in referring to storage capacity.

**machine check interruption.** An interruption that occurs as a result of an equipment malfunction or error. A machine check interrupt can be either hardware-recoverable, software-recoverable, or non-recoverable.

**\*main frame.** Same as *processor*.

**main storage.** (1) \* The general purpose storage of a computer. Usually, main storage can be accessed

directly by the operating registers. Contrast with *auxiliary storage*. (2) All program-addressable storage from which instructions may be executed and from which data can be loaded directly into registers. (3) See also *real storage, virtual storage*.

**maintenance and repair configuration.** The CE places a processor complex in this state so that he can maintain or repair an element.

**master console.** In a system with multiple consoles, the basic console used for communication between the operator and the system.

**master scheduler.** A control program routine that responds to operator commands and initiates the requested actions.

**master trace.** A centralized data tracing facility provided by the master scheduler for use in servicing message processing portions of MVS/XA.

**MCS.** Multiple-console support.

**message.** (1) \* An arbitrary amount of information whose beginning and end are defined or implied. (2) In telecommunications, a combination of characters and symbols transmitted from one point to another on a network. (3) \* See *error message, operator message*.

**message processing facility (MPF).** A facility used to control message retention, suppression, and presentation.

**message queue.** A queue of messages that are awaiting processing or waiting to be sent to a terminal.

**message text.** The part of a message consisting of the actual information that is routed to a user at a terminal or to a program.

**monitoring and system support facility (MSSF).** A logical unit of a processor complex that includes circuits to control monitoring, configuration, and other support-related functions. The MSSF supports normal operations and provides maintenance for the central processor. The MSSF consists of hardware (MSSE - monitoring and system support element) and microcode (the MSSM - monitoring and system support manager). A major part of the MSSF is located in the processor controller, and the remainder is located in the processing unit.

**\*monitor.** Software or hardware that observes, supervises, controls or verifies the operations of a system.

**MPF.** Message processing facility.

**MSC.** Mass storage control.

**MSS.** Mass storage system.

**MSSF.** Monitoring and system support facility.

**multiple-console support (MCS).** A feature of MVS that permits selective message routing to up to 99 operator's consoles.

**multiprocessing.** (1) \* Pertaining to the simultaneous execution of two or more computer programs or sequences of instructions by a computer network. (2) \* Loosely, parallel processing. (3) Simultaneous execution of two or more sequences of instructions by a multiprocessor.

**multiprocessing system.** A computing system employing two or more interconnected processors to execute programs simultaneously.

**multiprocessor.** (1) \* A computer employing two or more processors under integrated control. (2) A system consisting of two or more processors that can communicate without manual intervention.

**multiprogramming level.** The number of swappable users in a domain or in the system that are in main storage. The sum of the domain multiprogramming levels is the system multiprogramming level.

**NIP.** Nucleus initialization program.

**no-consoles condition.** A condition in which the system is unable to access any full-capability console device.

**nonstandard labels.** Labels that do not conform to American National Standard or IBM System/370 standard label conventions.

**nontemporary data set.** A data set that exists after the job that created it terminates. Contrast with *temporary data set*.

**nucleus.** That portion of an address space that contains the control program.

**nucleus initialization program (NIP).** The program that initializes the control program; it allows the operator to request last minute changes to certain options specified during system generation.

**\*offline.** Pertaining to equipment or devices not under control of the processor.

**\*online.** Pertaining to equipment or devices under control of the processor.

**operand.** (1) \* That which is operated upon. An operand is usually identified by an address part of an

instruction. (2) Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor.

**operating system.** Software that controls the execution of computer programs and that may provide scheduling, debugging, input/output control, accounting, compilation, storage assignment, data management, and related services.

**operator command.** A statement to the control program, issued via a console device, or control terminal, that causes the control program to provide requested information, alter normal operations, initiate new operations, or terminate existing operations.

**operator control station.** Under TCAM, any station that is eligible to enter the operator commands.

**operator message.** A message from the operating system or a problem program directing the operator to perform a specific function, such as mounting a tape reel, or informing him of specific conditions within the system, such as an error condition.

**output.** (1) Pertaining to a device, process, or channel path involved in an output process, or to the data or states involved. (2) One, or a sequence of, output states. (3) Same as *output device*.

**output class.** Under MVS/XA, any one of up to 36 different categories, defined at an installation, to which output data produced during a job step can be assigned.

**output device.** The device or collective set of devices used for conveying data out of another device. Same as *output (3)*.

**output queue.** See *output work queue*.

**output stream.** Diagnostic messages and other output data issued by an operating system or a processing program on output devices especially activated for this purpose by the operator. Same as *job output stream*.

**output stream control.** Same as *JES writer*.

**output work queue.** A queue of control information describing system output data sets that specifies to an output writer the location and disposition of system output.

**output writer.** A part of the job scheduler that transcribes specified output data sets onto a system output device independently of the program that produced the data sets.

**page.** (1) In virtual storage systems, a fixed-length block of instructions, data, or both, that can be transferred between real storage and external page storage. (2) To transfer instructions, data, or both, between real storage and external page storage.

**page data set.** A data set in external page storage in which pages are stored.

**page fault.** A program interruption that occurs when a page that is marked "not in real storage" is referred to by an active page.

**paging.** The process of transferring pages between real storage and external page storage.

**paging device.** A direct access storage device on which pages (and possibly other data) are stored.

**parameter.** (1) \* A variable that is given a constant value for a specific purpose or process. (2) See *keyword parameter*, *positional parameter*.

**password.** (1) A unique string of characters that a program, computer operator, or user must supply to meet security requirements before gaining access to data. (2) In systems with time-sharing, a one-to-eight-character symbol that the user may be required to supply at the time the user logs on the system. The password is confidential, as opposed to the user identification.

**physical unit.** (1) The control unit or cluster controller of an SNA terminal. (2) The part of the control unit or cluster controller that fulfills the role of a physical unit as defined by systems network architecture.

**PFK.** Program function key.

**PFK capability.** On a display console, indicates that program function keys are supported and were specified at system generation.

**positional parameter.** A parameter that must appear in a specified location, relative to other parameters. Contrast with *keyword parameter*.

**\*postmortem dump.** A static dump, used for debugging purposes, performed at the end of a machine run.

**printer.** (1) A device that writes output data from a system on paper or other media.

**priority.** A rank assigned to a task that determines its precedence in receiving system resources. See also *job priority*.

**private library.** A user-owned library that is separate and distinct from the system library.

**private volume.** A mounted volume that the system can allocate only to an output data set for which a specific volume request is made. A private volume is demounted after its last use in a job step. Contrast with *public volume*.

**problem program.** Any program that is executed when the processor is in the problem state; that is, any program that does not contain privileged instructions. This includes IBM-distributed programs, such as language translators and service programs, as well as programs written by a user.

**procedure.** (1) \* The course of action taken for the solution of a problem. (2) In a procedure-oriented language, an independent, named block of statements that defines a specific portion of a program.

**procedure library.** A program library in direct access storage containing job definitions. The reader/interpreter can be directed to read and interpret a particular job definition by an execute statement in the input stream.

**processor.** A functional unit that interprets and executes instructions.

**processor controller.** A part of the processor complex that monitors and supervises the ongoing operations in the system. The processor controller is a combination of hardware frames containing interface adapter elements, channel-to-channel adapters, and major portions of the MSSF.

**processor storage.** (1) General purpose storage that is part of a processor. The term pertains to physical locations in hardware devices. (2) Same as *real storage*.

**PROC statement.** A job control statement used in cataloged or in-stream procedures. It can be used to assign default values for symbolic parameters contained in a procedure. For in-stream procedures, it is used to mark the beginning of the procedure.

**\*program.** (1) A series of actions proposed in order to achieve a certain result. (2) Loosely, a routine. (3) To design, write, and test a program as in (1). (4) Loosely, to write a routine.

**program check interruption.** An interruption caused by unusual conditions encountered in a program, such as incorrect operands.

**program function key.** A key on the keyboard of a display device that passes a signal to a program to call for a particular program operation.

**program interruption.** See *interruption*, *program check interruption*.

**\*programmed check.** A check procedure designed by the programmer and implemented specifically as a part of his program.

**program status word (PSW).** A doubleword in main storage used to control the order in which instructions are executed, and to hold and indicate the status of the computing system in relation to a particular program.

**PSA.** Prefixed save area.

**pseudo-master console:.** A subsystem console that has system command authority similar to that of an MCS master console.

**PSW.** Program status word.

**public volume.** A mounted volume that the system can allocate to an output data set for which a nonspecific volume request is made. A public volume remains mounted until the device on which it is mounted is required for another volume. Contrast with *private volume*.

**queue.** (1) A waiting line or list formed by items in a system waiting for service; for example, tasks to be performed or messages to be transmitted in a message switching system. (2) To arrange in, or form, a queue.

**quiesced.** A state of a system in a global resource serialization complex. The system is not currently serializing global resources, but it can be used to restart this system but not the global resource serialization ring. The system suspends any of its tasks that try to obtain or release global resources.

**\*quiescing.** (1) The process of bringing a device or a system to a halt by rejection of new requests for work. (2) The process of bringing a system to a halt by rejection of new jobs.

**quiet.** A state of a CTC adapter in a global resource serialization complex. A system at the opposite end of the CTC adapter does not respond.

**\*read.** To acquire or interpret data from a storage device, a data medium, or any other source.

**reader.** (1) A device that converts information in one form of storage to information in another form of storage. (2) A part of the scheduler that reads an input stream into the system.

**real address.** The address of a location in real storage.

**real storage.** The main storage in a virtual storage system. Physically, real storage and main storage are identical. Conceptually, however, real storage represents only part of the range of addresses available to the user of a virtual storage system.

**recoverable ABEND.** An error in which control is passed to a specified routine that allows continued execution of the program. Contrast with *unrecoverable ABEND*.

**recoverable error.** An error condition that allows continued execution of a program.

**reel.** A mounting for a roll of tape.

**\*register.** A device capable of storing a specified amount of data, such as one word.

**remote job entry.** Submission of job control statements and data from a remote terminal, causing the jobs described to be scheduled and executed as though encountered in the input stream.

**\*rerun.** A repeat of a machine run, usually because of a correction, an interruption, or a false start.

**\*reset.** (1) To restore a storage device to a prescribed initial state, not necessarily that denoting zero. (2) To place a binary cell into the state denoting zero.

**resident.** Pertaining to a program that is permanently located in storage. For example, a system library on direct access storage.

**resource.** Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processor, data sets, and control or processing programs.

**\*restart.** To reestablish the execution of a routine, using the data recorded at a checkpoint.

**restarting.** A state of a system in a global resource serialization complex. The system was an inactive or quiesced system that is now in the process of rejoining the global resource serialization ring.

**rewind.** To return a magnetic or paper tape to its beginning.

**\*routine.** An ordered set of instructions that may have some general or frequent use.

**routing.** The assignment of the communications path by which a message or telephone call will reach its destination.



**routing code.** A code assigned to an operator message and used to route the message to the proper console.

**secondary console.** In a system with multiple consoles, any console except the master console. The secondary console handles one or more assigned functions on the multiple console system.

**secondary storage.** Same as *auxiliary storage*.

**security.** Prevention of access to or use of data or programs without authorization.

**\*selective dump.** A dump of one or more specified storage locations.

**selector pen attention.** Same as *light pen attention*.

**sequence.** (1) \* An arrangement of items according to a specified set of rules. (2) In sorting, a group of records whose control fields are in ascending or descending order, according to the collating sequence.

