

OS/390



MVS System Commands Summary

OS/390



MVS System Commands Summary

Note

Before using this information and the product it supports, be sure to read the general information under "Appendix. Notices" on page 55.

Eleventh Edition, December 2000

This is a major revision of GX22-0040-09.

This edition applies to Version 2 Release 10 of OS/390 (5647-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

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

FAX (Other Countries):

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

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

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

World Wide Web: <http://www.ibm.com/s390/os390/webqs.html>

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

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

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

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

© Copyright International Business Machines Corporation 1988, 2000. All rights reserved.

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

Contents

About This Book v

Who Should Use This Book v

Summary of Changes vii

Syntax Notation xiii

MVS System Commands Syntax 1

ACTIVATE Command 1

CANCEL Command (C) 1

CHNGDUMP Command (CD) 1

CONFIG Command (CF) 3

CONTROL Command (K) 4

DEVSERV Command (DS) 7

DISPLAY Command (D) 7

DUMP Command 21

DUMPDS Command (DD) 21

FORCE Command 22

HALT Command (Z) 23

IOACTION Command (IO) 23

LIBRARY Command (LI) 23

LOG Command (L) 23

LOGOFF Command 24

LOGON Command 24

MODE Command 24

MODIFY Command (F) 25

MONITOR Command (MN) 29

MOUNT Command (M) 29

MSGRT Command (MR) 29

PAGEADD Command (PA) 30

PAGEDEL Command (PD) 30

QUIESCE Command 30

REPLY Command (R) 30

RESET Command (E) 32

ROUTE Command (RO) 33

SEND Command (SE) 33

SET Command (T) 34

SETDMN Command (SD) 35

SETETR Command 35

SETGRS Command 35

SETIOS Command 35

SETLOAD Command 36

SETLOGRC Command 36

SETOMVS Command 36

SETPROG Command 37

SETRRS CANCEL Command 39

SETSMF Command (SS) 39

SETSMS Command 39

SETSSI Command 39

SETXCF Command 39

SLIP Command (SL) 42

START Command (S) 46

STOP Command (P) 47

STOPMN Command (PM) 48

STOPTR Command (PT) 48

SWAP Command (G) 49

SWITCH Command (I) 49

TRACE Command 49

TRACK Command (TR) 50

UNLOAD Command (U) 50

VARY Command (V) 51

WRITELOG Command (W) 54

Appendix. Notices 55

Trademarks 56

About This Book

This book states the purpose and shows the syntax of the MVS system commands (also known as "operator commands" or "console commands") you can use to control a processor under the OS/390 operating system. For the full explanations and detailed descriptions of the MVS system command functions, syntax, and parameters, see *OS/390 MVS System Commands*.

This book includes only the MVS system commands, which it lists in alphabetical order. For information about JES subsystem commands and their functions, see *OS/390 JES2 Commands* or *OS/390 JES3 Commands*.

For information about the system messages referred to in this publication, see *OS/390 MVS System Messages*.

Who Should Use This Book

OS/390 MVS System Commands Summary is intended for anyone using a console and operator commands to manage the OS/390 operating system. It assumes that the user understands the hardware controls and features of the installation and the general organization and functions of the OS/390 operating system.

Summary of Changes

| **Summary of Changes**
| **for GX22-0040-10**
| **as updated December, 2000**

| This publication contains information previously presented in GX22-0040-09, which
| supports OS/390 Version 2 Release 10.

| ***Changed Information***

- | • The CONFIG and DISPLAY M commands support real storage above 2
| gigabytes.
- | • DISPLAY IPLINFO now shows architecture level.
- | • SLIP ACTION=WAIT supports 64-bit GPRs.
- | • The DISPLAY OMVS command has two additional new operands: LIMITS and
| RESET. LIMITS displays information about OS/390 UNIX System Services
| parmlib limits and current system use. RESET resets the high-water mark for
| system limits to 0.
- | • The SETOMVS command has two new operands: PID= and LIMMSG. PID=
| dynamically changes a limit for a process, and LIMMSG specifies how to display
| console messages that indicate when parmlib limits are reaching critical levels.

| This and each prior edition of this book also reflect the addition, modification, and
| deletion of information to support miscellaneous terminology, maintenance, and
| editorial changes. Vertical lines in the left margin of this edition indicate technical
| changes or additions to the text and illustrations *in this edition*.

| **Summary of Changes**
| **for GX22-0040-09**
| **OS/390 Version 2 Release 10**

| This publication contains information previously presented in GX22-0040-08, which
| supports OS/390 Version 2 Release 9. The following summarizes the changes to
| that information.

| ***New Information***

- | • The DISPLAY Library Lookaside (D LLA) command is added.
- | • A new command (VARY SWITCH), and new keywords on existing commands
| (DISPLAY M, SWITCH; DISPLAY IOS, DCM; DISPLAY IOS, GROUP; and SETIOS
| DCM=ON | OFF | REFRESH) are added. These commands are included in the
| OS/390 Release 10 product to allow customers to become familiar with their
| function and syntax. They will not be fully functional until they can be issued in
| an LPAR clustering environment.

| ***Changed Information***

- | • The DISPLAY OMVS command has a new operand, BRL, which displays
| thread-level information for any thread that is in a byte-range lock wait.

| ***Deleted Information:***

- | • Duplexing of PLPA and Common is no longer supported.
- | • SWAP data sets are no longer supported.

**Summary of Changes
for GX22-0040-08
OS/390 Version 2 Release 9**

This publication contains information previously presented in GX22-0040-07, which supports OS/390 Version 2 Release 8. The following summarizes the changes to that information.

New Information

- The command DISPLAY SMS,SMSVSAM,QUIESCE is added.
- A parameter is added to the SET OMVS command to check the syntax of a parmlib member.
- Parameters are added to the SET OMVS command for use in a sysplex where shared HFS is used.
- Parameters are added to the DISPLAY OMVS command to display network routing and PFS information.
- New parameters are displayed with D OMVS,FILE for shared HFS.
- The ANALYSIS and FREELATCH variants are added to the VARY SMS,PDSE command.

Changed Information

- The list of MODIFY CATALOG command subparameters has been brought into consonance with that in *OS/390 DFSMS: Managing Catalogs*.

Deleted Information

- Text of parameter descriptions no longer appears in this book. As a *summary* this book now shows only the purpose and syntax of each command. The reader should refer to *OS/390 MVS System Commands* for the complete text.

**Summary of Changes
for GX22-0040-07
OS/390 Version 2 Release 8**

This publication contains information previously presented in GX22-0040-06, which supports OS/390 Version 2 Release 7. The following summarizes the changes to that information.

Changed Information

- Two parameters are added to the SETOMVS command: RESET=(xx), and SYSCALL_COUNTS=(YES|NO).
- Notices and the list of trademarks used in this book now appear as Appendix A following all command descriptions.

**Summary of Changes
for GX22-0040-06
OS/390 Version 2 Release 7**

This publication contains information previously presented in GX22-0040-05, which supports OS/390 Version 2 Release 6. The following summarizes the changes to that information.

New Information

DISPLAY EMCS command

This command displays the information about extended MCS (EMCS) consoles.

Changed Information

- Three new parameters are added to the MODIFY BPXOINIT command: SHUTDOWN=FORKS, RESTART=FORKS, and DUMP=pid.
- New subparameter ECshr (to support enhanced catalog sharing, ECS), is added to the MODIFY CATALOG command.
- New subparameter REPORT,PERFORMANCE (to display the performance of key events occurring in the catalog address space) is added to the MODIFY CATALOG command.

Summary of Changes for GX22-0040-05 OS/390 Version 2 Release 6

This publication contains information previously presented in GX22-0040-04, which supports OS/390 Version 2 Release 5. The following summarizes the changes to that information.

New Information

DISPLAY LOGGER command

This command displays the current status of the system logger, individual log streams, or log streams from a sysplex view.

Changed Information

- The SYS= parameter is added to the DISPLAY CONSOLES command.
- The MCONLY parameter is added to the DISPLAY CONSOLES command.
- The TITLE and PARMLIB parameters are added to the DUMP command.
- The RESMIL and TOLINT parameters are added to the SETGRS command.
- The MSGONLY parameter is added to the SETIOS command.
- As part of the name change of OS/390 OpenEdition to OS/390 UNIX System Services, occurrences of OpenEdition are changed to OS/390 UNIX System Services or its abbreviated name, OS/390 UNIX. OpenEdition may continue to appear in messages, panel text, and other code with OS/390 UNIX.

This edition of the book also reflects the addition, modification, and deletion of information to support miscellaneous terminology, maintenance, and editorial changes. These include the addition of a Table of Contents and page headings with the command names.

Summary of Changes for GX22-0040-04 OS/390 Version 2 Release 5

This publication contains information previously presented in GX22-0040-03, which supports OS/390 Version 2 Release 4. The revisions include only terminology, maintenance, and editorial changes.

Summary of Changes for GX22-0040-03 OS/390 Version 2 Release 4

The publication contains information previously presented in GX22-0040-02, which supports OS/390 Version 1 Release 3. The following summarizes the changes to that information.

New Information

DISPLAY RTLS

This command displays the current status of the run-time library services (RTLS) environment.

DISPLAY WLM,RESOURCE=resource

This command displays the status of resources.

DISPLAY WLM,SCHENV=schenv

This command displays the status of scheduling environments.

MODIFY WLM,RESOURCE=resource

This command changes resource state settings (to ON, OFF, or RESET).

SET RTLS=xx

This command allows you to change the RTLS configuration.

VARY WLM,POLICY=policy,REFRESH

This command discards historical workload characterization data, then resets and activates the named policy and starts data collection anew.

Changed Information

DISPLAY PROG,LPA

This command displays information about modules dynamically added to the LPA, and displays the minimum amount of CSA and ECSA that must remain after adding a module to the LPA.

SET PROG=xx

This command activates a PROGxx member containing definitions of modules that are to be added to or deleted from the LPA after IPL.

SETPROG LPA

This command enables you to identify modules to add to or delete from the LPA any time after IPL. It also allows you to specify threshold values for minimum amounts of CSA storage that still must be available after an ADD operation.