**\*sequential.** Pertaining to the occurrence of events in time sequence, with little or no simultaneity or overlap of events.

**service configuration.** The CE places a processor complex in a service configuration so that he can check the state of the processor complex or diagnose problems.

**session.** The period of time during which a user engages in a dialog with a conversational time-sharing system; the elapsed time from when a terminal user logs on the system until he logs off the system; in VTAM, the period of time during which a node is connected to an application program.

**\*set.** (1) A collection. (2) To place a storage device into a specified state, usually other than that denoting zero or space character. Contrast with *clear*.

**setup.** The preparation of a computing system to perform a job or job step. Setup is usually performed by an operator and often involves performing routine functions, such as mounting tape reels and loading card decks.

**shared DASD option.** An option that enables independently operating computing systems to jointly use common data residing on shared direct access storage devices.

**shared file.** A direct access device that may be used by two systems at the same time; a shared file may link two systems.

**\*software.** A set of programs, procedures, and, possibly, associated documentation concerned with the operation of a data processing system. For example,

compilers, library routines, manuals, circuit diagrams. Contrast with *hardware*.

**space.** (1) \* A site intended for the storage of data, for example, a site on a printed page or location in a storage medium. (2) \* A basic unit of area, usually the size of a single character. (3) \* One or more space characters. (4) \* To advance the reading or display position according to a prescribed format, for example, to advance the printing or display position horizontally to the right or vertically down. Contrast with *backspace*.

**spooled data set.** A data set written on an auxiliary storage device.

**spooling.** The reading and writing of input and output streams on auxiliary storage devices, concurrently with job execution, in a format convenient for later processing or output operations.

**standard job.** A JES3 job that consists of input service, main service, output service, and purge performed in that order.

**\*step.** (1) One operation in a computer routine. (2) To cause a computer to execute one operation. (3) See *job step*.

**step restart.** A restart that begins at the beginning of a job step. The restart may be automatic or deferred, where deferral involves resubmitting the job. Contrast with *checkpoint restart*.

**\*storage.** (1) Pertaining to a device into which data can be entered, in which they can be held, and from which they can be retrieved at a later time. (2) Loosely, any device that can store data.

**\*storage allocation.** (1) The assignment of blocks of data to specified blocks of storage.

**\*storage device.** A device into which data can be inserted, in which they can be retained, and from which they can be retrieved.

**storage reconfiguration.** A function that makes an area of defective storage unavailable and frees any system resources associated with it.

**\*store.** (1) To enter data into a storage device. (2) To retain data in a storage device. (3) A storage device.

**subsystem.** A secondary or subordinate system, usually capable of operating independently of, or asynchronously with, a controlling system.

**subsystem-id.** The three-digit address of a specific MSS device. These devices are part of the MSS subsystem and do not have a unit control block in the host system.

**supervisor.** The part of a control program that coordinates the use of resources and maintains the flow of processor operations.

**SVC interruption.** An interruption caused by the execution of a supervisor call instruction, causing control to be passed to the supervisor.

**swap.** (1) To write the main storage image of a job to auxiliary storage and read the image of another job into main storage. (2) To write the active pages of a job to external page storage and read pages of another job from external page storage into real storage.

**swap data set.** A data set dedicated to the swapping operation.

**swapping.** (1) A process that writes a job's main storage image to auxiliary storage, and reads another job's main storage image into main storage. (2) A paging technique that writes the active pages of a job to external page storage and reads pages of another job from external page storage into real storage.

**\*syntax.** (1) The structure of expressions in a language. (2) The rules governing the structure of a language.

**SYSGEN.** System generation.

**SYSIN.** A system input stream; also, the name used as the data definition name of a data set in the input stream.

**SYSLOG.** System log.

**SYSOUT.** A system output stream; also, an indicator used in data definition statements to signify that a data set is to be written on a system output unit.

**\*system.** (1) An assembly of methods, procedures, or techniques united by regulated interaction to form an organized whole. (2) An organized collection of people, machines, and methods required to accomplish a set of specific functions.

**system generation (SYSGEN).** The process of using an operating system to assemble and link together all of the parts that constitute another operating system.

**system input device.** A device specified as the source of an input stream.

**system library.** A collection of data sets in which the various parts of an operating system are stored.

**system log.** A data set in which job-related information, operational data, descriptions of unusual occurrences, commands, and messages to or from the operator may be stored. Abbreviated **SYSLOG**.

**system management facilities (SMF).** A control program feature that provides the means for gathering and recording information that can be used to evaluate system use.

**system output device.** A device assigned to record output data for a series of jobs.

**system output writer.** A job scheduler function that transcribes specified output data sets onto a system output unit, independently of the program that produced the data sets.

**system programmer.** (1) A programmer who plans, generates, maintains, extends, and controls the use of an operating system with the aim of improving the overall productivity of an installation. (2) A programmer who designs programming systems and other applications.

**system queue area (SQA).** An area of virtual storage reserved for system-related control blocks.

**system recovery machine checks (SR).** Indicates that malfunctions were detected but have been successfully corrected or circumvented without loss of system integrity.

**system residence volume.** The volume on which the nucleus of the operating system and the highest-level index of the catalog are located.

**system resource.** Any facility of the computing system that may be allocated to a task.

**system restart.** (1) A restart that allows reuse of previously-initialized input and output work queues. Same as *warm start*. (2) A restart that allows reuse of a previously-initialized link pack area. Same as *quick start*.

**task.** A unit of work for the processor; therefore, the basic multiprogramming unit under the control program.

**TCAM.** Telecommunications access method.

**telecommunications access method (TCAM).** A method used to transfer data between main storage and remote or local terminals. Application programs use either **GET** and **PUT** or **READ** and **WRITE** macro instructions to request the transfer of data, which is performed by a message control program. The message control program synchronizes the transfer, thus eliminating delays for terminal input/output operations.

**teleprocessing (TP).** The processing of data that is received from or sent to remote locations by way of telecommunication lines.

**temporary data set.** A data set that is created and deleted in the same job. Contrast with *nontemporary data set*.

**terminal.** A device, usually equipped with a keyboard and some kind of display, capable of sending and receiving information over a link.

**terminal user.** In systems with time-sharing, anyone who is eligible to log on.

**text.** The data portion of a telecommunications message.

**time-sharing.** (1) \* Pertaining to the interleaved use of the time of a device. (2) A method of using a computing system that allows a number of users to execute programs concurrently and to interact with the programs during execution.

**TOD clock.** A binary counter that measures the time elapsed at Greenwich, England, in microseconds, since 0 hours, 0 minutes, 0 seconds, January 1, 1900. See also *Greenwich mean time*.

**TP.** Teleprocessing.

**trace.** (1) The record of a series of events. (2) To record a series of events as they occur.

**tracing routine.** A routine that provides a historical record of specified events in the execution of a program.

**unallocate.** See *deallocate*.

**universal character set (UCS).** A printer feature that permits the use of a variety of character arrays.

**unrecoverable ABEND.** An error condition that results in abnormal termination of a program. Contrast with *recoverable ABEND*.

**unrecoverable error.** An error that results in abnormal termination of a program.

**update.** To modify a master file with current information according to a specified procedure.

**user.** Anyone who requires the services of a computing system. See also *terminal user*.

**user exit.** A point in an IBM-supplied program at which a user exit routine may be given control.

**user exit routine.** A routine written by a user to take control at a user exit of a program supplied by IBM.

**USERID.** User identification.

**user identification (USERID).** A one-to-eight-character symbol identifying a system user.

**utility program.** A problem program designed to perform an everyday task, such as transcribing data from one storage device to another.

**Vector Facility.** A processor with an attached Vector Facility can run programs that issue vector instructions.

**virtual address.** An address that refers to virtual storage and must, therefore, be translated into a real storage address when it is used.

**virtual address space.** The virtual storage assigned to a job, terminal user, or system task.

**virtual storage.** Addressable space that appears to the user as real storage, from which instructions and data are mapped into real storage locations. The size of virtual storage is limited by the addressing scheme of the computing system (or virtual machine) and by the amount of auxiliary storage available, rather than by the actual number of real storage locations.

**virtual telecommunications access method (VTAM).** A set of IBM programs that control communication between terminals and application programs.

**volume.** (1) That portion of a single unit of storage which is accessible to a single read/write mechanism, for example, a drum, a disk pack, or part of a disk storage module. (2) A recording medium that is mounted and demounted as a unit, for example, a reel of magnetic tape, a disk pack, a data cell.

**volume serial number.** A number in a volume label that is assigned when a volume is prepared for use in the system.

**volume table of contents (VTOC).** A table on a direct access volume that describes each data set on the volume.

**VTAM.** Virtual telecommunications access method.

**VTOC.** Volume table of contents.

**wait state.** (1) The condition of a task that is dependent on one or more events in order to enter the ready condition. (2) The condition of a processor when all operations are suspended.

**warm start.** Same as *system restart (1)*.

**warning message.** An indication that a possible error has been detected. Contrast with *error message*.

**work file.** (1) In sorting, an intermediate file used for temporary storage of data between phases. (2) See also *work volume*.

**work volume.** A volume made available to the system to provide storage space for temporary fields or data sets at peak loads.

**\*write.** To record data in a storage device or a data medium. The recording need not be permanent, such as the writing on a cathode ray tube display device.

**writer.** See *output writer*.



# Index

## A

- A (RELEASE command) 4-1
  - abend (abnormal end of task) completion code 4-162
  - ACR (alternate CPU recovery) 1-27, 4-109
  - action message retention facility
    - activating 4-42
    - controlled by MPFLSTxx 3-5
    - deactivating 3-21, 4-42
    - deleting action messages 4-38, 4-39
    - description of 3-21
    - DISPLAY MPF command 4-86
    - displaying action messages not retained 3-19, 3-24, 4-86
    - displaying action messages that are retained 3-67
    - displaying outstanding action messages 4-78
    - displaying the status of 4-42
    - retrieving action messages 3-21
    - when JES3 uses MCS consoles 3-22
  - action messages
    - backup of 3-18
    - deleting 3-38
    - deleting retained 3-36, 4-38
    - displaying information about 3-22
    - limiting number received at a console 3-67
    - removing from screen 3-41, 3-66
    - requeuing 2-50
    - retained by action message retention facility 3-19, 3-21
  - activating
    - action message retention facility 3-21, 4-42
    - general WTO user exit IEAVMXIT 3-23, 4-44
  - active consoles
    - display status of 4-69
  - ADD dump mode 4-13
  - adding
    - auxiliary storage 4-125
    - dump data sets 4-97
  - address
    - direct 4-177
    - indirect 4-177
    - wraparound 4-172
  - address compare in SLIP, definition 4-164
  - address space 4-156, 4-160, 4-161
    - qualifying direct or indirect address to 4-181
  - ADSETxx member of SYS1.PARMLIB 4-142
  - alarm (see also audible alarm)
    - no-consoles condition 2-54
  - altering
    - current console specifications 4-48
    - display area specifications 4-37
  - alternate console
    - arrangement of 3-12
    - assigning 3-12, 4-202
    - illustration of chain 3-12
    - selection of 3-12
    - setting up alternate console chain 3-56
    - switching to 2-50
  - alternate console chain, illustration of 3-12
  - alternate CPU recovery (ACR) 1-27, 4-109
  - alternate nucleus, loading 1-3
  - alternate paths 4-30, 4-31, 4-211
  - apostrophes, how to use 4-8, 4-9
  - ASSIGN command
    - summary of 4-2
  - assigning
    - automatic message deletion 3-38, 4-48
    - command groups 4-203
    - console operating mode 3-58
    - master console 3-11
    - MCS consoles 3-9, 4-202
    - message levels 3-26, 3-58, 4-54
    - new master console 3-11
    - PFK tables 4-46
    - program function keys 3-65, 4-46
    - roll mode of message deletion 3-38, 4-48
    - roll-deletable mode of message deletion 3-38, 4-48
    - routing codes 3-25, 3-57, 4-203
    - secondary console 3-11
    - the hardcopy log to a device 3-6
  - audible alarm
    - description of 2-5
  - authorizing writers 1-13
  - automatic
    - rebuilding a global resource serialization ring 1-33
    - restart 1-11
    - volume recognition (AVR) 1-24
  - automatic mode of message deletion
    - defining 4-48
    - description of 3-32, 3-38
  - AVR (automatic volume recognition) 1-24
- ## B
- backup console 3-12
  - base address 4-179
  - base/displacement shorthand 4-179
  - binary compare in SLIP, definition 4-163
  - blank screen
    - responding to 2-48, 2-50
  - boxing devices 1-24
  - broadcast data set 1-10
  - broadcast messages
    - description of 3-26
    - recorded in hardcopy log 2-45, 3-6
    - routing of 4-53
  - buffer shortage 2-51, 2-52, 2-53
- ## C
- C (see CANCEL command)
  - cache, subsystem 4-125

- CANCEL action
  - performing 2-19, 2-32, 2-36, 2-37, 2-41, 2-43, 2-45
- CANCEL command 1-10
  - complete syntax 4-10-4-11
  - examples of 4-12
  - summary of 4-2
  - with an inactive console 2-49
- cancelling
  - action message retention facility 3-21, 4-42
  - automatic message deletion 3-38
  - conversational message deletion 4-48
  - job 1-10, 4-10
  - message routing 4-124
  - nonconversational message deletion 4-48
  - status displays for an inactive console 2-49
- catalog address space
  - communicating with 4-116
- CD (see CHNGDUMP command)
- CF (see CONFIG command)
- changing
  - action message retention facility status 4-42
  - color, intensity, and highlighting options 4-142
  - console characteristics 3-1
  - console operating mode 4-52
  - current console specifications 4-48
  - display area specifications 3-44, 4-37
  - dump mode and options 4-13
  - general WTO user exit IEAVMXIT 4-44
  - master console 4-205
  - MPFLSTxx member of SYS1.PARMLIB 3-20
  - number of allowed WTO message buffers 4-43
  - performance group of a job 1-11
  - PFK definitions 4-45
  - PFK tables 3-6
  - PFKTABxx member in effect 3-6
  - SMF options 4-132
  - SRM (system resources manager) parameters 4-141
  - status
    - of action message retention facility 4-42
    - of general WTO user exit IEAVMXIT 4-44
    - of secondary console 4-208
  - system commands a console can receive 4-202
  - system parameters 1-7
  - time interval for dynamic displays 3-51, 4-51
- changing information in the entry area 2-43
- channel path
  - placing online or offline 1-28, 4-29
  - reconfiguring 4-29
- checkpoint/restart 1-11, 1-12
- CHNGDUMP command 4-13
  - complete syntax 4-15-4-24
  - summary of 4-2
- clear dump data sets 4-97
- clear virtual I/O (CVIO) option 4-125
- clock, setting 4-131
- CLOCKxx member of SYS1.PARMLIB 4-141
  - OPERATOR parameter 1-4, 4-131
- CLPA (create link pack area) option 4-125
- cold start 1-3
- color
  - description of 2-5
- color options
  - changing 4-142
  - displayed 3-24, 4-86
  - on the 3278-2,2A,3,4, and 5 display station 2-20
  - on the 3279-2A,2C, and 3A display station 2-22
- command delimiter 2-38
- command groups 3-9
  - assigning 4-203
  - figure showing 3-10
- commands
  - checking PFK commands 2-40
  - correcting entry errors 2-38
  - entering
    - using keyboard 2-37
    - using selector pen 2-42
    - with program function keys 2-38
  - format of 4-9
  - syntax guidelines for 4-8
- COMMNDxx member of SYS1.PARMLIB 4-148
- communicating
  - among several consoles 1-7
  - with all terminal users 4-138
  - with other operators 4-136
  - with specified users 4-137
- composite console
  - assigning 2-2
  - definition of 2-2
- CONFIG command
  - complete syntax 4-25-4-34
  - examples of 1-28, 1-29, 1-30, 4-31, 4-33, 4-34
  - routed by MSGRT command 3-46, 4-122
  - summary of 4-2
- configuration
  - changing 1-12
  - displaying information about 4-87
  - displaying system 4-64
- considerations
  - for using FORCE command 4-101
  - for using shared DASD 1-15
- console
  - activity, lack of 2-49
  - alternate 3-12
  - cluster 3-54
    - setting up and using a master 3-55
  - control commands 3-9, 3-10
  - devices 2-1
  - failure 3-12
  - handling in error conditions 2-48
  - hardware errors 2-49
  - inactivity
    - cause of 2-49
    - response to 2-49
  - message area sizes 3-43
  - message roll rate 3-62
  - messages, record of 1-19
  - operating mode 3-15
  - operations 2-1
  - recovery 2-49, 2-54
  - responding to an inactive 2-49

- secondary 2-3, 2-4
  - definition of 2-4
  - uses of 2-4
- setting up alternate console chain 3-56
- switching 3-11, 3-12
- console characteristics
  - displaying 3-13
- console cluster, example of 3-54
- console inactivity
  - cause of 2-49
- console message backups
  - responding to 2-51
- console operating modes
  - changing from one mode to another 4-52
  - defining for consoles in console cluster 3-58
  - definitions of 3-15
  - effect on display screen 2-6
  - setting up message stream console 3-18
  - setting up status display console 3-18
- console specifications
  - changing 4-48
  - displaying 4-48
- consoles
  - devices supported by MVS/XA 2-2
  - displaying status of 4-69
  - hardware at IPL 1-1
  - MCS at IPL 1-2
  - NIP at IPL 1-1
- CONSOLxx member of SYS1.PARMLIB 3-55
  - ALTERNATE parameter 3-3, 4-202
    - example of 3-56, 3-64
  - AMRF parameter 3-4, 4-42
    - example of 3-21
  - AREA parameter 3-3, 4-37
    - example of 3-44, 3-59, 3-64
  - AUTH parameter 3-3, 4-202
    - example of 3-25, 3-56
  - CMDLEVEL parameter 3-4, 4-205
    - example of 3-8, 3-61, 3-64
  - commands that relate to 3-2
  - CON parameter 3-3, 4-48
    - example of 3-32, 3-62, 3-64
  - CONSOLE statement 3-3
  - DEFAULT statement 3-5
  - DEL parameter 3-3, 4-48
    - example of 3-39, 3-41, 3-62, 3-64
  - description of 3-2
  - DEVNUM parameter 3-3, 3-4, 4-205
    - example of 3-7, 3-11, 3-13, 3-16, 3-56, 3-64
  - general description of 2-4
  - HARDCOPY statement 3-4
  - INIT statement 3-4
  - LEVEL parameter 1-19, 3-3, 4-53
    - example of 3-27, 3-58, 3-64
  - MFORM parameter 3-3, 4-48
    - example of 3-28, 3-41
  - MLIM parameter 2-51, 2-53, 3-4, 4-43
  - MONITOR parameter on the CONSOLE
    - statement 3-3, 4-118
    - example of 3-52
  - MONITOR parameter on the INIT statement 3-4,
    - 4-118
      - example of 3-53
  - MPF parameter 2-23, 2-28, 2-34, 3-4, 3-5, 4-142
    - example of 3-20
  - MSGRT parameter 3-3, 4-122
    - example of 3-46, 3-62, 3-64
  - PFK parameter 3-4, 3-6, 4-142
    - example of 3-64, 3-69
  - PFKTAB parameter 3-3, 3-5
    - example of 3-64, 3-69
  - relationship with CONTROL command 3-3
  - RLIM parameter 2-51, 2-53, 3-4, 4-43
  - RNUM parameter 3-3, 4-48
    - example of 3-39, 3-62, 3-64
  - ROUTCODE parameter 3-3, 3-4, 4-202, 4-205
    - example of 3-7, 3-25, 3-57, 3-64
  - RTME parameter 3-3, 4-48
    - example of 3-39, 3-41, 3-62, 3-64
  - SEG parameter 3-3, 4-48
    - example of 3-36, 3-62, 3-64
  - system's use of 2-47
  - UEXIT parameter 3-4, 4-44
    - example of 3-24
  - USE parameter 3-3, 4-52
    - example of 3-58, 3-64
  - UTME parameter 3-3, 4-51
    - example of 3-51, 3-61
- contents compare in SLIP, definition 4-164
- CONTROL
  - E,SEG
    - controlling 3-35
- CONTROL command 4-35, 4-54
  - A
    - examples 3-40, 3-44, 3-45, 3-60, 4-37
    - syntax 4-37
  - action routed by MSGRT command 4-122
  - C,A
    - examples 4-39
    - syntax 4-38
  - C,CE
    - syntax 4-38
  - C,D
    - examples 3-49, 4-39
    - syntax 4-39
  - C,E
    - examples 3-36, 4-39
  - C,I
    - examples 4-39
    - syntax 4-38
  - D
    - examples 2-42, 3-37, 3-48, 3-49, 4-40
    - syntax 4-40
  - E
    - examples 2-42, 3-34, 3-37, 3-40, 3-48, 4-41
    - syntax 4-41
  - M,AMRF
    - examples 3-21, 4-42
    - syntax 4-42
  - M,MLIM

- examples 4-43
- syntax 4-43
- M,RLIM
  - examples 4-43
- M,UEXIT
  - examples 4-44
  - examples of 3-24
  - syntax 4-44
- N,PFK
  - examples 3-30, 3-31, 4-46
  - syntax 4-45
- Q
  - examples 2-49, 4-48
  - syntax 4-47
- relationship to CONSOLE statement in
  - CONSOLxx 3-3
- S
  - examples 3-28, 3-33, 3-36, 3-39, 3-41, 3-62, 4-50
  - syntax 4-48
- summary of 4-2
- T
  - examples 2-49, 3-51, 3-61
  - syntax 4-51
- V,LEVEL
  - examples 3-27, 3-58, 4-54
  - syntax 4-52, 4-53
- V,USE
  - examples 3-17, 3-58
  - syntax 4-52
- CONTROL command, recorded on hardcopy log 2-47
- controlling
  - external writer 1-13
  - format of messages 2-8, 3-28
  - global resource serialization complex 1-17, 4-214
  - hardcopy log 4-205
  - highlighting of messages 2-22, 2-23
  - intensity of messages 2-22, 2-23
  - JES3 through MCS consoles 3-22
  - jobs 1-10
  - machine check interruptions being monitored or recorded 4-107, 4-109
  - message backup 3-41
  - message color 2-22, 2-23
  - message deletion 3-40
  - message numbering 3-35, 4-40
  - messages in a console cluster 3-54
  - messages processed by MPF 3-19, 4-141
  - operating system 1-6
  - routing of status displays 3-46, 4-122
  - shared DASD 1-14
  - status displays 3-42
  - status displays on a status display console 3-18
  - system information recording 1-19, 2-45
  - system messages 3-19
  - time-sharing 1-9
- conversational mode
  - definition of 3-32
  - deleting messages manually 3-33
  - entering commands assigned to PFKs in 2-41