Summary of Changes for GX22-0040-02 OS/390 Release 3

This publication contains information previously presented in GX22-0040-01, which supports OS/390 Version 1 Release 2. The following summarizes the changes to that information.

New Information

SETGRS command

This command provides the ability to migrate from a global resource serialization (GRS) ring complex to a star complex, without a complex-wide IPL.

DISPLAY WLM,APPLENV=applenv

This command displays the status of application environments.

VARY WLM,APPLENV=*applenoname*

This command controls application environments (with REFRESH, QUIESCE, and RESUME options).

Changed Information

CANCEL

Special CANCEL considerations may apply if the system is part of a global resource serialization complex.

Summary of Changes for GX22-0040-01 OS/390 Release 2

This publication contains information previously presented in GX22-0040-00, which supports OS/390 Version 1 Release 1. The following summarizes the changes to that information.

New Information

SETLOAD command

This command provides the ability to change the parmlib concatenation (logical parmlib) dynamically.

SETGRS command

This command provides the ability to migrate from a global resource serialization ring complex to a star complex without a complex-wide IPL.

DISPLAY PARMLIB

This command displays information about the logical parmlib setup for a system.

DISPLAY IPLINFO

This command displays information about the general IPL information that the system uses.

DISPLAY PROG,EXIT,EXITNAME=*exitname*,DIAG

This command displays diagnostic information about the specified exit, including entry point and load addresses.

DISPLAY PROG,LNKLST

This command displays information about the LNKLST concatenation defined through PROGxx or LNKLSTxx.

Changed Information

SETXCF COUPLE

GRS is added to the list of services supported by the TYPE=(name,name...) parameter.

DISPLAY GRS

The Display command now includes the global resource serialization star configuration. Message ISG020I is replaced by message ISG343I.

CANCEL

GRS=STAR is added to the list of parameters that invoke global resource serialization.

Summary of Changes for GX22-0040-00 OS/390 Release 1

This publication contains information previously presented in GX22-0022, which supports MVS/ESA System Product Version 5. The revisions include only terminology, maintenance, and editorial changes.

Syntax Notation

You must follow certain syntactical rules when you code commands. As you code the MVS commands described in this book, use Table 1 for help with the syntax.

Table 1. System Command Syntax Notation

Notation	Meaning	Example Book Syntax
Apostrophes	Must be entered as shown.	SEND 'message',NOW
Comma	Must be entered as shown.	DISPLAY C,K
Ellipsis ...	The parameter can be repeated. Do not enter the ellipsis.	VARY (conspec[,conspec]...),ONLINE
Lower Case Parameter	A parameter must be substituted. You can enter the command and the parameter in either upper or lower case.	MOUNT devnum
Or-bar ()	You must enter one of the items. You cannot enter more than one.	ACTIVATE[,RECOVER=SOURCE TARGET]
Parentheses and special characters	Must be entered as shown.	DUMP COMM=(text)
Single parameter in brackets	The parameter is optional.	DISPLAY DMN[=domainum]
Stacked items in braces	You must enter one of the items. You cannot enter more than one.	MN {DSNAME} {SPACE} {STATUS}
Stacked items with or-bars () and brackets	Optional, mutually exclusive parameters. Enter one or none.	CD RESET [,SDUMP ,SYSABEND ,SYSUDUMP ,SYSMDUMP ,ALL]
Underline	If you do not enter one of the parameters, the system supplies the underlined parameter, which is the default	K T [,REF] [, <u>UTME</u> =nnn]
Upper Case Parameter	The parameter must be spelled as shown. You can enter the command and the parameter in either upper or lower case.	DISPLAY SMF

MVS System Commands Syntax

ACTIVATE Command

Purpose: Use the ACTIVATE command to activate or test a new I/O configuration definition dynamically.

```
ACTIVATE { [,IODF=xx] [,EDT=xx] [,PROC=procname] [,CFID=id] }
         {      [,RECOVER=SOURCE|TARGET] [,ACTIOCD=xx]      }
         { [,SOFT[=VALIDATE|=NOVALIDATE] ]                  }
         { |,TEST                                           }
         { |,FORCE                                          }
         { |,FORCE={DEVICE                                }
         { |          {CANDIDATE                          }
         { |          {(DEVICE,CANDIDATE)}                }
         { |          {(CANDIDATE,DEVICE)}                }
```

Note: Do not specify a comma before the first parameter following ACTIVATE.

CANCEL Command (C)

Purpose: Use the CANCEL command to end an active job, started task, or time-sharing user immediately.

```
C {jobname          } [,DUMP] [,A=asid] [,ARMRESTART]
  {U=userid         }
  {[jobname.]identifier}
```

CHNGDUMP Command (CD)

Purpose: Use the CHNGDUMP command to change the mode and system dump options list for any dump type, or to request structures to be dumped when one or more systems connected to a coupling facility fail.

Example: The following CHNGDUMP command removes options from the options lists.

CHNGDUMP Command

```

CD DEL[,SDUMP[=(option[,option]...)]
      [,Q={YES|NO}]
      [,TYPE={XMEM|XMEME}]
      [,ALL]
      [,SYSFAIL,STRLIST={ALL|(STRNAME=strname[,STRNAME=strname]...)}]]
      [,SYSAEND][,SDATA=(option[,option]...)]
      [{,SYSUDUMP} |,PDATA=(option[,option]...)]
      [,ALL]
      [,SYSDUMP[=(option[,option]...)]
      [,ALL]
      [,ALL]
  
```

Example: The following CHNGDUMP command resets the options lists.

```

CD RESET[,SDUMP
      [,SYSAEND]
      [,SYSUDUMP]
      [,SYSDUMP]
      [,ALL]
  
```

Example: The following CHNGDUMP command sets the dump modes and options.

```

CD SET,{NODUMP
      {
      {OVER
      {ADD
      {SDUMP[=(option[,option]...)]
      { [,Q={YES|NO}]
      { [,TYPE={XMEM|XMEME}]
      { [,BUFFERS={nnnnK|nnnM}]
      { [,MAXSPACE=xxxxxxxM]
      { [,MSGTIME=yyyyy]
      { [,SYSFAIL,STRLIST=(s-option[,s-option]...)]
      { [,NODUMP]
      { |,OVER
      { |,ADD
      {
      {SYSAEND][,SDATA=(option[,option]...)] [,NODUMP]
      {SYSDUMP} |,PDATA=(option[,option]...)] |,OVER
      { |,ADD
      {
      {SYSDUMP[=(option[,option]...)] [,NODUMP]
      { |,OVER
      { |,ADD
  
```

Where **s-option** represents:

```

STRNAME=strname
[,CONNAME=conname ]
[ ]
[,ACCESSTIME={ENFORCE|NOLIMIT|NOLIM} ]
[ ]
[,LOCKENTRIES ]
[ ]
[,USERCNTLS ]
[ ]
[,EVENTQS ]
[ ]
[, (EMCONTROLS={ALL|(list)}) ]
[ ]
[, ({COCLASS|STGCLASS|LISTNUM}={ALL|(list)}
  {[,ADJUNCT={CAPTURE|DIRECTIO}] [,ENTRYDATA={UNSERIALIZE|SERIALIZE}}] ]
[ {[,SUMMARY] } ]

```

CONFIG Command (CF)

Purpose: Use the CONFIG command to change or check the configuration of the system.

Example: The following CONFIG command reconfigures the system (available processors, storage sections, vector facilities, and channel paths) directly.

```

CF {{CPUAD|CPU}(x[,x]...)[, {ONLINE|ON} [,VFON ] |, {OFFLINE|OFF}] }
{
  |,VFOFF
}
{VF(x[,x]...)[, {ONLINE|ON} |, {OFFLINE|OFF}] }
{
  {STORAGE|STOR}{{ddddM} } [, {ONLINE|ON} |, {OFFLINE|OFF}] }
  {
    {{ddddM-ddddM} }
  }
  {
    {(E=id) }
  }
  {ESTOR(E=id)[, {ONLINE|ON} |, {OFFLINE|OFF}] }
  {
    {CHP{(xx) } [, {ONLINE|ON} [NOVARY] |, {OFFLINE|OFF} [,UNCOND]] }
    {
      {(aa-bb) } |,FORCE
    }
    {
      {(list) }
    }
    {
      {(ALL,id) }
    }
  }

```

Example: The following CONFIG command reconfigures the system with a CONFIGxx member of SYS1.PARMLIB.

```
CF MEMBER[(member-id)]
```

Example: The following CONFIG command displays the system configuration so that you can decide which processors, central or expanded storage elements, vector facilities, and channel paths you want to bring online/offline. It then brings online/offline items specified in response to message IEE522D.

CONFIG Command

```
CF {ONLINE|ON } [,L={ a    }]
   {OFFLINE|OFF}      {cc   }
                       {cca  }
                       {name }
                       {name-a}
```

CONTROL Command (K)

Purpose: Use the CONTROL command to control the screen display of MCS consoles.

Example: The following CONTROL command changes the display area specifications.

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

Example: The following CONTROL command deletes one or more action messages that the action message retention facility has retained.

```
K C,{A|I|E|CE},{id|id-id[,id|id-id]...}
```

Example: The following CONTROL command stops an in-line status display.

```
K C,D,id[,L={a    }]
   {cc   }
   {cca  }
   {name }
   {name-a}
```

Example: The following CONTROL command displays information on the screen.

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

Example: The following CONTROL command removes a single deletable message from the screen.

CONTROL Command

```
K E[,nn          ]
    |,nn,nn
    |,SEG
    |,F
    |,N
    |,PFK
    |,D[,L={a    }]
        {cc   }
        {cca  }
        {name }
        {name-a}
```

Example: The following CONTROL commands activate (Y) or deactivate (N) the action message retention facility.

```
K M[,AMRF={Y|N}]
    |,REF
```

```
K M[,REF|[,MLIM=nnnn][,RLIM=mmm]]
```

```
K M[,UEXIT={Y|N}]
    |,REF
```

Example: The following CONTROL command displays the maximum number of write-to-log requests that the system can hold in buffers.

```
K M[,LOGLIM={nnnnn|0}]
    |,REF
```

Example: The following CONTROL command displays or changes the maximum time that ROUTE commands wait for a response before aggregating responses.

```
K M[,ROUTIME=nnn ]
    |,REF
```

Example: The following CONTROL command dynamically increases the maximum number of reply IDs.

```
K M[,RMAX=nnnn ]
    |,REF
```

CONTROL Command

Example: The following CONTROL command defines the program function keys.

```
K N,PFK={{(nn1{,CMD='text[;text]...'})[,CON={Y|N}]} }
        { ,KEY=nn2[,nn2]... }
        {
        {nnnnnnn[,L={cc|name}]}
        }
```

Example: The following CONTROL command changes or displays a console's message queue.