- entering commands with selector pen 2-43
- message deletion 3-32, 4-48
- PFKs in 3-29, 3-31, 4-45
- correcting command entry errors 2-38
- CPU
  - placing online or offline 1-28
- create link pack area (CLPA) option 4-125
- critical action messages
  - routing of 4-53
- critical eventual action messages
  - description of 3-26
  - displaying retained awaiting response 4-78
  - routing of 4-53
- CTC adaptor status
  - displaying information about 1-17, 4-87, 4-88
- cursor
  - control keys 2-12
  - description of 2-5, 2-12
  - moving 2-37
  - used to frame a display 3-48
- cursor control keys
  - description of 2-12
- cursor detect action
  - performing 2-37
- CVIO (clear virtual I/O) option 4-125

D

- D (see DISPLAY command)
- DASD (direct access storage device)
  - shared 1-14-1-17
- data definition (DD) statements 1-22
- data set status
  - displaying 3-52
- date, setting 4-141
- DCCF 2-56
- DDR (dynamic device reconfiguration) 1-25, 4-192, 4-193
- deactivating
  - action message retention facility 4-42
  - general WTO user exit IEAVMXIT 4-44
- debugging a SLIP trap 4-166
- deferred restart 1-11, 1-12
- defining
  - commands for PFKs 3-28, 3-31, 3-65, 4-45
  - console characteristics 3-1
  - message levels for consoles 4-53
  - PFK table 3-5, 3-65, 3-69, 4-45
  - routing codes for consoles 3-24, 3-57, 4-203
  - status display console 3-18
- degradation machine check interruptions, controlling
  - recording of 4-106, 4-107
- deleting
  - action messages for which action has been taken 3-32, 4-38
  - description of automatic message deletion 3-38
  - dump data sets 4-98
  - eventual action messages 4-38
  - finding appropriate deletion specifications 3-40



- immediate action messages 4-38
- message numbers 4-41
- messages 3-32, 3-39
  - from message stream consoles 3-18
  - from notices section of broadcast data set 4-140
  - in conversational mode 4-48
  - in nonconversational mode 4-48
  - manually in conversational mode 3-33
  - manually in nonconversational mode 3-34
- nonaction messages 3-32
- retained action messages 3-36, 4-38
- retained immediate or eventual action messages 4-38
- SLIP traps 4-175
- status displays from display areas 3-48
- verifying deletion request 3-34
- verifying specifications 4-48
- delimiter of commands 2-38
- demount messages, adding information to 3-53
- descriptor codes
  - definition of codes 1,2,3, and 11 3-20
  - routing by 4-53
- device
  - allocation 1-22
    - displaying 4-62
  - assignment 1-22
  - boxing 1-24
  - moving online or offline 4-209
  - status
    - displaying 4-62
- devices
  - supported and not supported by DDR 1-25-1-26
  - supported as MVS/XA consoles 2-3
- devnum parameter of system commands 1-23
- DEVSERV command
  - complete syntax of 4-55
  - examples of 4-56
  - use of 1-7
- direct access storage device (DASD)
  - shared 1-14-1-17
- direct address, in indirect addressing 4-177
- directing
  - command responses to specific consoles 3-62
- disabled console communications
  - replying to messages 4-128
- disabled console communications facility 2-56
- disabled loop 1-8
- displacement, in indirect addressing 4-177
- display areas
  - changing specifications for 3-42, 4-37
  - establishing 3-42
  - example of setting up 3-45
  - framing 3-48
  - illustration of two areas 3-45
  - routing status displays to 3-46
  - setting up 3-59
  - setting up for console cluster 3-59
- DISPLAY command 4-57, 4-92
  - A
    - examples 3-46
- ASM
  - examples 4-62
- C,K
  - examples 4-72
  - syntax 4-72
- CONSOLES
  - examples 2-52, 3-6, 3-27, 4-71
  - sample output 2-52, 3-13
  - syntax 4-69
- DMN
  - examples 4-85
  - syntax 4-85
- DUMP
  - examples 4-74
  - syntax 4-72
- examples 4-62
- GRS
  - examples 1-18, 1-34, 4-91
  - sample output of 1-17, 1-35
  - syntax 4-87
- JOBS
  - examples 4-77
  - syntax 4-75
- M
  - examples 4-68
  - syntax 4-64
  - use of 1-30
- MPF
  - examples 4-87
  - syntax 4-86
- PFK
  - example of output 3-69
  - examples of 2-40, 3-30
  - syntax 4-83
- R
  - examples 3-22, 3-36, 4-39, 4-81
  - syntax 4-78
- routed by MSGRT command 1-18, 3-46, 4-122
- SLIP
  - examples 4-84
  - syntax 4-84
- SMF
  - syntax 4-82
- summary of 4-3
- syntax 4-60
- T
  - syntax 4-84
- TS
  - examples 4-77
  - syntax 4-75
- U
  - examples 4-64
  - syntax 4-62
  - use of 1-7
- display console
  - compared with composite console 2-4
  - definition of 2-2
  - features on 2-4
- display screen areas

- on consoles 2-6
- displaying \*
  - action message retention facility status 4-42
  - action messages awaiting action 3-22
  - action messages not retained 3-24, 4-86
  - active jobs 1-12, 3-46
  - active units of work 4-75
  - color, intensity, and highlighting options 3-24, 4-86
  - configuration
    - of hardware 4-64
  - console characteristics 3-13
  - console operating mode 3-14, 4-52
  - console specifications 3-14, 4-48
  - CONTROL command options and functions 4-72
  - current system status 1-6
  - data set status 3-52
  - device status and allocation 4-62
  - display area specifications 3-45, 4-37
  - domain description table 4-85
  - dump data set status 4-72
  - dump options 4-72
  - extended storage status 4-66
  - general WTO user exit IEAVMXIT status 4-44
  - global resource serialization system and CTC
    - adapter information 1-18, 4-87
  - hardcopy information 3-6
  - hardcopy log information 4-71
  - hardware system area (HSA) location and length 4-66
  - information about hardcopy log 3-27
  - job information 1-6, 3-52, 4-197
  - logged-on time-sharing information 1-10
  - logged-on time-sharing users 1-10, 4-75
  - machine check interruptions being monitored or recorded 4-107
  - message deletion specifications 3-14, 3-41
  - message levels not assigned to any console 3-26
  - messages currently being suppressed 4-86
  - MPF message processing 3-24, 4-86
  - MSGRT routing instructions 3-47
  - number of active batch jobs 4-75
  - number of allowed WTO message buffers 3-14, 4-43
  - number of allowed WTOR message buffers 3-14, 4-43
  - page data set information 4-60
  - PFK definitions 2-40, 3-14, 4-83
  - real storage status 4-67
  - resource name lists (RNLs) 1-17
  - retained messages awaiting response 4-78
  - routing codes assigned to a console 4-69, 4-71
  - routing codes not assigned to any console 3-26, 4-71
  - SLIP trap information 4-84
  - SMF data 4-82
  - status of channel paths 4-55
  - status of consoles 3-14
  - status of consoles having message backlog 4-71
  - status of devices 1-7, 4-55
  - status of user exit IEAVMXIT 4-44
  - status of user exits 3-24, 4-86
  - suppressed messages 4-86
  - swap data set information 4-60
  - system activity 1-6, 4-75
  - system requests 4-78
  - time and date 2-49, 4-84
  - time interval for dynamic displays 4-51
  - TSO information 3-52
- DLOG 2-48
- domain description table
  - changing 4-144
  - displaying 4-85
- DS (see DEVSERV command)
- dump
  - data sets 4-97
  - modes and options 4-13
  - options
    - displaying 4-72
    - specifying 4-93, 4-133
- dump analysis and elimination (DAE)
  - parameters 4-141, 4-142
  - suppression of the dump 4-155
- DUMP command
  - complete syntax 4-93-4-96
  - examples of 4-95-4-96
  - summary of 4-3
- DUMPDS command
  - complete syntax 4-97-4-100
  - examples of 4-100
  - summary of 4-3
- duplicate lines 4-48
- dynamic device allocation 1-22
- dynamic device reconfiguration (DDR) 1-25, 4-192, 4-193
- dynamic display
  - stopping 3-52
- dynamic status displays
  - description of 3-42
  - establishing 3-49

E

- E (see RESET command)
- enabled wait 2-54
- end line, system checking for 2-49
- ENTER action
  - performing 2-36
- entering
  - commands 2-38
  - commands with keyboard 2-37
  - system parameters 4-130
  - with selector pen 2-43
- entry area
  - after deletion request 3-33
  - blanking out 2-45
  - changing information in 2-43
  - correcting errors in 2-43
  - description of 2-6
  - illustration of 3-43

- inserting characters in 2-44
- erasing
  - inline status display 3-49
  - out-of-line status display 3-48
  - screen 4-41
  - status displays 3-42, 3-48
- error code 1-3
- error events, trapping 4-147
- error messages
  - responding to 2-49
- error recovery 1-22
  - in SMF 1-20
- establishing
  - display areas 3-42
  - dynamic status displays 3-49
  - TRACK display in console cluster 3-61
- eventual action messages
  - deleting 4-38, 4-39
  - description of 3-26
  - displaying retained awaiting response 4-78
  - routing of 4-53
- examples of commands
  - ASM 4-62
  - CANCEL 4-12
  - CONFIG 1-15, 1-28, 1-29, 1-30, 4-31, 4-33, 4-34
  - CONTROL A 3-40, 3-45, 3-60, 4-37
  - CONTROL C,A 3-36, 4-39
  - CONTROL C,D 4-39
  - CONTROL C,E 3-36, 4-39
  - CONTROL C,I 4-39
  - CONTROL D 2-42, 3-37, 3-48, 3-49, 4-40
  - CONTROL E 2-42, 3-33, 3-34, 3-37, 3-40, 3-48
  - CONTROL M,AMRF 3-21, 4-42
  - CONTROL M,MLIM 4-43
  - CONTROL M,RLIM 4-43
  - CONTROL M,UEXIT 3-24, 4-44
  - CONTROL N,PFK 3-30, 3-31, 4-46
  - CONTROL Q 4-48
  - CONTROL S 3-28, 3-33, 3-36, 3-39, 3-41, 3-62, 4-50
  - CONTROL T 3-51, 3-61
  - CONTROL V,LEVEL 3-27, 3-58, 4-54
  - CONTROL V,USE 3-17, 3-58
  - DEVSERV P 4-56
  - DISPLAY A 3-46, 4-77-4-78
  - DISPLAY ASM 4-62
  - DISPLAY C,K 4-72
  - DISPLAY CONSOLES 2-52, 4-71
  - DISPLAY DMN 4-85
  - DISPLAY DUMP 4-74
  - DISPLAY GRS 1-17, 1-34, 4-91
  - DISPLAY JOBS 4-77
  - DISPLAY M 4-68
  - DISPLAY PFK 2-40, 3-30, 3-69
  - DISPLAY R 3-22, 3-36, 4-81-4-82
  - DISPLAY SLIP 4-84, 4-85
  - DISPLAY T 2-49
  - DISPLAY TS 4-77, 4-78
  - DUMP 4-95-4-96
  - DUMPDS 4-100
  - FORCE 1-21, 4-103