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

Example: The following CONTROL command changes or displays message deletion and format specifications.

```
K S[,REF ]
    [,CON={Y|N}][,SEG=nn][,DEL={Y|N|R|RD|W}]
    [,RNUM=nn][,RTME=nnn][,MFORM=(option[,option]...)]
[,L={cc|name}]
```

Example: The following CONTROL command changes or displays time intervals for dynamic displays.

```
K T[,REF ][,L={cc } ]
    [,UTME=nnn {name}]
```

Example: The following CONTROL commands change the operating mode or message levels of a console.

```
K V[,REF ][,L={cc } ]
    [,USE={FC|SD|MS}][,CMDSYS={sysname|*}]
    {name}
```

```
K V[,REF ][,L={cc } ]
    [,LEVEL=(type[,type]...)]
    {name}
```

DEVSERV Command (DS)

Purpose: Use the DEVSERV command to request a display of the status of DASD and tape devices.

```

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

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

{QDASD | QD}{,? |

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

{QTAPE | QT}{,? |

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

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

```

DISPLAY Command (D)

Purpose: Use the DISPLAY command to display information about the operating system, the jobs and application programs that are running, the processor, devices that are online and offline, central and expanded storage, workload management service policy and mode status, and the time of day.

Example: The following DISPLAY command shows APPC configuration information.

DISPLAY Command

```

D APPC,{TP[,SUMMARY|SUM|S] [, {ASID|A}=asid] }
  { |,LIST|,L [ASNAME=asname] }
  { |,ALL|,A [DIR=IN|OUT] }
  { [IT=sssss[.ttt]] }
  { [LLUN=lluname] }
  { [LTPN=ltpname] }
  { [PNET=pnetid] }
  { [PLUN=pluname] }
  { [PTPN=ptpname] }
  { [SCHED={schedname}] }
  { {*NONE*} }
  { [STPN=stpname] }
  { [USERID=userid] }
  { }
  {UR[,SUMMARY|SUM|S] [,URID=urid] }
  { |,LIST|,L [LUWID=luwid] }
  { |,ALL|,A [PNET=pnetid] }
  { [PLUN=pluname] }
  { [LLUN=lluname] }
  { }
  {SERVER[,SUMMARY|SUM|S] [, {ASID|A}=asid] }
  { |,LIST|,L [ASNAME=asname] }
  { |,ALL|,A [LLUN=lluname] }
  { [STPN=stpname] }
  { }
  {LU[,SUMMARY|SUM|S] [,LLUN=lluname] }
  { |,LIST|L [PNET=pnetid] }
  { [PLUN=pluname] }
  { |,ALL|A [SCHED={schedname}] }
  { {*NONE*} }
  { }

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

```

Example: The following DISPLAY command shows ASCH configuration information.

```

D ASCH{[,SUMMARY|SUM|S] [, {ASID|A}=asid] }
  { |,LIST|,L [,{CLASS|C}=classname] }
  { |,ALL|,A [LTPN=ltpname] }
  { [QT=sssss[.ttt]] }
  { [TPST=schedtype] }
  { [USERID=userid] }
  { }

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

```

Example: The following DISPLAY command shows page data set information.

```

D ASM[,PLPA ]
  |,COMMON
  |,LOCAL
  |,ALL
  |,PAGE=[dsname|ALL]
  |,PAGEDEL

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

```


DISPLAY Command

Example: The following DISPLAY command shows a summary of CONTROL command operands.

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

Example: The following DISPLAY command shows information about the coupling facilities that are attached to the system.

```
D CF[,CFNAME={{(cfname[,cfname]...)}}
```

Example: The following DISPLAY command shows the console group definitions in effect for the sysplex.

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

Example: The following DISPLAY command shows console status information.

```
D {CONSOLES}, {{ACTIVE|A } [,CA={name  
{C } } {{SS } {(name[,name]...)} {(rr-ss) } }  
{{NACTIVE|N} [MSTR] {(rr[,ss]...)} }  
{ [,SYS=system name] }  
{ }  
{KEY [=key] }  
{ {BACKLOG|B} }  
{ {MASTER|M} [,SYS=system name] }  
{ }  
{ {MCONLY} }  
{ * }  
{ {LIST|L} }  
{ }  
{ {HARDCOPY|HC} [,SYS=system name] }  
{ }  
{ CN={xx } [,ROUT={NONE|ALL|rr } ] }  
{ { (xx[,yy]...) } {(rr-ss) } }  
{ { (xx-yy) } {(rr[,ss]...)} }  
{ [,SYS=system name] }  
{ }  
{ HCONLY }  
{ }  
{ U={ ([/]devnum1 [, [/]devnum2]...)} }  
{ { ([/]lowdevnum-[/]highdevnum)} }  
{ { [/]devnum } }  
{ MSTR [,SYS=system name] }  
[,L={a|cc|cca|name|name-a}]
```

DISPLAY Command

Example: The following DISPLAY command shows a list of major names or resource information for the specified resource(s).

```
D DLF[,RES={{qname|*}[,rname|,*]}}[,HEX]
[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows the domain description table or a specific table entry (*domainnum* from 2 to 129).

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

Note: The above command is not valid on systems operating in workload management goal mode.

Example: The following DISPLAY command shows the dump options or dump data set status.

```
D {DUMP},{ {STATUS|ST|S} }
{D } {
  {OPTIONS|O}
  {
    {TITLE|T } {,AUTODSN={aaa|ALL}
    {ERRDATA|ER|E} {
      {,DSN={ALL|(ALL)
      {nn|(nn[,nn]...)
      {nn-nn|(nn-nn[,nn-nn]...)
      {(nn[,nn]...,nn-nn[,nn-nn]...)}
      {
        {,DUMPID={xxx|(yyy[,zzz]...)
        {aaa-bbb|(aaa-bbb[,ccc-ddd]...)
        {(yyy[,zzz]...,aaa-bbb[,ccc-ddd]...)}
    }
  }
[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows information about extended MCS (EMCS) consoles.

```

D EMCS, {SUMMARY|S }
      {INFO|I }
      {FULL|F }
      {STATUS=A|N|L|B[{nn}]|ERR }
      {ST }
      {CN=consname|* }
      {SYS=sysname|* }
      {KEY=keyname|* }
      {AUTH={ANY }
        {MASTER }
        {SYS }
        {IO }
        {CONS }
        {ALL }
        {INFO }
        {SYSONLY }
        {IOONLY }
        {CONSONLY }
        {ALLONLY }
        {INFOONLY }
      }
      {ATTR={ANY }
        {YES }
        {ROUT }
        {UD(YES|NO) }
        {HC }
        {AUTO(YES|NO) }
        {MN }
        {NONE }
      }
      {DOM={ANY }
        {NORMAL }
        {ALL }
        {NONE }
        {YES }
      }
      [,L={a|cc|cca|name|name-a}]
  
```

Example: The following DISPLAY command shows current external timer reference (ETR) synchronization.

```

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

Example: The following DISPLAY command shows global resource serialization (GRS) information.

DISPLAY Command

```

D GRS{ [,SYSTEM|,LINK|,ALL|,A|,DELAY|,D|,SUSPEND|S| [,CONTENTION|,C]
      [,RES=(qname|*[,rname|,*)] [,HEX] [,DEV=[/]devnum] [, {SUSPEND|S}]
      [,RNL={CONVERSION|CON|C}]
      {ALL|A}
      {EXCLUSION|EXCL|E}
      {INCLUSION|INCL|I}

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

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

      {,ANALYZE|,ANALYSE|,AN
        {,BLOCKER|BLOCK
        {,WAITER|WAIT
        { [,SYSTEM|,SYS=sysname|*] [,ASID|,AS=asid]
          [,JOBNAME|,JOB=jobname]

        {DEPENDENCY|DEPEND|DEP
          { [,SYSTEM=sysname|*] [,ASID=asid
            [,JOBNAME=jobname]

        { [,RES=(qname,rname), {SCOPE|SCO=SYSTEM[S],SYSTEM[S]|SYS[S]={sysname|*}]
          { [,COUNT|,CNT=nn] [,DETAIL|,DET]

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

Example: The following DISPLAY command shows IOS-related configuration information.

```

D IOS,CONFIG[(EDT)|(HSA)|(ALL)]
[,L={a|cc|cca|name|name-a}]
  
```

Example: The following DISPLAY command shows missing interruption handler (MIH) time intervals.

```

D IOS,MIH[,TIME={ALL|option}
      [
      [, {DEV }={([/]devnum[, [/]devnum1)...}
      [, {DEVX } {([/]devnum-[/]devnum1[, [/]devnum2-[/]devnum3)...}
      [, {TDEV }
      [, {TDEVX }

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

Example: The following DISPLAY command shows information about the devices the IOACTION STOP command affects.

```

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

Example: The following DISPLAY command shows information about the system's initial program loading (IPL).

DISPLAY Command

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

Example: The following DISPLAY command shows system activity information.

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

Example: The following DISPLAY command shows how to request library lookaside information.

```
D LLA
```

Example: The following DISPLAY command shows the status of the system logger, individual log streams, or log streams from a sysplex view.

```
D LOGGER[,Status ]
      [,Connection [,LSName=logstreamname [,Jobname=mvsjobname] [,SUMM ]] ] ]
                                     [,Detail]
      |,Jobname=mvsjobname [,LSName=logstreamname] [,SUMM ]]
                                     [,Detail]
      |,SYSPLEX [,LSName=logstreamname]
      |,DASDONLY
      [,Logstream [,LSName=logstreamname] [,STRName=structurename] ]
                                     |,DASDONLY
      [,STRucture [,STRName=structurename] ]
```

Example: The following DISPLAY command shows the logrec error and environmental record recording medium.

```
D LOGREC[, {CURRENT|CURR} | {DATASET|DSN} | {ALL|A}]
      [,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows system configuration information.

DISPLAY Command

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

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

Example: The following DISPLAY command specifies the status of the MVS message service and a list of languages available.

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

Example: The following DISPLAY command shows information about message processing and presentation.