- LOG 4-105
- MODE 4-108, 4-110
- MODIFY 4-44, 4-111, 4-112, 4-113, 4-115, 4-117
- MONITOR 3-52, 3-53, 4-119
- MOUNT 4-121
- MSGRT 3-47, 3-63, 4-124
- PAGEADD 4-126
- REPLY 4-129
- RESET 4-134
- SEND 4-9, 4-136-4-137-4-138-4-140
- SET 3-30, 4-142, 4-143
- SETDMN 4-145
- SETSMF 4-146
- SLIP 4-154-4-174, 4-181
- START 4-183, 4-184, 4-185-4-186
- STOP 4-189
- STOPMN 4-190
- STOPTR 4-191
- SWAP 4-192, 4-193
- TRACE 4-196
- TRACK 2-44, 3-49, 4-198
- VARY 1-17, 1-35, 3-7, 3-8, 3-11, 3-13, 3-26, 3-57, 3-58, 3-61, 4-204-4-212, 4-214
- VARY HARDCPY 3-9, 4-208
- WRITELOG 4-216
- extended highlighting
  - description of 2-5
- extended storage
  - display status of 4-69
  - displaying status of 4-66
  - placing online or offline 1-29
- external interrupt key, use in master console
  - failure 2-54
- external writer
  - controlling 1-13
  - specifying data set selection criteria for 4-114
  - starting 4-183

## F

- F (see MODIFY command)
- failing console 2-50, 3-12
- failing CTC adapter in global resource serialization ring 1-33
- failing global resource serialization ring 1-31
- failing master console 2-54
- flagged messages
  - defined 3-34
  - deleting 3-34, 4-41
  - removed under automatic mode 3-38
- FORCE command 1-10
  - cautions about using 4-101
  - complete syntax 4-101-4-103
  - examples of 1-21, 4-103
  - summary of 4-3
- FORCE operand of CONFIG command
  - cautions about using 1-25, 4-30
- FORCE operand of VARY command
  - cautions about using 1-25, 4-211-4-212

- format of messages
  - controlling 2-8, 3-28
- framing
  - status displays 3-48, 4-40
  - with the CONTROL command 3-48, 4-40
  - with the cursor 3-48
  - with the selector pen 3-48
- full-capability mode
  - changing to output-only mode 4-52
  - console in 3-15
  - console in console cluster 3-54
  - cursor on console 2-5
  - defined 3-15
  - display screen for, illustration 2-7
  - keyboard on console 2-5
  - operations on display consoles 2-36

**G**

- G (see SWAP command)
- generalized trace facility (GTF) 1-21
  - starting 4-184
- global resource serialization
  - complex
    - building 1-5, 1-17
    - monitoring and maintaining 1-17
  - CTC adapter failure 1-33
  - DISPLAY GRS command 1-18, 4-87
  - displaying information about 1-17, 4-87
  - example of two-system ring 1-18
  - listing of major resource names 4-87
  - ring rebuilding 1-31
  - VARY GRS command 4-214
- GMT (Greenwich mean time) 4-131, 4-132
  - displaying 4-84
- Greenwich mean time (GMT) 4-131, 4-132
  - displaying 4-84
- GRSCNFxx member of SYS1.PARMLIB 1-31, 1-33
- GTF (generalized trace facility) 1-21
  - starting 4-184

**H**

- H (see HOLD command)
- HALT command 1-36
  - complete syntax 4-104
  - summary of 4-4
- halting
  - printing of status display 4-39
  - status display in progress 3-49, 4-39
  - TRACK display 4-191
- handling consoles in error conditions 2-48
- hard machine check interruptions, controlling
  - monitoring of 4-106, 4-109
- hardcopy devices assigned by MVS/XA 2-47
- hardcopy log
  - assigning to a device 3-6
  - changing status of 4-206

- CONTROL command recorded on 2-47
  - controlling 2-45, 4-205
  - controlling in CONSOLxx 3-4
  - description of 1-19
  - device, bypassing 3-11
  - display of messages only appearing on 4-71
  - displaying information about 3-6, 4-71
  - format of first line 2-45
  - format of split first line 2-46
  - format of subsequent lines of multi-line messages 2-46
  - keeping TRACK display out of 3-61
  - messages appearing on 1-19, 2-47, 3-7, 3-27, 4-53
  - minimum routing codes for 4-207
  - purpose of 2-45, 3-6
  - routing commands to 3-8
  - routing messages to 3-7
  - type of system-assigned 2-47
- hardware console
  - definition of 1-1
  - processing MVS messages at 2-56
- hardware malfunctions 1-24
- highlighting of messages
  - changing 4-142
  - displaying options in effect 3-24, 4-86
  - on the 3277-2 display station 2-14
  - on the 3278-2,2A,3,4, and 5 display stations 2-20
  - on the 3279-2A,2C, and 3A display station 2-22
  - on 3290 information panel 2-28
- HOLD command
  - summary of 4-4
- hot I/O detection 1-24

**I**

- I (see SWITCH command)
- I/O configuration
  - loading 1-3
- I/O control commands 3-9, 3-10
- IEAABD00 4-13, 4-17, 4-22
- IEACMDxx member of SYS1.PARMLIB 4-148
- IEADMP00 4-13, 4-17, 4-18, 4-22
- IEADMR00 4-13, 4-17, 4-18, 4-22
- IEAICSxx member of SYS1.PARMLIB 4-142
- IEAOPTxx member of SYS1.PARMLIB 4-142
- IEASLPxx member of SYS1.PARMLIB 4-148
- IEASYSxx member of SYS1.PARMLIB 1-4, 3-29
- IEAVMXIT user exit
  - changing status of 4-44
  - description of 3-23
  - displaying status of 3-24, 4-44, 4-86
  - status when MPF is off 3-19
- IEA019A 4-31
- IEA101A 1-3, 1-4, 3-2
- IEA116A 4-131
- IEA180 3-6
- IEA230E 2-53
- IEA231A 2-53
- IEA347A 1-4

IEA404A 2-51  
 IEA405E 2-51  
 IEA406I 2-53  
 IEA420A 2-54, 2-55  
 IEA421E 4-31  
 IEA886A 1-4, 4-131  
 IEA888A 1-4, 1-5, 4-131, 4-132  
 IEA903A 4-131, 4-132  
 IEECVXIT user exit 3-19  
 IEE025I 4-210  
 IEE041I 3-37  
 IEE082I 4-127  
 IEE094D 4-93, 4-95, 4-96, 4-133  
 IEE100E 4-30  
 IEE104I 4-75  
 IEE105I 4-75  
 IEE106I 4-63  
 IEE112I 4-78  
 IEE131D 4-30  
 IEE136I 4-84  
 IEE143I 2-50, 2-55  
 IEE144I 4-42, 4-43, 4-44  
 IEE152I 1-5, 2-10, 2-16, 3-17  
     id in the instruction line 2-21, 2-23, 2-34  
 IEE157E 3-33  
 IEE159E 2-14, 2-20, 2-21, 2-22, 2-33, 2-34, 3-39  
 IEE160I 2-21, 2-34, 3-42  
 IEE161I 2-21, 2-34  
 IEE163I 2-17, 2-21, 2-34  
 IEE164I 2-13, 2-18  
 IEE170E 2-22, 2-49  
 IEE171E 2-22, 2-49  
 IEE189I 4-86  
 IEE235I 2-40  
 IEE249I 4-69  
     sample of 2-52  
 IEE250I 4-54  
 IEE312I 4-78  
 IEE313I 4-210, 4-211  
 IEE334I 2-15, 3-34, 4-104  
 IEE354I 4-132  
 IEE357A 4-132  
 IEE360I 1-21  
 IEE379I 1-15  
 IEE450I 4-63  
 IEE507D 4-30  
 IEE508E 4-30  
 IEE712I 4-211  
 IEE719I 1-15  
 IEE721I 2-42  
 IEE722I 2-42  
 IEE735I 4-84  
 IEE744I 2-55  
 IEE752I 4-127  
 IEE800D 4-212  
 IEE828E 3-27  
 IEE922I 4-51  
 IEE956A 4-132  
 IEE967I 4-82  
 IEE980I 1-21  
 IEF006I 1-12  
 IEF008I 1-12  
 IEF209I 1-12  
 IEF225D 1-11  
 IEF238D 1-22, 1-25  
 IFASMFDP 1-20  
 IFB010D 4-131  
 IGF500D 4-192  
 IGF502E 4-192  
 IGF509D 4-192  
 IKJ024D 4-113  
 IKT010D 4-189  
 immediate action messages  
     deleting 4-38, 4-39  
     displaying retained awaiting response 4-78  
     routing of 4-53  
 inactive console  
     characteristics of 2-49  
     responding to 2-49  
 indirect address 4-177  
 indirect addressing  
     DATA parameter 4-180  
     LIST, SUMLIST, and TRDATA parameters 4-180  
 indirection indicator, in indirect addressing 4-177  
 information panel (see 3290 information panel)  
 informational commands 3-9, 3-10  
 informational messages  
     description of 3-26  
     routing of 4-53  
 initial program loader (IPL) 1-3  
 initialization 1-3  
     from shared device 1-14  
 initializing the system software 1-3  
 installation performance specification (IPS)  
     changing 4-142  
     displaying the name of the current 4-85  
 instruction line  
     description of 2-6  
     illustration of 3-43  
 instructions for message routing, establishing 4-122  
 intensity  
     description of 2-5  
 intercepting system events 4-147  
 interrupt key, use in recovering messages 2-51  
 invalid wait state 1-8  
 IODEVICE statement in MVSCP 2-1  
 IOS109I 1-24  
 IOS110A 1-24  
 IOS111A 1-24  
 IOS112A 1-24  
 IOS120A 1-14  
 IPL (initial program loader)  
     procedure 1-3  
     program 1-3  
     volume 1-16  
 IPS (installation performance specification)  
     changing 1-7, 4-142  
     displaying the name of the current 4-85  
 IRA200I 4-125  
 IRA201I 4-125  
 ISG005I 1-6

ISG006I 1-5, 1-6  
ISG009D 1-5, 1-6  
ISG011I 1-35  
ISG013I 1-35  
ISG020I 1-17, 1-35  
ISG022E 1-33  
ISG023E 1-31, 1-33  
ISG024I 1-31, 1-33  
ISG025E 1-31, 1-33, 1-34  
ISG046E 1-33, 1-34

## J

JES event trace facilities 1-19  
JES2  
    automatic command facility 3-63  
    commands used in console cluster 3-63, 3-66, 3-67  
    controlled by MCS consoles 2-2  
    example of starting 4-183  
    initialization data set 3-63  
    message routing codes 3-24  
    procedure for starting 1-5  
JES3  
    action message retention facility 3-21  
    commands used for console recovery 2-56  
    controlling through MCS consoles 3-22  
    DSP names as keynames 3-22  
    managed through MCS consoles 2-2  
    message routing codes 3-24  
    procedure for starting 1-5  
    using MCS consoles 2-2  
    varying consoles online 4-209  
job information  
    displaying 3-52  
job parameters, modifying 4-111

## K

K (see CONTROL command)  
keyname, displaying messages by 4-80  
keyword relationships for SLIP commands 4-152  
keyword summary for SLIP command 4-176  
keywords for SLIP SET commands 4-153

## L

L (see LOG command)  
link pack area expansion 1-12  
listing notices section of broadcast data set 4-140  
LLA (LNKLST lookaside)  
    defined 4-187  
    modifying LNKLST directories 4-44, 4-116  
LLA address space  
    starting 4-187  
    terminating 4-189  
LNKLST lookaside (LLA)  
    modifying directories 4-44, 4-116

starting LLA address space 4-187  
terminating address space 4-189  
load  
    function 1-3  
    initial microprogram (IML) 1-2  
    process 1-3  
loading  
    alternate nucleus 1-3  
    I/O configuration 1-3  
    parameter field 1-3  
location operand  
    defined 3-46, 4-60  
    use in routing status displays 3-46  
locked keyboard  
    responding to 2-50  
LOG command 1-20  
    complete syntax 4-105  
    example of 4-105  
    summary of 4-4  
logical console, defined 2-23  
logical reconfiguration  
    definition of 1-26  
loop, disabled 1-8  
lowercase, rules about using 4-8, 4-9