```
D MPF[, {MSG|M}
      |, {COLOR|C}
      |, CMD

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

Example: The following DISPLAY command shows information about OS/390 UNIX.

```
D OMVS[{,SUMMARY|S}
      |, {ASID|A}=ALL
      |, {ASID|A}=asid
      |, U=userid
      |, {PID}=processid[,BRL]
      |, {FILE|F[,CAPS|C]}
      |, {VSERVER|V}
      |, {PFS|P}
      |, {CINET|CI}=A11|TPname
      |, {OPTIONS|O}
      |, {LIMITS|L[,PID=ProcessId][,RESET]}

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

Example: The following DISPLAY command shows sysplex-wide operator information.

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

DISPLAY Command

Example: The following DISPLAY command shows the parmlib data sets and volume serial numbers that are defined in LOADxx and the MASTER JCL (when there are no LOADxx parmlib statements).

```
D PARMLIB [,ERRORS|E]
[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows PFK definitions and tables.

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

Example: The following DISPLAY command shows information about registered products or the product enablement policy.

```
D PROD, {REGISTERED|REG}
      {STATE      }
      {STATUS     }
[,OWNER(o)] [,NAME(n)] [,FEATURENAME(fn)] [,ID(id)] [,ALL]
```

Example: The following DISPLAY command shows entries in the list of APF-authorized libraries.

```
D PROG,APF[,ALL
|,DSNAME=libname
|,ENTRY=xxx
|,ENTRY=(xxx-yyy)]
[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows the dynamic exits.

```
D PROG,EXIT, {{EXITNAME|EX|EN}=exitname } [,DIAG]
             {{EXITNAME|EX|EN}=exitname* }
             {{MODNAME|MOD}=modname      }
             {[ALL] [,IMPLICIT|,IMP]     }
[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows information about LNKLST sets for the LNKLST concatenation and associated jobs.

DISPLAY Command

```
D PROG,LNKLST[,NAME=[1nk1stname|CURRENT] ]
              [,NAMES ]
              [,USERS,[CURRENT|NOTCURRENT|NAME=1nk1stname]]
              [,ASID=asid ]
              [,JOBNAME=jobname ]
```

Example: The following DISPLAY command shows the entry point, load point, and length information about modules dynamically added to the LPA, and shows the minimum amount of CSA and ECSA that must remain after adding a module to the LPA.

```
D PROG,LPA{,MODNAME=modname}
           {,CSAMIN }

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

Example: The following DISPLAY command shows information about outstanding action messages (descriptor codes 1, 2, 3, and 11), WTORs, devices awaiting mount requests to be fulfilled, and units requiring intervention.

```
D R[,U ]
    [,KEY[,SYS=sysname] [,CN=(ALL)]
    [,I ] [,msgformat] [,MSG=msgid] [,SYS=sysname] [,KEY=keyname|MOUNT]
    [, ] [,JOB=jobname]
    [,E ] [,CN={xx|name|(ALL)}] [,ROUT={ALL|(rrr[,sss]...)}
    [,CE ] [(rrr-sss[,rrr-sss]...)]
    [,R
    [,M
    [, {LIST | L | ALL | A | }
    [,L={a|cc|cca|name|name-a}]

(See Note)
```

Note: If you supply all commas between DISPLAY R and the operands that have equal signs, you get default values. However, you should supply only one comma before the L operand, even if you omit the preceding operands.
Example: DISPLAY R,I,L=2B

Example: The following DISPLAY command shows the current status of the run-time library services (RTLS) environment. It lists information about the physical and logical libraries in use, the users of the logical libraries, and the cache use for a given library or for all libraries.

DISPLAY Command

```
D RTLS[,NAMES[,LIBRARY=lname[,VERSION=ver] [,CURRENT|,SEQNUM=num|,ALL] ]
[
  [,PHYSICAL,LIBRARY=pname[,CURRENT|,SEQNUM=num|,ALL] [,MODULE=mod|,LOGICAL]]
[
  [,LOGICAL{,LIBRARY=lname[,VERSION=ver] [,CURRENT|,SEQNUM=num|,ALL] } ]
[
  {,MODULE=mod|,USERS} ] ]
[
  {,JOBNAME=jobname } ] ]
[
  {,ASID=asid } ] ]
[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows detailed information about the SLIP trap identified by identifier **xxxx**. If **xxxx** is not specified, the system displays all traps and tells whether each is enabled.

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

Example: The following DISPLAY command shows the SMF data set names and their status (S) or the current SMF options (O).

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

Example: The following DISPLAY command shows storage management subsystem (SMS) information.

DISPLAY Command

```

D SMS[, {ACTIVE|A} ]
  [, CACHE ]
  [, CFCACHE(structurename|*) ]
  [, CFLS ]
  [, CFVOL(void) ]
  [, {DRIVE|DRI} (name|ALL) [, STATUS ] ]
  [, ,DETAIL ]
  [, {LIBRARY|LIB} (name|ALL) [, STATUS [, LISTDRI]] ]
  [, ,LISTDRI ]
  [, ,DETAIL ]
  [, MONDS(specmask|*) ]
  [, OAM ]
  [, OPTIONS ]
  [, OSMC [, TASK(name)] ]
  [, SHCDS ]
  [, SMSVSAM [, ALL] ]
  [, SMSVSAM, QUIESCE ]
  [, {STORGRP|SG} {(storgrp|ALL)} [, LISTVOL ] ]
  [, ,DETAIL ]
  [, {TRACE|T} ]
  [, {VOLUME|VOL} (volume) ]

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

```

Example: The following DISPLAY command shows information about all subsystems defined to MVS.

```

D SSI[, {LIST|L} | {ALL|A}] [, {DYNAMIC|DYN|D}={YES|Y} | {NO|N}]
  [, {FUNC|F}=funclist ]
  [, {STATUS|STAT|ST}={ACTIVE|ACT} | {INACTIVE|INACT|I} ]
  [, {SUBSYS|SUB}=subsysname ]

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

```

Example: The following DISPLAY command shows static system symbols and their associated substitution text.

```

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

```

DISPLAY Command

Example: The following DISPLAY command shows the local time of day and date and Greenwich Mean Time (GMT).

```
D T
```

Example: The following DISPLAY command shows component trace status.

```
D TRACE[,COMP=cname[,SUB=(subname)][,N=nnn] [,SUBLEVEL] ]
      [
      [,COMP={(cname[,cname]...) | ALL}
      [
      [,WTR={(name[,name],...) | ALL}
      [,TT
      ]

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

Example: The following DISPLAY command shows device status and allocation.

```
D {U[,devicetype] [,ONLINE ] [[, [/] devnum] [,nnnn]}
  {
  [,OFFLINE ]
  {
  [,ALLOC ]
  {
  [,AUTOSWITCH|AS]
  {U,IPLVOL
  {U,VOL=volser

(See Note)

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

Note: Supply all commas between DISPLAY U and a specified operand.

Example: The following DISPLAY command shows the workload management service policy, service definition, mode, application environments, resources, and scheduling environments.

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

Example: The following DISPLAY command shows a summary of the current sysplex.

DISPLAY Command

```

D XCF[, {PATHIN|PI} ]
[   [, {DEVICE|DEV}={ ([/] indevnum[, [/] indevnum...)] |ALL} ]
[   [, {STRNAME|STRNM}={ (strname[, strname]...) |ALL} ]
[   [, {SYSNAME|SYSNM}=(sysname[, sysname]...) ]
[   [, {STATUS|STAT}=( [STARTING] [, RESTARTING] [, WORKING]
[   [, STOPPING] [, STOPFAILED] [, INOPERATIVE]
[   [, LINKING] [, QUIESCING) ) ]
[ ]
[ ]
[ , {PATHOUT|PO} ]
[   [, {DEVICE|DEV}={ ([/] outdevnum[, [/] outdevnum...)] |ALL} ]
[   [, {STRNAME|STRNM}={ (strname[, strname]...) |ALL} ]
[   [, CLASS={ (classname[, classname]...) |ALL} ]
[   [, {SYSNAME|SYSNM}=(sysname[, sysname]...) ]
[   [, {STATUS|STAT}=( [STARTING] [, RESTARTING] [, WORKING]
[   [, STOPPING] [, STOPFAILED] [, INOPERATIVE]
[   [, LINKING] [, QUIESCING) ) ]
[   [, REBUILDING] [, QUIESCED) ) ]
[ ]
[ , {LOCALMSG|LM} [, CLASS={ (classname[, classname]...) |ALL} ]
[ ]
[ , {GROUP|GRP} ,groupname[, membername|ALL] ]
[ ]
[ , {SYSPLEX|S} [, systemname|ALL] ]
[ ]
[ , {COUPLE|CPL} [, TYPE={ (name[, name]...) |ALL} ]
[ ]
[ , {CLASSDEF|CD} ]
[   [, CLASS={ (classname| (classname[, classname]...) |ALL} ]
[   [, {GROUP|G}=groupname ]
[ ]
[ , {STRUCTURE|STR} ]
[   [, {STRNAME|STRNM}={ (strname[, strname]...) |ALL} ]
[   [, {CONNAME|CONNM}={ (conname[, conname]...) |ALL} ]
[   [, {STATUS|STAT}=( [ALLOCATED] [, NOTALLOCATED]
[   [, POLICYCHANGE] [, DEALLOCPENDING]
[   [, LARGERCFRMDS] [, REBUILD] [, STRDUMP]
[   [, ALTER] [, FPCONN] [, NOCONN) ) ]
[ ]
[ , {CF} [, {CFNAME|CFNM}={ (cfname[, cfname]...) |ALL} ]
[ ]
[ , {POLICY|POL} [, TYPE={ (name[, name]...) |ALL} ]
[ ]
[ , {PRSMPOLICY|PRSMPOL} ]
[ ]
[ , {ARMSTATUS|ARMS} ]
[   [, {RESTARTGRP|RG}=rgname ]
[   [, {ELEMENT|EL}=elname| {JOBNAME|JOB}=jobname ]
[   [, INITSYS=initsys ]
[   [, CURRSYS=currsys ]
[   [, STATE=( [ {STARTING|START} ] [, {AVAILABLE|AVAIL} ] [, FAILED]
[   [, {RESTARTING|RESTART} ] [, {RECOVERING|RECOVER} ) ] ]
[   [, DETAIL ]
[ ]
[ , L={ a | cc | cca | name | name-a } ]

```

DUMP Command

Purpose: Use the DUMP command to request a system dump of virtual storage (SVC dump). The system responds by prompting you in message IEE094D for the dump options. Specify these options by using the REPLY command. The title of the dump, which you specify by enclosing 1-100 characters in parentheses or single or double quotes, becomes the first record in the dump data set. COMM and TITLE are synonyms.

```
DUMP {COMM={ (title) } [,PARMLIB=xx]          }
      { 'title' } | [,PARMLIB=(xx[,xx]...)]
      { "title" } [SYMDEF=(symdef[,symdef]...)]
      {TITLE={ (title) }                      }
      { 'title' }
      { "title" }
```

DUMPDS Command (DD)

Purpose: Use the DUMPDS command to:

- Change the system's list of dump data sets and resources
- Clear full SYS1.DUMP data sets and make them available for dumps
- Set up and alter the configuration of automatic dump data set allocation

Example: The following DUMPDS command adds the specified direct access data sets to the list of SYS1.DUMP data sets.

```
DD ADD, {DSN={nn                               } }
        { { (nn[,nn]...)                       } }
        { { nn-nn                               } }
        { { (nn-nn[,nn-nn]...)                 } }
        { { (nn[,nn]...,nn-nn[,nn-nn]...)     } }
        { { ALL                                 } }
        { { (ALL)                              } }
        {                                       }
        {SMS={class                            } }
        { { (class[,class]...)                 } }
        {                                       }
        {VOL={volser                           } }
        { { (volser[,volser]...)              } }
```

Where **class** represents:

```
{storclas                                     }
{ ([DATA|D=[dataclas]] [,MGMT|M=[mgmtclas]] [,STOR|S=[storclas]]) }
```

Example: The following DUMPDS command specifies whether or not to allocate dump data sets automatically when requesting a dump.

```
DD ALLOC={ACTIVE|INACTIVE}
```

Example: The following DUMPDS command empties the specified data set and marks it available to receive a dump.

DUMPDS Command

```
DD CLEAR,DSN={nn                }
              {(nn[,nn]...)     }
              {nn-nn            }
              {(nn-nn[,nn-nn]...) }
              {(nn[,nn]...,nn-nn[,nn-nn]...)}
              {ALL              }
              {(ALL)            }
```

Example: The following DUMPDS command removes from the system's list of dump data set resources specific SYS1.DUMP data sets, SMS classes, or DASD volumes.

```
DD DEL,{DSN={nn                } }
        {  {(nn[,nn]...)     } }
        {  {nn-nn            } }
        {  {(nn-nn[,nn-nn]...) } }
        {  {(nn[,nn]...,nn-nn[,nn-nn]...)} }
        {  {ALL              } }
        {  {(ALL)            } }
        {  }
        {SMS={class           } }
        {  {(class[,class]...) } }
        {  {ALL              } }
        {  {(ALL)            } }
        {  }
        {VOL={volser          } }
        {  {(volser[,volser]...) } }
        {  {ALL              } }
        {  {(ALL)            } }
```

Where **class** represents:

```
{storclas
{([DATA|D=[dataclas]],[MGMT|M=[mgmtclas]],[STOR|S=[storclas]])}
```

Example: The following DUMPDS command establishes a name pattern for automatically allocated dump data sets.

```
DD NAME=name-pattern
```

FORCE Command

Purpose: Use the FORCE command to terminate a named batch job, started task, or APPC/MVS transaction program. **CAUTION:** Use this command as a last resort when the CANCEL command fails to perform its function after you have issued it several times.

```
FORCE {jobname                }[,ARM] [,A=asid] [,ARMRESTART]
      {U=userid                }
      {[jobname.]identifier}
```

WARNING: BE SURE YOU ARE AWARE OF ALL THE CONSEQUENCES OF ISSUING A FORCE COMMAND.

HALT Command (Z)

Purpose: Use the HALT command to record statistics before stopping the operating system. Store the internal I/O device error counts in the logrec data set, empty the SMF buffers on to the active SMF data set in SYS1.MANx, close the system log, and put it on the print queue.

```
Z EOD
```

IOACTION Command (IO)

Purpose: Use the IOACTION command to stop and resume I/O activity to direct access storage devices (DASDs) without varying the DASD offline, when the DASD is shared between systems AND is in recovery by the input/output system (IOS). **CAUTION:** Use this command only in response to the IOS recovery messages IOS427A and IOS062E and wait state X'062'.

```
IO {STOP,DEV=([/]devnum[/],[/]devnum)...}
   {STOP,DEV=([/]lowdevnum-[/]highdevnum[/],[/]lowdevnum-[/]highdevnum)...}
   {RESUME,DEV=([/]devnum[/],[/]devnum)...}|ALL
   {RESUME,DEV=([/]lowdevnum-[/]highdevnum[/],[/]lowdevnum-[/]highdevnum)...}
```

Note: You can enter individual device numbers and ranges on the same command. For example:

```
IO RESUME,DEV=(/2233,/990-/1012,160)
```

LIBRARY Command (LI)

Purpose: Use the LIBRARY command to perform any of several tasks associated with tape drives and tape volumes.

For a detailed discussion of the LIBRARY command, including the parameters, syntax, and the tasks that command can perform, refer to *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* and *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

LOG Command (L)

Purpose: Use the LOG command to make an entry into the system log (up to 122 characters), the OPERLOG, or both. The entry goes to the master console if the log is temporarily inactive.

```
L 'text'
```

LOGOFF Command

LOGOFF Command

Purpose: Use the LOGOFF command to log off from an MCS console. This command will end your terminal session.

```
LOGOFF
```

Notes:

1. You must issue LOGOFF when you leave your console and your installation requires operators to log on before issuing commands or performing secured functions.
2. When your installation requires a LOGON command, LOGOFF leaves the console in a secure state. The system does not accept commands from this console until another LOGON command is completed. The console is in roll mode (MODE=R).
3. When LOGON is automatic at your installation, the system issues another MCS LOGON command to that MCS console.

LOGON Command

Purpose: Use the LOGON command to identify yourself to the system when your installation requires operators to log on before issuing commands.

```
IEE187I  ENTER LOGON PARAMETERS  
  
LOGON {userid}      PASSWORD {password}  
  
GROUP [racfgroup]  SECLABEL [label]
```

MODE Command

Purpose: Use the MODE command to control the actions of recovery management when certain types of machine check interruptions occur.

Example: The following MODE command controls the recording of hard machine check interruptions.

```
MODE {PD} [, INTERVAL={nnnn}] [, RECORD[=nnn ] [, CPU={x }]  
    {SD}           {300 }           |=ALL   {ALL}  
    {IV}           |                 |=25  
    {TC}           |                 |=16  
    {PT}           |                 |=5  
    {CC}  
    {VS}  
    {PS}  
    {AD}  
    {SL}  
    {SC}
```


Example: The following MODE command controls the recording and reporting of system recovery and degradation machine check interruptions.

```
MODE {SR} [,QUIET ] [,CPU={x|ALL}]
      {DG} |,RECORD[=nnn ]
              |
              | =ALL[,REPORT=nnn]
              | =50
              | =1
```

Example: The following MODE command displays the event counters and recording monitoring status associated with each type of machine check interruption.

```
MODE [STATUS]
```

MODIFY Command (F)

Purpose: Use the MODIFY command to pass information to a job or started task. You can communicate with a currently running program only if it was designed to recognize input from the MODIFY command.

Example: The following MODIFY command changes the job parameters as the programmer specifies.

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

Example: The following MODIFY command starts TSO/TCAM time-sharing once TCAM is active.

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

Example: The following MODIFY command stops TSO/TCAM time-sharing.

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

Example: The following MODIFY command modifies TSO/VTAM time-sharing.

```
F [jobname.]identifier,{USERMAX=nnnn}
                        {USER={SIC } }
                        {      {FSTOP} }
```

MODIFY Command

Example: The following MODIFY command controls OS/390 UNIX System Services, terminates an OS/390 UNIX process or thread, shuts down the OS/390 UNIX initiators, or requests a SYSMDUMP for a process.

```
F BPX0INIT,{APPL=appl_data}
           {DUMP=pid}
           {FILESYS=}
           {FORCE=pid[.tid]}
           {RESTART=FORKS}
           {SHUTDOWN={FILESYS | FORKINIT | FORKS}}
           {TERM=pid[.tid]}
```

Example: The following MODIFY command communicates with the catalog address space. Use this command only at the direction of the system programmer.

```
F CATALOG,{ABEND{(id)          }
           {          {(yyyyyyy)}
           {          {(ALLOCATE)}
           {          {(ANALYSIS)}
           {          {(MODIFY)  }
           {ALIASLEVEL(n)
           {ALLOCATE(catname)[,{NOISC | NOVLF}]
           {ALLOCATED[(vvvvvv)]
           {CATMAX(nnn)
           {CLOSE(catname)
           {{DUMPON | DUMPOFF}
           {ECSHR({(ADD,catname)  })
           |{(REMOVE,catname)
           |{(STATUS,catname)
           {ECSHR(AUTOADD)
           {ECSHR({CONNECT  })
           |DISCONNECT
           |STATUS
           {END(id)[,{REDRIVE | NOREDRIVE}]
           {ENTRY[(cname) | (mmmmmmm)]
           {{ISC | NOISC}(catname)
           {LIST[(id) | (yyyyyyy)]
           {LISTJ[jobname]
           {OPEN[(vvvvvv)]
           {REPORT
           {REPORT,CACHE[(catname)]
           {REPORT,DUMP
           {REPORT,PERFORMANCE[(RESET)]
           {RESTART
           {{ROTATE | NOROTATE}
           {{SYS%ON | SYS%OFF}
           {TASKMAX(nnn)
           {UNALLOCATE[catname]
           {VCLOSE(vvvvvv)
           {{VLF | NOVLF}(catname)
           {{VUNALLOCATE | NOVUNALLOCATE}
           {{WARNING | NOWARNING}
```

Example: The following MODIFY command changes the processing mode for the data lookaside facility (DLF).