## M

M (see MOUNT command)  
machine check interruptions  
    controlling recording of 4-106  
machine checks  
    controlling recording and monitoring status  
        of 4-107-4-110  
    displaying recording and monitoring status  
        of 4-107  
making console assignments 3-9  
manual  
    rebuilding a global resource serialization ring 1-32  
master catalog 1-4  
    specifying an alternate 1-4  
master console 2-3  
    assigning 3-11  
    bypassing 3-11  
    changing assignment to another console 4-205  
    cluster, setting up and using 3-55  
    definition of 1-1  
    description of 2-3  
    reassigning 3-11  
    recovering from failure 2-54  
    routing messages to 3-25  
    sending messages to 3-25  
    switching assignment to another console 2-54, 3-11  
master console-only commands 3-9, 3-10  
master trace 1-21, 4-196  
match 4-147  
MCS consoles  
    at IPL 1-2  
    attributes of 2-4

- defining in CONSOLxx member of
  - SYS1.PARMLIB 2-4
  - subsystems' use of 2-2
- MEMTERM error 4-167, 4-168
- message area
  - description of 2-6
  - illustration of 3-43
  - sizes on different consoles 3-43
- message backup
  - avoiding 3-41
  - displaying messages 4-47
- message buffer shortage 2-52
- message deletion specifications 3-32-3-39
  - defining 3-40, 4-48
  - displaying 3-41, 4-48
- message format
  - changing 3-28
- message handling portion of the control program 2-49
- message identifier 2-8
- message levels
  - assigning to consoles 3-26, 3-58, 4-53
  - displaying message levels not assigned to any console 3-26
  - relationship with routing codes 3-27
- MESSAGE macro in JES3 3-22
- message numbering
  - aid in manual message deletion 3-37
  - controlling 3-35, 4-40
  - displaying consecutive numbers 3-37
  - stopped by automatic message deletion 3-37
  - when not recommended 3-37
- message presentation
  - controlled by MPFLSTxx 3-5
  - controlled by MPFLSTxx member 3-20, 4-142
  - displaying options in effect 3-24, 4-86
  - on the 3277-2 display station 2-14
  - on the 3278-2,2A,3,4, and 5 display station 2-20
  - on the 3279 models 2B and 3B 2-22
  - on the 3279 models 3X, S3G, 2X, and S2B 2-22
  - on the 3279-2A,2C, and 3A display station 2-22
  - on 3290 information panel 2-28
- message processing facility (MPF)
  - changing messages suppressed 3-19, 4-142
  - controlled by MPFLSTxx 3-5
  - controlling message presentation 3-20
  - description of 3-19
  - displaying MPF options in effect 3-19, 4-86
  - displaying status of 3-24, 4-86
  - ending message suppression 3-19, 4-142
  - messages unable to be suppressed by 3-20
  - writing suppressed messages to hardcopy log 2-47
- message roll rate
  - setting of 3-62, 4-48
  - temporarily suspending 3-37
- message routing codes 3-24
- message stream mode
  - console in console cluster 3-54
  - defined 3-15
  - deleting messages from consoles 3-18
  - display screen for, illustration 2-7
- message suppression 3-5
- message traffic, lack of 2-49, 2-55
- messages
  - avoiding loss of 2-51
  - controlling format of 3-28
  - controlling status displays 3-42
  - deleting 3-32, 4-38, 4-39
    - adjusting deletion specifications 3-41
    - description of automatic message deletion 3-38
  - format of 2-8
  - processed by IEAVMXIT 3-23, 4-44
  - processed by MPF 3-19, 4-141
  - processed by user exits 3-23
  - recovering lost 2-50
  - responding to error 2-48, 2-49
  - sent from one operator to another 4-136
  - suppressed by MPF 3-20, 4-141
  - types not requeued 2-50
- MN (see MONITOR command)
- MODE command
  - complete syntax 4-106-4-110
  - examples of 4-108, 4-110
  - summary of 4-4
- MODIFY command 1-9, 1-11, 1-13
  - communicating with catalog address space 4-116
  - complete syntax 4-111-4-117
  - examples of 4-44, 4-111, 4-112, 4-113, 4-115, 4-117
  - external writer
    - causing to pause 4-115
    - specifying data set selection criteria for 4-114
  - job parameters 4-111
  - replacing LNKLST directories 4-116
  - specifying data set selection criteria for external writer 4-114
  - starting and stopping TSO/TCAM
    - time-sharing 4-112
  - summary of 4-4
  - TSO/TCAM
    - starting 4-112
    - stopping 4-112
  - TSO/VTAM, modifying 4-113
- modifying
  - external writer's criteria for selecting data sets 1-13
  - job parameters 1-11, 4-111
  - LNKLST directories 4-116
  - SLIP traps 4-174
  - TSO/VTAM 1-10
- MONITOR command
  - action routed by MSGRT command 4-122
  - complete syntax 4-118
  - example of 3-52, 3-53, 4-119
  - summary of 4-4
  - tracking jobs 1-6
  - tracking the terminal users 1-10
- monitoring of machine check interruptions 4-106
- mount
  - characteristics 1-14, 1-15
    - for shared DASD 1-15
  - messages 3-53
- MOUNT attribute 1-23
- MOUNT command 1-16, 1-23

- complete syntax 4-120
- example of 4-121
- summary of 4-4
- unsuccessful mount 4-11
- mounting tape drives
  - with a partitionable processor 1-27, 4-29
- moving
  - online or offline
    - channel paths 4-25, 4-30
    - CTC adaptors 1-33
    - I/O device 4-209
    - I/O path 4-212
    - processors 4-25
    - real storage sections 4-25
    - resources 1-28
    - secondary console 4-208
    - Vector Facilities 4-25
  - status display to next frame 3-48
- MPFLSTxx member of SYS1.PARMLIB 2-34, 3-23
  - changing 3-20, 4-142
  - defining highlighting 2-22, 2-23, 2-28, 2-34
  - description of 3-5
  - setting 4-142
- MR (see MSGRT command)
- MSGRT command
  - checking current routing instructions 3-47
  - complete syntax 4-122-4-124
  - description of 1-7
  - examples of 3-47, 3-63, 4-124
  - routing action of CONTROL command 4-122
  - routing displays to other consoles 3-46, 4-122
  - stopping effects of 3-47, 4-124
  - summary of 4-4
- multi-console mode, defined 2-23
- multichannel hardware switch 1-14
- multiple-console
  - configuration 2-3
  - support
    - for MCS consoles 2-3
- multiple-console support (MCS)
  - consoles at IPL 1-2
- multiple-line messages at console failure 3-13
- multiprocessor
  - definition of 1-27
- MVS configuration program
  - defining devices 2-1
- MVS/XA consoles, devices supported as 2-3
- MVSCP
  - defining devices 2-1, 3-12

N

- NIP (nucleus initialization program) processing 2-53
- NIP console 1-3
  - definition of 1-1
- no-consoles condition 2-54, 2-55, 2-56
  - notification of 2-54
  - recovering from 2-54
- no-match condition for SLIP processing 4-148

- NODUMP dump mode 4-13
- nonconversational mode
  - definition of 3-32
  - deleting messages manually 3-34
  - entering commands assigned to PFKs in 2-41
  - entering commands with selector pen 2-43
  - message deletion 3-32, 4-48
  - PFKs in 3-29, 3-31, 4-45
- nucleus
  - alternate 1-3
    - loading 1-3
  - identifier 1-3
  - primary 1-3
  - standard 1-3
- null reply 4-129

O

- offline devices 2-1
- offline status
  - placing console in 3-11
- OLTEP (online test executive program) 1-22
- online devices 2-1
- online test executive program (OLTEP) 1-22
- operating mode (see console operating mode)
- operator information area
  - description of 2-7
  - illustration of 3-43
- operator-requested DDR 4-192
- options for CHNGDUMP command 4-19
- output-only mode
  - changing to full-capability mode 4-52
  - console in 2-2
  - defined 3-15
  - used for hardcopy log 2-2
- OVER dump mode 4-13

P

- P (see STOP command)
- PA (see PAGEADD command)
- page data sets 1-17
  - adding 4-125
  - displaying information 4-60
- PAGEADD command
  - complete syntax 4-126
  - examples of 4-126
  - summary of 4-5
- PAGNUM system parameter 4-125
- parameters
  - job entry subsystem 1-2
  - system 1-3
- partitioning a multiprocessor 1-27, 1-28
- PASSWORD data set 1-17
- path, placing online or offline 4-212
- PER (program event recording) events, trapping 4-147
- performance group 4-134
- performance objective 4-145



- performing
  - system restart function 1-8
- performing basic keyboard actions 2-36
- PFK display line
  - example of 2-7
  - on 3277 models 1 and 2 2-9
- PFK tables
  - defining 3-5
  - definition of 3-6, 3-28, 4-45
  - description of 3-5
  - displaying definitions in 2-40, 4-83
  - example of defining 3-65, 3-69
  - example of defining commands in 3-29
- PFKs (program function keys)
  - defining commands for 3-28, 3-31, 3-65, 4-45
  - definition errors 2-39
  - definitions at system generation 2-39
  - display line on certain consoles 2-6
  - displaying commands defined for 2-40, 4-83
  - displaying numbers on 2250-3 display unit 4-40
  - displaying numbers on 3277-2 display station 2-42, 4-40
  - entering commands with 2-38
  - in conversational mode 3-29, 3-31
  - initial definitions 2-39
  - keyboard 2-5
  - summary of keys for console cluster 3-69
- PFKTABxx member of SYS1.PARMLIB
  - changing 3-6, 4-142
  - description of 3-5
  - example of 2-40, 3-67
  - example of coding 3-29
  - example of defining 3-65
  - PFK parameter 3-5
- physical reconfiguration
  - definition of 1-26
- physically-partitioned mode
  - definition of 1-28
- placing
  - channel path online or offline 4-29
  - console in offline status 3-11
  - I/O device online or offline 4-209
  - I/O path online or offline 4-212
  - processor online or offline 4-26
  - resources online or offline 4-208
  - secondary console online or offline 4-208
  - storage online or offline 4-27
- PM (see STOPMN command)
- pointer, in indirect addressing 4-177
- primary nucleus 1-3
- primary SMF data set 1-20
- printing the system log 1-20, 2-48, 4-216
- processor
  - placing online or offline 4-26
  - reconfiguring 4-26
- processor complex
  - definition of 1-26
- program event recording (PER) events, trapping 4-147
- program function keyboard
  - description of 2-5
  - on 3180 display station 2-32
  - on 3277 model 2 2-13
  - on 3290 information panel 2-27
- program function keys (see PFKs)
- program status word (PSW) 1-3, 1-8
- pseudo-master console
  - status displayed 4-71
- PSW (program status word) 1-3, 1-8
- PT (see STOPTR command)
- PURGE command
  - summary of 4-5

**Q**

- QSAM-supported devices 1-13
- quick start 1-3
- QUIESCE command 1-8, 1-36
  - complete syntax 4-127
  - summary of 4-5
- quiescing
  - global resource serialization system 1-18, 4-215
  - system 1-36