MODIFY Command

```
F DLF,MODE={DRAIN|D }
           {QUIESCE|Q}
           {NORMAL|N }
```

Example: The following MODIFY command causes the data lookaside facility (DLF) to use the specified COFDLFxx member of SYS1.PARMLIB.

```
F DLF,NN=xx
```

Example: The following MODIFY command displays the limits from the COFDLFxx parmlib member currently in effect.

```
F DLF[, {STATUS|ST|S}]
      |,SM
      |,SB
```

Example: The following MODIFY commands builds and replaces library lookaside (LLA) directories.

```
F LLA, {REFRESH }
      {UPDATE=xx}
```

Example: The following MODIFY command operates with the network file system server.

```
F [MVS NFS. |jobname.] identifier,
   {FREEZE={ON|OFF}           }
   {LIST={MOUNTS|DSNAMES}     }
   {RELEASE=datasetname[(member)] }
   {STATUS                     }
   {STOP                       }
   {UNMOUNT=name              }
```

Example: The following MODIFY command diagnoses problems for the network file system server.

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

Example: The following MODIFY command displays information about the object access method (OAM) or requests that OAM perform a specified service: object management, space management, or recovery functions.

MODIFY Command

```
F [OAM.|jobname.]identifier,
  {{LABEL|L}[,9247      ] }
  {      |,3995-133    }
  {      |,3995WORM    }
  {      |,3995REWR    }
  {{START|S},{OSMC      } }
  {      {STORGRP,stogrprname } }
  {      {LIBMGT,libraryname  } }
  {      {DASDSM,stogrprname  } }
  {      {RECOVERY,volser     } }
  {      {OBJRECV,collectionname,objectname} }
  {      {MOVEVOL,volser      } }
  {{STOP|P},{OAM        } }
  {      {OSMC          } }
  {      {STORGRP,stogrprname} }
  {      {MOVEVOL,volser     } }
  {{DISPLAY|D},{GROUP[,stogrprname]} }
  {      {VOL,volser       } }
```

Example: The following MODIFY command switches workload management modes and changes resource settings.

```
F WLM,[MODE={GOAL|COMPAT}]
      ,[RESOURCE=resourceName,{ON|OFF|RESET}]
```

Example: The following MODIFY command specifies the criteria that an external writer is to use in selecting data sets for processing.

```
F [XWTR.|jobname.]identifier,
  {{CLASS|C}=[classes]    }
  {{DEST|D}=[LOCAL        ]}
  {      |remote-workstation-name }
  {{FORMS|F}=[forms-name] }
  {{JOBID|J}=[JOBnnnn ]   }
  {      |STCnnnn         }
  {      |TSUnnnn         }
  {{WRITER|W}=[STDWTR     ]}
  {      |user-writer-name }
```

Example: The following MODIFY command causes an external writer to pause.

```
F [XWTR.|jobname.]identifier,{PAUSE|P}={FORMS }
                                     {DATASET}
```

MONITOR Command (MN)

Purpose: Use the MONITOR command to display jobnames, data set status, and time-sharing user sessions continuously, and to add certain information to mount and demount messages.

```
MN {JOBNAMES[,T] [,L={a|cc|cca|name|name-a}] }
    {DSNAME }
    {SPACE }
    {STATUS[,L={a|cc|cca|name|name-a}] }
    {SESS[,T] [,L={a|cc|cca|name|name-a}] }
```

MOUNT Command (M)

Purpose: 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.

```
M {[/]devnum } ,VOL=({NL},serial) [,USE={STORAGE}]
   {devicetype } {SL} {PUBLIC }
                  {AL} {PRIVATE}
```

MSGRT Command (MR)

Purpose: 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.

```
MR { [D=(operand[,operand]...)] [,L={a } ] }
    { |TR=A } {cc } }
    { |K } {cca } }
    { |CF } {name } }
    { |MN } {name-a} }
    { }
    {REF }
    { }
    {NONE }
    { }
    {CONTn } }
```

PAGEADD Command

PAGEADD Command (PA)

Purpose: Use the PAGEADD command to add auxiliary storage space (local page data sets and swap data sets) to the system. **CAUTION:** Use this command only at the direction of your system programmer.

```
PA { [PAGE=] } { dsname[, dsname] ... }  
  { SWAP= }  
  { NONVIO= }
```

PAGEDEL Command (PD)

Purpose: Use the PAGEDEL command to delete, replace, or drain (quiesce) local page data sets or swap data sets. **WARNING:** Use this command only at the direction of your system programmer. Misuse can seriously impact system performance by causing a shortage of auxiliary storage.

```
PD { DELETE, { PAGE | SWAP } = { dsname[, dsname] ... } }  
  {  
    { REPLACE, { PAGE | SWAP } = { (dsname, rdsname) [, (dsname, rdsname)] ... } }  
    {  
      { DRAIN, { PAGE | SWAP } = { dsname[, dsname] ... } }  
    }  
  }
```

QUIESCE Command

Purpose: Use the QUIESCE command to put the system in a manual state without affecting job step timing.

```
QUIESCE
```

REPLY Command (R)

Purpose: Use the REPLY command to respond to system requests for information. The verb (REPLY, or R) is not required when you respond to a request. The general syntax for the REPLY command is:

```
[R] id[,][ 'text' | text]
```

Example: The following REPLY command would be your response to system requests during recovery processing.

```
R [00|0] [,][ 'text' | text]
```

REPLY Command

Example: The following REPLY command sets the time of day clock. If you specify GMT, the system assumes Greenwich Mean Time. Otherwise, it assumes local time.

```
R 00, ' [DATE=yyyy.ddd] [,CLOCK=hh.mm.ss] [,GMT] '
```

Example: The following REPLY command would be your response to message ITT006A, which the system issues after you issue a TRACE CT (component trace) command, to prompt you for the options you want to specify.

```
R id[,ASID=(nnnn[,nnnn]...)]  
    [,JOBNAME=(name[,name]...)]  
    [,OPTIONS=(name[,name]...)]  
    [,WTR={membername|DISCONNECT}]  
    [,CONT|,END ]
```

Note: When you specify CONT or END, it must be the last parameter on the input line.

Example: The following REPLY command would be your response to message IEE094D, which the system issues after you issue a DUMP command, to prompt you for the dump options you want to specify.

```
R id,U  
  
or  
  
R id[,ASID=(n[,n]...)] [,JOBNAME=(name[,name]...)] [,TSONAME=(name[,name]...)]  
    [,DSPNAME=(dspname-entry[,dspname-entry]...)]  
    [, {PROBDESC|PROB|PD}=key-spec] [,REMOTE=(request[,request]...)]  
    [,SDATA[=(option[,option]...)] [,STOR=(beg,end[,beg,end]...)]  
    [,STRLIST=(s-option[,s-option]...)]  
    [,CONT|,END]
```

Notes:

1. When you specify CONT or END, it must be the last parameter on the input line.
2. The CONT keyword does not work within a SYSP= list.
3. When you specify U, it must be the first parameter following the identification number.

Where **request** represents:

```
{GRPLIST={group(member) } }  
{ { (group(member[,member]...)[,group(member[,member]...)]... ) } }  
{ } }  
{SYSLIST={sysinfo| (sysinfo[,sysinfo]...)} }  
{ [,DSPNAME|,DSPNAME=(dspname-entry[,dspname-entry]... ) ] }  
{ [,SDATA|,SDATA=(option[,option]... ) ] }  
{ [,STOR|,STOR=(beg,end[,beg,end]... ) ] }
```

REPLY Command

Where **s-option** represents:

```
STRNAME=strname
[ ,CONNAME=conname ]
[ ]
[ ,ACCESSTIME={ENFORCE|NOLIMIT|NOLIM} ]
[ ]
[ ,LOCKENTRIES ]
[ ]
[ ,USERCNTLS ]
[ ]
[ ,EVENTQS ]
[ ]
[ , (EMCONTROLS={ALL|(list)}) ]
[ ]
[ , ( {COCLASS|STGCLASS|LISTNUM}={ALL|(list)} ]
[ { [,ADJUNCT={CAPTURE|DIRECTIO}] [,ENTRYDATA={UNSERIALIZE|SERIALIZE}] } ]
[ { [,SUMMARY] } ) ]
```

Example: The following REPLY command would be your response to message IEE357A, which the system issues to allow you either to specify that you do not want to change the SMF values (U), or to specify the options (separated by commas) your system programmer provides.

```
R id,{U
    {options}}
```

Example: The following REPLY command would be your response to message IEA101A, which the system issues during system initialization, to prompt you to specify system parameters.

```
R [0|00],{U
    {
    {'parm=',' } [,CONT] }
    {'parm=parm,' } }
    {'parm=value' } }
    {'parm=(value[,value]...[,L])' } }
    {'parm=(value[,value]...[,L]),parm=value'}}
```

RESET Command (E)

Purpose: Use the RESET command to force a hung MCS console offline or to change certain aspects of work currently executing.

Example: The following RESET command forces a hung MCS console offline.

```
E [CN(consname)]
```


RESET Command

Example: The following RESET command can change the performance group or service class of an executing job, time-sharing user, or started task. Also, in goal mode only, it can quiesce a problem job or address space or resume a quiesced job or address space. **CAUTION:** Use this command only at the direction of the system programmer.

```
E jobname[,A=asid],{PERFORM=nnn      }
                    {SRVCLASS=classname}
                    {QUIESCE|Q       }
                    {RESUME           }
```

ROUTE Command (RO)

Purpose: Use the ROUTE command to direct a command to one or more systems in a sysplex for processing.

```
RO {sysname,text      }
   {
   { [T=nnn,] { *ALL           } [,L={a      } ] }
   {          { sysgrpname     } {cc      } }
   {          { *OTHER         } {cca     } }
   {          { (sysname[,sysgrpname,sysname...]) } {name   } }
   {          { name-a         } {name-a  } }
```

SEND Command (SE)

Purpose: 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.

Example: The following SEND command communicates messages to other operators at MCS consoles.

```
SE {'message'},{BRDCST      }
   {msgno      } {OPERATOR=routecode}
                   {CN=console  }
```

Example: The following SEND command communicates with specific time-sharing users.

```
SE {'message'},USER=(userid[,userid]...),{NOW|LOGON},{WAIT|NOWAIT}
   {msgno      }
```

Example: The following SEND command sends a message to all terminal users currently logged on the system.

SEND Command

```
SE {'message'}, {NOW|LOGON}, {WAIT|NOWAIT}, {ROUTE={*ALL|systemname|groupname}}
{msgno }
```

Example: The following SEND command stores messages in the broadcast data set.

```
SE {'message'}, {USER=(userid[,userid]...)}, SAVE
{ msgno } {ALL }
```

Example: The following SEND command lists the requested message (or all messages if that operand is omitted) in the notices section of the broadcast data set.

```
SE msgno,DELETE
```

Example: The following SEND command deletes the specified message from the broadcast data set.

```
SE [msgno,]LIST
```

SET Command (T)

Purpose: Use the SET command to initialize the local time and date and many of the system parameters in SYS1.PARMLIB. Use this command only on direction of the system programmer.

```
T [[DATE=yyyy.ddd] [,CLOCK=hh.mm.ss]] [RESET]
[,IPS=xx] [,OPT=xx] [,ICS=xx] [,SMF=xx] [,DAE=xx]
[,MPF={ (xx[,xx]...) |NO}]
[,SLIP=xx] [,PFK=xx] [,IOS=xx] [,EXS=xx] [,SMS=xx] [,MMS={xx|NO}] [,PROG=xx]
[,DIAG=xx] [,GRSRNL=(xx[,xx]...)]
[,APPC=(xx[,xx]...[,L])] [,ASCH=(xx[,xx]...[,L])] [,SCH=(xx[,xx]...[,L])]
[,CNGRP={ (xx[,xx]...) |NO }] [,PROD=(xx[,xx]...)]
[,OMVS=(xx[,yy]...[,nn])]
[,RTLS=(xx[,xx]...)]
```

Note: You may specify the operands in any order, and must specify at least one operand. Do not put a comma before the first operand you specify. If you specify DATE or RESET in a position other than the first, be sure to precede it with a comma. If you specify only one parmlib member with APPC=, ASCH=, SCH=, GRSRNL=, CNGRP=, MPF=, OMVS=, or RTLS=, you do not need to enter the parentheses.

SETDMN Command (SD)

Purpose: Use the SETDMN command to change existing values of parameters in a single domain. Issue it only on direction of the system programmer. It is not valid on systems operating in workload management goal mode.

```
SD domainnum, {MIN=n1 [, MAX=n2] [, ASRV=(n0, n9) ] }
               {
               |, DSRV=(n0, n9) }
               |, FIXCIDX=nnn }
               }
               { {ASRV=(n0, n9) } [, MIN=n1] [, MAX=n2] }
               { {DSRV=(n0, n9) } }
               { {FIXCIDX=nnn } }
               }
               { MAX=n2 [, ASRV=(n0, n9) ] [, MIN=n1] }
               {
               |, DSRV=(n0, n9) }
               |, FIXCIDX=nnn }
               }
```

SETETR Command

Purpose: Use the SETETR command to enable external time reference (ETR) ports that have been disabled.

```
SETETR PORT=n
```

SETGRS Command

Purpose: Use the SETGRS command to migrate a currently active GRS (global resource serialization) ring complex to a GRS star complex or to modify the current RESMIL or TOLINT values.

```
SETGRS {MODE=STAR }
        { [RESMIL=nnnn] [, TOLINT=nnnn] [, SYNCHRES={YES|NO}] }
        |RESMIL=OFF
```

SETIOS Command

Purpose: Use the SETIOS command to add, delete, modify, or replace any previously-specified missing interruption handler (MIH) or input/output timing (IOT) parameter dynamically.

```
SETIOS DCM={YES|NO|REFRESH}
        MIH[, class=mm:ss [, class=mm:ss] ...]
           [, MOUNTMSG={YES|NO}]
           [, DEV={{[/]devnum [, [/]devnum} ...} }, TIME=mm:ss, IOTIMING=mm:ss]
           { {[/]lowdevnum-[/]highdevnum } }
           [, MSGONLY={YES|NO}]
```

SETLOAD Command

SETLOAD Command

Purpose: Use the SETLOAD command to switch dynamically from one parmlib concatenation (logical parmlib) to another without having to initiate an IPL.

```
SETLOAD xx,PARMLIB[, {DSNAME|DSN}=dsn]  
[, {VOLUME|VOL|VOLSER}=vo1]
```

SETLOGRC Command

Purpose: Use the SETLOGRC command to change the logrec error and environmental recording medium.

```
SETLOGRC {LOGSTREAM|DATASET|IGNORE}
```

SETOMVS Command

Purpose: Use the SETOMVS command to change the options dynamically that OS/390 UNIX System Services uses. These options are originally set in the BPXPRMxx member of SYS1.PARMLIB at the time of initially program loading (IPL'ing) the system.

SETOMVS Command

SETOMVS	SETOMVS EXTENSIONS (sysplex exclusive)
SETOMVS [FORKCOPY=(COPY COW)] [,IPCSEMNIDS=ipcsemnids] [,IPCSEMNOPS=ipcsemnops] [,IPCSEMNSEMS=ipcsemnsems] [,IPCMSGQBYTES=ipcmsgqbytes] [,IPCMSGNIDS=ipcmsgnids] [,IPCSHMMPAGES=ipcshmpages] [,IPCSHMNIDS=ipcshmnids] [,IPCSHMNSEGS=ipcshmnsegs] [,IPCSHMSPAGES=ipcshmspapes] [,IPCMSGQMNUM=ipcmsgqmnum] [,LIMMSG=NONE SYSTEM ALL] [,MAXASSIZE=maxassize] [,MAXCORESIZE=maxcoresize] [,MAXCPUIME=maxcpuime] [,MAXFILEPROC=maxfileproc] [,MAXFILESIZE=(maxfilesize NOLIMIT)] [,MAXMMAPAREA=maxmmaparea] [,MAXPROCSYS=maxprocsys] [,MAXPROCUSER=maxprocuser] [,MAXPTYs=maxpty] [,MAXSHAREPAGES=maxsharepages] [,MAXTHREADS=maxthreads] [,MAXTHREADTASKS=maxthreadtasks] [,MAXUIDS=maxuids] [,PID=pid,processlimitname=newvalue] [,PRIORITYGOAL=(n) NONE] [,PRIORITYPG=(n) NONE]; [,STEPLIBLIST='stepliblist'] [,SUPERUSER=superuser] [,SYNTAXCHECK='parmlibmember'] [,TTYGROUP=ttygroup] [,USERIDALIASTABLE=useridaliastable] [,VERSION='string']	SETOMVS FILESYS ,FILESYSTEM=filesystem ,AUTOMOVE=YES NO ,SYSNAME=sysname * or SETOMVS FILESYS ,FILESYSTEM=filesystem ,AUTOMOVE=YES NO or SETOMVS FILESYS ,FILESYSTEM=filesystem ,SYSNAME=sysname * or SETOMVS FILESYS ,MOUNTPOINT=mountpoint ,AUTOMOVE=YES NO ,SYSNAME=sysname * or SETOMVS FILESYS ,MOUNTPOINT=mountpoint ,AUTOMOVE=YES NO or SETOMVS FILESYS ,MOUNTPOINT=mountpoint ,SYSNAME=sysname * or SETOMVS FILESYS ,FROMSYS=sysname ,SYSNAME=sysname * Note: FILESYSTEM, MOUNTPOINT, and FROMSYS are mutually exclusive parameters. When you specify FILESYS, you must supply one of these three parameters.

SETPROG Command

Purpose: Use the SETPROG command to update the authorized program facility (APF) list, dynamic exits, and LNKLIST concatenations, and to manage the LPA content.

Example: The following SETPROG command updates the APF list.

```
SETPROG APF{,FORMAT={DYNAMIC|STATIC}
           {
           {,{ADD|DELETE},DSNAME|LIBRARY=libname,{SMS|VOLUME=volume} }
           }
```

Example: The following SETPROG command updates dynamic exits.

SETPROG Command

```

SETPROG EXIT, {ADD, EXITNAME=exitname, MODNAME=modname      }
              { [ , STATE={ACTIVE|INACTIVE}]                }
              { [ , DSNAME=dsname]                          }
              { [ , JOBNAME={jobname|*}]                     }
              { [ , ABENDNUM=(n[ , CONSEC])]                  }
              { [ , FIRST|LAST]                               }
              { }
              {ATTRIB, EXITNAME=exitname, KEEPRC=(compare, kk)}
              { }
              {DELETE, EXITNAME=exitname, MODNAME=modname    }
              { [ , FORCE={YES|NO}]                           }
              { }
              {MODIFY, EXITNAME=exitname, MODNAME=modname     }
              { [ , STATE={ACTIVE|INACTIVE}]                 }
              { [ , JOBNAME={jobname|*}]                     }
              { }
              {UNDEFINE, EXITNAME=exitname                    }
              { }

```

Example: The following SETPROG command updates linklist concatenations by:

- Defining a LNKLST set of data sets for the LNKLST concatenation
- Adding data sets to or deleting data sets from the LNKLST set
- Removing the definition of a LNKLST set from the system
- Testing for the location of a specific module in the LNKLST concatenation
- Activating a LNKLST set as the LNKLST concatenation for the system
- Updating an address space for jobs to use a LNKLST set.

```

SETPROG LNKLST, {DEFINE, NAME=lnklstname[ , COPYFROM=lnklstname] [ , NOCHECK] }
               {ADD, NAME=lnklstname,
               DSNAME=dsname[ , VOLUME=volser] [ , ATBOTTOM      ]
               [ , ATTOP          ]
               [ , AFTER=dsname]
               [ , CONCAT(CHECK | NOCHECK)] }
               {DELETE, NAME=lnklstname, DSNAME=dsname }
               {UNDEFINE, NAME=lnklstname }
               {TEST, NAME=lnklstname, MODNAME=name }
               {ACTIVATE, NAME=lnklstname }
               {UPDATE, {JOB=jobname} }
               {ASID=asid }
               {UNALLOCATE }
               {ALLOCATE }