**R**

- R (see REPLY command)
- reading a job from a direct access volume or tape 4-183
- real storage
  - display status of 4-69
  - displaying status of 4-67
  - job 1-12
  - moving online or offline 1-28, 4-27
  - specifying configuration 1-2
- REASON keyword on the SLIP command 4-161
- reassigning the master console 3-11
- reconfiguration
  - definition of 1-26
- reconfiguring
  - channel path 4-29
  - dynamic device (DDR) 1-25
  - I/O device 4-209
  - I/O path 4-212
  - processor 4-26
  - resources 1-26-1-30, 4-208
  - storage 4-27
- recording
  - of machine check interruptions, controlling 4-106
  - system events 1-21
- recovering
  - from inactive console condition 2-49
  - from master console failure 2-54
  - from no-consoles condition 2-54
  - lost messages 2-50
- recovery
  - alternate CPU (ACR) 1-27
  - error 1-22
  - for global resource serialization problems 1-18, 1-31

- hot I/O 1-24
- RELEASE command
  - summary of 4-5
- removing
  - channel path 4-29, 4-30
  - data from entry area 2-45
  - dump data sets 4-99
  - information from console screen 4-41
  - last path to allocated, TP, in-use, or reserved device 4-29, 4-30
  - messages from the screen 3-38
  - MSGRT routing instructions 3-47, 4-122
  - system from the global resource serialization complex 4-214
  - tape or DASD volumes 4-199
- REPLY command
  - complete syntax 4-128-4-133
  - examples of 4-129
  - summary of 4-5
  - to invoke automatic restart 1-11
- replying
  - during system recovery 4-129
  - to disabled console communications messages 4-128
  - to set the TOD (time-of-day) clock 4-131
  - to specify dump options 4-93, 4-133
  - to specify SMF options 4-132
  - to specify system parameters 4-130
  - to system information requests 4-129
  - to system requests during recovery processing 4-129
  - WAIT for shared DASD 1-14
  - with dump options 4-133
- rerouting message queues 4-47
- reserve conversion RNL
  - displaying contents of 1-17, 4-87
- reserve/release function 1-14
- reserving a volume 1-23
- RESET command
  - complete syntax 4-134
  - example of 4-134
  - summary of 4-5
- resetting
  - dump options to initial values 4-22
  - dump type to ADD 4-22
  - system dump options list 4-17
  - time and date 4-131
- resource allocation 1-22
- resource contention information
  - displaying 1-17, 4-87, 4-89
- resource name lists (see RNLs)
- responding to
  - blank screen 2-48, 2-50
  - console message backups 2-51
  - error conditions with consoles 2-48
  - error messages 2-48, 2-49
  - hardware errors 2-49
  - inactive console condition 2-49
  - locked keyboard 2-50
  - master console failure 2-54
  - no-consoles condition 2-54
  - system errors 2-48
  - WTO buffer shortages 2-51
  - WTOR buffer shortages 2-53
- restart
  - automatic 1-11
  - deck 1-12
  - deferred 1-12
  - system 1-8
    - reasons on system control (SYSCTL) frame 1-8
- RESTART key 1-8
- restarting
  - global resource serialization ring 1-32, 1-34, 1-35, 4-214
  - job 1-11
  - SMF 1-21
  - system 1-8
- restoring
  - screen after hardware error 2-49
- reverse video
  - description of 2-28
  - on the 3279 models 3X, S3G, 2X, and S2B 2-22
  - on the 3290 information panel 2-28
  - on 3279 models 2B and 3B 2-22
- RNLs
  - displaying contents of 1-17, 4-87
- roll mode of message deletion
  - defining 4-48
  - description of 3-32
  - for action messages 3-41
  - for consoles in a console cluster 3-62, 3-67
- roll-deletable mode of message deletion
  - defining 3-41, 4-48
  - description of 3-32
  - for consoles in a console cluster 3-62
  - on message stream consoles 3-18
- routing
  - action of CONTROL command 4-122
  - action of MONITOR command 4-122
  - broadcast messages 4-53
  - critical action messages 4-53
  - critical eventual action messages 4-53
  - eventual action messages 4-53
  - immediate action messages 4-53
  - informational messages 4-53
  - messages 3-25
    - by descriptor code 4-53
    - by message level 4-53
    - by routing codes 3-24, 4-202
  - status displays 3-46-3-47, 4-122
  - system failure messages 4-53
  - using MSGRT command 4-122
  - WTO messages 4-53
- routing codes
  - assigning default 3-5
  - assigning to a console 3-25, 3-57, 4-202
  - description of 3-25
  - displaying information about consoles
    - receiving 4-71

- displaying routing codes not assigned to any console 3-26
- for hardcopy log 4-207
- listing of 3-24
- relationship with message levels 3-27

routing messages

- to hardcopy log 3-7

## S

S (see START command)

saving messages in broadcast data set 4-139

SCHEDxx member of SYS1.PARMLIB 4-196

scope of message deletion, defining 3-35

scratch volume 1-23

screen areas

- on consoles of different operating modes 2-6

screen characters

- description of 2-8

screen format

- illustration of 3-43

SD (see SETDMN command)

SDUMP dump type 4-13

SE (see SEND command)

secondary console 2-3, 2-4

- assigning 3-11

- definition of 2-4

- description of 2-3

- placing online or offline 4-208

- uses of 2-4

secondary SMF data set 1-20

selector pen

- defining commands for 3-29

- deleting messages using 3-32

- description of 2-5

- entering commands in conversational mode 2-43

- entering commands in nonconversational mode 2-43

- entering commands with 2-42

- on 3277 model 2 2-13

- use of 2-37

- used to frame a display 3-48

SEND command 1-7

- complete syntax 4-136-4-140

- examples of 4-9, 4-136-4-137-4-138-4-140

- for time-sharing communication 1-10

- summary of 4-5

sending

- messages to consoles 4-202

- messages to other operators 4-136

- messages to other users 4-137, 4-138

- status displays to another console 4-122

sending commands

- to hardcopy log 3-8

serialization, global resource (see global resource serialization)

SET command

- complete syntax 4-141-4-143

- examples of 3-20, 3-31, 4-142-4-143

- relationship to CONSOLE statement in

  - CONSOLxx 3-4

- summary of 4-6

- used by REPLY 4-131

SETDMN command

- complete syntax 4-144-4-145

- examples of 4-145

- summary of 4-6

SETSMF command

- complete syntax 4-146

- example of 4-146

- summary of 4-6

setting

- dump modes and options 4-15

- instruction fetch SLIP trap 4-151

- non-PER SLIP trap 4-150

- storage alteration SLIP trap 4-152

- successful branch SLIP trap 4-151

- time and date 1-4, 1-7

- TOD (time-of-day) clock 4-131

shared DASD

- operating guidelines 1-15

- option 1-14

sharing data sets 1-16

side

- definition of 1-28

- displaying resources in 4-67

- displaying status of 1-30

single-image mode

- definition of 1-28

SL (see SLIP command)

SLIP command

- complete syntax 4-149-4-181

- examples of 4-154-4-174, 4-181

- IF or SB PER trap 4-151

- indirect addressing 4-177

- keyword relationships 4-152

- keyword summary 4-176

- non-PER trap 4-150

- SA PER trap 4-152

- SET examples 4-173

- summary of 4-6

SLIP keyword summary 4-176

SLIP trap information, displaying 4-84

SMF

- description of 1-20

- displaying data 4-82

- restarting 1-21

- specifying options 4-132

- switching data sets 1-21, 4-194

SMFPRMxx member of SYS1.PARMLIB 1-20, 4-142, 4-146

special screen characters

- description of 2-8

- on console screens 2-17

- on the 3180 model 1 2-31

- on the 3277-2 display station 2-12

- on the 3278 display station 2-17

- on the 3279 display station 2-17

specifying

- dump options 4-133
- shared DASD mount characteristics 1-15
- SMF options 4-132
- system parameters 4-130
- SRM (system resources manager)
  - changing parameters 4-141
- standard nucleus 1-3
- START command 1-9, 1-10, 1-13, 1-21
  - complete syntax 4-182-4-187
  - DEALLOC procedure 1-16
  - examples of 4-183, 4-184, 4-185-4-186
  - summary of 4-6
- starting
  - external writer 1-13, 4-183
  - GTF (generalized trace facility) 4-184
  - job 1-10, 4-182
  - LLA address space 4-187
  - time-sharing 1-9
  - TSO/TCAM 4-112
  - TSO/VTAM 4-186
- status commands
  - controlled by TRACK command 4-197
- status display areas
  - establishing 3-42
- status display mode
  - console in console cluster 3-54
  - defined 3-15
  - display screen for, illustration 2-7
  - setting up a console 3-18
- status displays
  - at console failure 2-50
  - controlled by CONFIG command 1-7, 4-25
  - controlled by DISPLAY command 1-6, 4-57
  - controlled by TRACK command 1-6
  - controlling 3-18, 3-42-3-52, 4-122
  - description of 3-42
  - erasing 3-42, 3-48
  - routing to consoles 3-46, 4-122
- STOP command 1-9, 1-10, 1-21
  - complete syntax 4-188-4-189
  - examples of 4-189
  - summary of 4-6
- stop routine 1-10
- STOPMN command
  - action routed by MSGRT command 4-122
  - complete syntax 4-190
  - example of 4-190
  - stopping monitoring of terminal use 1-10
  - summary of 4-7
- stopping
  - dynamic display 3-52
  - I/O on a channel path quickly with FORCE operand of CONFIG 4-30
  - job 1-10
  - message routing 4-124
  - monitoring of terminal use 1-10
  - status display 4-39
  - system 1-36
  - time-sharing 1-9
  - TRACK display 4-191
  - TSO/TCAM 4-112
  - STOPTR command
    - action routed by MSGRT command 4-122
    - complete syntax 4-191
    - examples of 4-191
    - summary of 4-7
    - use of 1-6
  - storage
    - placing online or offline 4-27
    - reconfiguring 4-27
  - subsystem cache 4-125
  - summary of system commands 4-1-4-7
  - suppressing
    - messages from consoles 3-20
  - suspending
    - message roll rate 3-37
  - SVC dump 4-152, 4-154, 4-155, 4-156, 4-157, 4-158, 4-159
  - SWAP command 1-25
    - complete syntax 4-192
    - examples of 4-192, 4-193
    - summary of 4-7
  - swap data sets
    - adding 4-125
    - displaying information 4-60
  - swapping devices of unlike densities 1-25
  - SWITCH command 1-21
    - complete syntax 4-194
    - summary of 4-7
  - switching
    - console 2-51, 2-54, 3-12
  - symbolic group names 1-23
  - syntax of commands, guidelines 4-8
  - SYSABEND dump type 4-13
  - SYSCTL (CC012) frame 1-3
  - SYSLOG 2-47, 2-48, 2-53, 3-6
  - SYSMDUMP dump type 4-13
  - SYSOUT 1-13
    - defaults for external writer 1-13
  - system
    - activity recording 2-45
    - activity, displaying 4-75
    - control commands 3-9, 3-10
    - dump options list 4-13
    - error conditions 2-48
    - information recording, controlling 1-19
    - monitoring 1-22
    - parameters 1-3
    - partitioning 1-28
    - recovery machine check interruptions, controlling recording of 4-107
    - recovery processing 2-56
    - requests
      - displaying 4-78
      - responding to 4-129
    - reset 1-14
    - residence volume 1-3, 1-26
    - response, lack of 2-49
    - restart 1-8
      - reasons on system control (SYSCTL) frame 1-8

- system control (SYSCTL) frame 1-8
- trace 1-21
- system command groups 3-9, 3-10
  - assigning 4-203
- system console
  - definition of 1-1
- system failure messages
  - routing of 4-53
- system inclusion RNL
  - displaying contents of 1-17, 4-87
- system log
  - controlling 2-45
  - message class 3-6
  - purpose of 1-20, 2-48
- system management facilities (see SMF)
- system messages
  - controlling 3-19
- system parameters 1-7
- system-assigned hardcopy log 2-47
- system-initiated DDR 4-193
- systems exclusion RNL
  - displaying contents of 1-17, 4-87
- SYSUDUMP dump type 4-13
- SYS1.BROADCAST 1-16, 4-135, 4-136, 4-139, 4-140
  - contents of 1-10
- SYS1.COMMNDxx member of
  - SYS1.PARMLIB 4-209
- SYS1.DUMPxx 1-17, 4-97
- SYS1.IMAGELIB 1-16
- SYS1.LINKLIB 1-16
- SYS1.LOGREC 1-17, 4-147, 4-155
- SYS1.LPALIB 1-16
- SYS1.MANx 1-17
- SYS1.NUCLEUS 1-4, 1-17
- SYS1.PARMLIB 3-23
  - description of some members 1-4
  - using to define consoles 3-2
- SYS1.PROCLIB 1-16
- SYS1.STGINDEX 1-17
- SYS1.SVCLIB 1-17
- SYS1.UADS 1-16

## T

T (see SET command)

tape

- blank 1-23
- label 1-23
- label verification 4-120, 4-121
- unlabeled 1-23

tapemark 1-23

TCAM (telecommunications access method) 1-9, 4-111, 4-112

TCAS (terminal control address space) 4-113

telecommunications access method (TCAM) 1-9, 4-111, 4-112

terminal control address space (TCAS) 4-113

terminating

- job 1-10, 4-10
- LLA address space 4-189

- MOUNT command 4-10
- time-sharing user 4-10

time and date

- displaying 4-84
- setting 4-131

time interval for dynamic displays
 

- changing 3-51

time stamp, adding to message 3-52

time-sharing option (TSO) 1-9

TOD (time-of-day) clock

- setting 1-4, 4-131

TR (see TRACK command)

trace

- master 1-21
- system 1-21, 4-147

TRACE command 1-21

- complete syntax 4-195

- examples of 4-196

- summary of 4-7

trace table 1-21

TRACK command

- complete syntax 4-197

- example of 2-44, 4-198

- routed by MSGRT command 3-46, 4-122

- setting up TRACK display 3-60

- summary of 4-7

- use of 1-6

TSO (time sharing option) 1-9

- displaying information about 3-52

TSO/TCAM

- starting 4-112

- stopping 4-112

TSO/VTAM, modifying 4-113

typing commands 2-38

## U

U (see UNLOAD command)

uniprocessor

- definition of 1-26

unlabeled tape 1-23

UNLOAD command 1-23

- complete syntax 4-199

- summary of 4-7

unloading

- tape or DASD volumes 4-199

updating

- dynamic status display 4-40

uppercase, rules about using 4-8, 4-9

USE attribute 1-23

user exit routines

- controlled by MPFLSTxx 3-5

- description of use in processing messages 3-23

using

- console cluster 3-54

- system restart function 1-8

## V

V (see VARY command)

value compare in SLIP, definition 4-164

VARY command 1-15, 1-16, 1-17, 1-18, 1-25

- complete syntax 4-201-4-215
- CONSOLE
  - examples of 3-13, 3-26
  - examples of 1-35, 3-57, 3-58, 4-204-4-212, 4-214
- HARDCPY
  - examples of 3-7, 3-8, 3-9, 3-61
- MSTCONS
  - examples of 3-11, 4-205
  - relationship to HARDCOPY statement in CONSOLxx 3-4
  - summary of 4-7

VARY CONSOLE command, rejected during no-consoles condition 2-55

varying
 

- alternate method of recovering lost messages 2-50
- console 3-11, 4-202
- devices defined as 3270 model X 3-11
- global resource serialization system 4-214
- hardcopy log 4-205
- I/O device 4-209
- I/O path 4-212
- master console 4-205
- resources 4-208
- secondary console 4-208

VATLSTxx member of SYS1.PARMLIB 1-15, 1-23

Vector Facilities
 

- bringing online 4-26
- taking offline 4-26

verifying deletion request 3-34

virtual telecommunications access method (VTAM) 1-9, 4-111, 4-113

VIRTUAL=REAL (V=R) 1-12

volume
 

- characteristics 1-15
- mounting 1-15, 1-23, 4-120

VTAM (virtual telecommunications access method) 1-9, 4-111, 4-113

W

W (see WRITELOG command)

wait state 2-51, 2-53, 2-54
 

- enabled 2-54
- invalid 1-8
- restartable 1-8, 1-24

warm start 1-3

warning line
 

- description of 2-7
- illustration of 3-43

workload level 4-145

write-to-operator (WTO) buffers 2-51, 2-54
 

- changing the number of allowed 4-43
- shortage of 4-47

write-to-operator-with reply (WTOR) buffers 2-51
 

- changing the number of allowed 4-43

WRITELOG command 1-20
 

- complete syntax 4-216
- example of 4-216
- summary of 4-7
- to force printing of SYSLOG 2-48

WTL macro 1-20

WTO (write-to-operator) buffers
 

- changing the number of allowed 4-43
- shortage of 2-51, 2-54, 4-47

WTO messages
 

- buffer limits 2-51
- controlled by user exit IEAVMXIT 3-23
- routing of 4-53

WTOBFRS initialization parameter 4-42

WTOR (write-to-operator-with-reply) buffers
 

- changing the number of allowed 4-43
- shortage of 2-51

WTOR messages
 

- buffer limits 2-51

X

XWTR procedure 1-13

Z

Z (see HALT command)

1

1403 printer
 

- console type 2-3
- used as hardcopy log 2-2

2

2501-B1,B2 reader
 

- console type 2-3

2540-1 reader
 

- console type 2-3

2740-1 communications terminal 2-55
 

- console type 2-3
- used as console 2-2

3

3084 processor complex
 

- partitioning 1-28

3090 model 400 processor complex
 

- partitioning 1-28

3180-1 display station
 

- as operator console 2-29
- initial message deletion specifications 3-35
- message area sizes 3-43
- message presentation 2-33
- special screen characters 2-31

- summary of features 2-6
- 3203-5 console printer
  - console type 2-3
- 3211-1 printer
  - console type 2-3
- 3277-2 display station
  - console type 2-3
  - description of 2-9
  - illustration of 2-10
  - in console cluster 3-55
  - in message stream mode 3-15
    - illustration of 3-15
  - initial message deletion specifications 3-35
  - intensity of messages 2-14
  - message area sizes 3-43
  - pressing back-tab key 2-50
  - program function keyboard on 2-13
  - special screen characters 2-12
  - summary of features 2-6
  - using selector pen 2-42
- 3278-2,2A,3,4, and 5 display station
  - as operator console 2-2, 2-14
  - console type 2-3
  - initial message deletion specifications 3-35
  - message area sizes 3-43
  - message presentation 2-20
  - special screen characters 2-17
  - summary of features 2-6
- 3279-2A,2B,2C,3A, and 3B display station
  - as operator console 2-14
  - audible alarm on 2-20
  - console type 2-3
  - ERASE INPUT key on 2-19
  - initial message deletion specifications 3-35
  - message area sizes 3-43
  - message presentation 2-22
  - special screen characters 2-17
  - summary of features 2-6
- 3279-2B and 3B display station
  - message presentation 3-20
- 3279-3X,S3G,2X, and S2B display station
  - initial message deletion specifications 3-35
  - message area sizes 3-43
  - message presentation 2-22
  - summary of features 2-6
- 3284-1 and 2 printer
  - console type 2-3
  - used as hardcopy log 2-2
- 3286-1 and 2 printer
  - console type 2-3
- 3290 information panel
  - as console cluster 3-64
  - console type 2-3
  - description of 2-23
  - highlighting of messages 2-28
  - illustration of 2-23
  - illustration of four-part console 2-25
  - initial message deletion specifications 3-35
  - message area sizes 3-43
  - message presentation 2-28
  - summary of features 2-6
  - using two-part or four-part screen 2-25
- 3348-70F data module 4-192
- 3480 tape device 4-200, 4-209, 4-210
- 3505-B1 and B2 card reader
  - console type 2-3
  - used as composite console 2-2
- 3525-5 card punch
  - console type 2-3
  - used as composite console 2-2





**This Newsletter No.** GN28-1199  
**Date** November 16, 1987  
**Base Publication No.** GC28-1206-5  
**File No.** S370-40

**Prerequisite Newsletters/  
Supplements** None

## **MVS/Extended Architecture Operations: System Commands**

©Copyright IBM Corp. 1982, 1987

This Technical Newsletter contains replacement pages for *MVS/XA Operations: System Commands* in support of APAR # OY09591, and APAR # OY10056.

Before inserting any of the attached pages into *MVS/XA Operations: System Commands*, read *carefully* the instructions on this cover. They indicate when and how you should insert pages.

| <u>Pages to<br/>be Removed</u> | <u>Attached Pages<br/>to be Inserted*</u> |
|--------------------------------|---|
| Cover - Edition Notice         | Cover - Edition Notice                    |
| xv - xviii                     | xv - xviii                                |
| 2-37 - 2-38                    | 2-37 - 2-38.2                             |
| 3-3 - 3-4                      | 3-3 - 3-4                                 |
| 3-21 - 3-22                    | 3-21 - 3-22                               |
| 4-111 - 4-112                  | 4-111 - 4-112                             |
| 4-115 - 4-118                  | 4-115 - 4-118                             |

\*If you are inserting pages from different Newsletters/Supplements and *identical* page numbers are involved, always use the page with the latest date (shown in the slug at the top of the page). The page with the latest date contains the most complete information.

A change to the text or to an illustration is indicated by a vertical line to the left of the change.

### **Summary of Amendments**

This Technical Newsletter contains information on the END(ID) option for the MODIFY CATALOG command.

This Technical Newsletter documents the removal of JES3 action message retention facility (JAMRF). This is in support of APAR # OY09591.

This Technical Newsletter also contains the MVS Command Delimiter which allows the operator to issue multiple commands. This is in support of APAR # OY10056.

*Note: Please file this cover letter at the back of the publication to provide a record of changes.*

**IBM Corporation, Information Development, Dept. D58, Building 921-2,  
P.O. Box 390, Poughkeepsie, New York 12602**



.

.



.

.

.







GC28-1206-5

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

**Note:** *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comment are:

Clarity    Accuracy    Completeness    Organization    Coding    Retrieval    Legibility

If you wish a reply, give your name, company, mailing address, and date:

---

---

---

---

What is your occupation? \_\_\_\_\_

How do you use this publication? \_\_\_\_\_

Number of latest Newsletter associated with this publication: \_\_\_\_\_

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

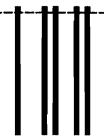
Reader's Comment Form

Cut or Fold Along Line

Fold and tape

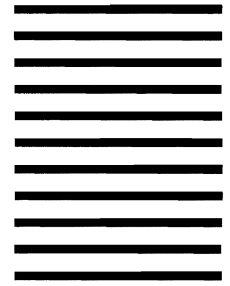
Please Do Not Staple

Fold and tape



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Department D58, Building 921-2  
PO Box 390  
Poughkeepsie, New York 12602



Fold and tape

Please Do Not Staple

Fold and tape

Printed in U.S.A.