```

Example: The following SETPROG command specifies, any time after IPL, modules to add to or delete from the LPA, and the minimum amounts of CSA storage that must remain available after an ADD operation.

```

SETPROG LPA, {ADD, [MODNAME=(modname... , modname) | MASK=mask]
             { [ , DSNAME=[dsname | LNKLST] ] }
             { [ , FIXED] [ , PAGEPROTPAGE] }
             { }
             {DELETE, MODNAME=(modname... , modname) }
             { [ , FORCE=YES [CURRENT | OLDEST] ] }
             { }
             {CSAMIN=(below, above) }
             { }

```

SETRRS CANCEL Command

Purpose: Use the SETRRS CANCEL command to cancel (abnormally terminate) resource recovery services (RRS).

```
SETRRS CANCEL[,DUMP|NODUMP]
```

SETSMF Command (SS)

Purpose: Use the SETSMF command 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. Use the abbreviation SS to contrast this SETSMF command with the SET SMF command.

```
SS parameter(value[,value]...)
```

SETSMS Command

Purpose: Use the SETSMS command to change a subset of storage management subsystem (SMS) parameters from the console without changing the active IGDSMSxx member of SYS1.PARMLIB. Contrast this SETSMS command with the SET SMS command.

```
SETSMS parameter(value)[,parameter(value)]...
```

SETSSI Command

Purpose: Use the SETSSI command to add, activate, or deactivate a subsystem dynamically.

```
SETSSI {ADD,{SUBNAME|SUB|S}=subname
        [, {CONSNAME|C}=consname]
        [, {INITRTN|I}=initrtn[, {INITPARM|P}=initparm] }
        {
        {DEACTIVATE|DEACT},{SUBNAME|SUB|S}=subname }
        {
        {ACTIVATE|ACT},{SUBNAME|SUB|S}=subname }
        }
```

SETXCF Command

Purpose: Use the SETXCF command to control the cross-system coupling facility (XCF).

Example: The following SETXCF COUPLE command can switch the current alternate XCF couple data set to the primary XCF couple data set, or specify an alternate XCF couple data set, or change options specified in the COUPLExx parmlib member.

SETXCF Command

```
SETXCF COUPLE, {PSWITCH }
                { }
                {ACOUPLE=(alternatedsname[,alternatevolume]) }
                { }
                {INTERVAL=timeinterval }
                { }
                {OPNOTIFY=timeinterval }
                { }
                {CLEANUP=timeinterval }
                { }
                {MAXMSG=defaultmaxmsgbuffers }
                { }
                {RETRY=defaultretrylimit }
                { }
                {CLASSLEN=defaultclasslength }
                { }
                {TYPE=(name[,name]...), }
                { }
                {   {PCOUPLE=(primarydsname[,primaryvolume]) } }
                {   {ACOUPLE=(alternatedsname[,alternatevolume])} }
                {   {PSWITCH } }
                { }
                { }
```

Example: The following SETXCF FORCE command cleans up resources related to structures in a coupling facility. The resources can be either structures actively in use in the sysplex, or dumps associated with structures pending deallocation.

```
SETXCF FORCE,
  {STRUCTURE,STRNAME=(strname[,strname]...)}
  {CONNECTION,STRNAME=strname,CONNAME={ (conname[,conname]...) | ALL} }
  {STRDUMP,STRNAME=strname[,STRDUMPID=strdumpid] }
  { }
  {STRDUMPSERIAL,STRNAME=strname[,STRDUMPID=strdumpid] }
  { }
```

Example: The following SETXCF MODIFY command changes current XCF parameters.

SETXCF Command

```
SETXCF MODIFY, {PATHIN, {DEVICE=([/] indevnum[, [/] indevnum]...)} }
{
  {STRNAME=(strname[, strname]...)} }
{
  [,MAXMSG=maxmsgbuffers] }
{
  [,RETRY=retrylimit] }
{
  }
{PATHOUT, {DEVICE=([/] outdevnum[, [/] outdevnum]...)} }
{
  {STRNAME=(strname[, strname]...)} }
{
  [,CLASS=classname] }
{
  [,MAXMSG=maxmsgbuffers] }
{
  [,RETRY=retrylimit] }
{
  }
{LOCALMSG, MAXMSG=maxmsgbuffers }
{
  [,CLASS=class-name] }
{
  }
{CLASSDEF, CLASS=classname }
{
  [,CLASSLEN=classlength] }
{
  [,MAXMSG=defaultmaxmsgbuffers] }
{
  [,ADDGROUP=(groupname[, groupname]...)] }
{
  [,DELGROUP=(groupname[, groupname]...)] }
}
```

Example: The following SETXCF PRSMPOLICY (or PRSMPOL) command either activates an XCF PR/SM policy or deactivates a currently active XCF PR/SM policy.

```
SETXCF PRSMPOLICY, {DEACTIVATE|ACTIVATE=memname}
```

Example: The following SETXCF START command starts signalling paths and defines transport class definitions for use by the XCF.

```
SETXCF START, {CLASSDEF, CLASS=classname }
{
  [,CLASSLEN=classlength] }
{
  [,MAXMSG=maxmsgbuffers] }
{
  [,GROUP=(groupname[, groupname]...)] }
{
  }
{PATHIN, {DEVICE=([/] indevnum[, [/] indevnum]...)} }
{
  {STRNAME=(strname[, strname]...)} }
{
  [,MAXMSG=maxmsgbuffers] }
{
  [,RETRY=retrylimit] }
{
  }
{PATHOUT, {DEVICE=([/] outdevnum[, [/] outdevnum]...)} }
{
  {STRNAME=(strname[, strname]...)} }
{
  [,MAXMSG=maxmsgbuffers] }
{
  [,RETRY=retrylimit] }
{
  [,CLASS=classname] }
{
  }
{POLICY, TYPE=name, POLNAME=polname }
{
  }
{REBUILD, {POPULATECF=cfname} }
{
  {DUPLEX, } }
{
  {STRNAME=(strname[, strname]...)} }
{
  {CFNAME=(cfname[, cfname]...)} }
{
  [,LOCATION={NORMAL|OTHER}] }
{
  [,LESSCONN={TERMINATE|CONTINUE}] }
{
  }
{ALTER, STRNAME=strname, SIZE=size }
}
```

SETXCF Command

Example: The following SETXCF STOP command removes XCF signalling paths and transport class definitions.

```

SETXCF STOP, {PATHIN, {DEVICE=([/]indevnum[, [/]indevnum]...)} }
              {          {STRNAME=(strname[, strname]...)} }
              {          [, UNCOND=NO|YES] }
              {          }
              {PATHOUT, {DEVICE=([/]outdevnum[, [/]outdevnum]...)} }
              {          {STRNAME=(strname[, strname]...)} }
              {          [, UNCOND=NO|YES] }
              {          }
              {CLASSDEF, CLASS=classname }
              {          }
              {POLICY, TYPE=name }
              {          }
              {REBUILD, {POPULATECF=cfname} }
              {          {DUPLEX,} }
              {          {STRNAME=(strname[, strname]...)} }
              {          {, KEEP=NEW|OLD} }
              {          {CFNAME=(cfname[, cfname]...)} }
              {          }
              {ALTER, STRNAME=strname }

```

SLIP Command (SL)

Purpose: Use the SLIP command to control serviceability level indication processing (SLIP), a diagnostic aid that intercepts or traps certain system events. The three types of the SLIP command set, modify, and delete SLIP traps. This table shows an overview of the SLIP command syntax.

SLIP SET[,options],END	Command for an error event trap (non-PER)
SLIP SET,IF[,options],END	Command for an instruction fetch PER trap
SLIP SET,SB1[,options],END SLIP SET,SBT[,options],END	Commands for a successful branch PER trap
SLIP SET,SA[,options],END SLIP SET,SAS[,options],END	Commands for a storage alteration PER trap
SLIP MOD[,options]	Command to modify an existing trap
SLIP DEL[,options]	Command to delete an existing trap

Notes:

- You must specify SET, MOD, or DEL immediately following SLIP.
- If you specify IF, SB1, SBT, SA, or SAS, it must immediately follow SET.
- You must specify END at the end of all SLIP SET commands.

Examples: The following tables show the syntax for various forms of the SLIP SET command.

```

SLIP SET

[,ADDRESS=(start[,end])          ]
[,LPAEP=(name[,start[,end]])    ]
[,LPAMOD=(name[,start[,end]])   ]
[,NUCEP=(name[,start[,end]])   ]
[,NUCMOD=(name[,start[,end]])  ]
[,PVTEP=(name[,start[,end]])   ]
[,PVTMOD=(name[,start[,end]])   ]

[,ASID=(asid[,asid]...)         ]
[,COMP=code[,REASON=code]       ]
[,DATA=(comparison[,comparison]...) ]
[,ERRTYP=(type[,type]...)       ]
[,JOBNAME={userid | jobname}     ]
[,JSPGM=name                     ]
[,MODE= (mode[,mode]...[,ANY | EVERY]) ]
[,PSWASC=(mode[,mode]...)       ]

[,ACTION=[IGNORE[,option]       ]
          [(nodump[,nodump]...)[,option] ]
          [NOSUP[,option]         ]
          [RECORD                 ]
          [SVCD[,options]         ]
          [TRACE[,options]        ]
          [TRDUMP[,options]       ]
          [WAIT[,options]         ]

[,ENABLE | ,DISABLE]

[,IDGROUP=idgroup]

[,MATCHLIM=m ]
[,MATCHLIM=1  for ACTION=SVCD or ACTION=SYNCSVCD

[,DEBUG]

[,ID=trapid]

[,OK]

[,RBLEVEL={ERROR | NOTSVRB | PREVIOUS} ]

,END

```

SLIP Command

```
SLIP SET, {IF }
          {SB1}
          {SBT}

[,LPAEP=(name[,start[,end]]) ]
[,LPAMOD=(name[,start[,end]])]
[,NUCEP=(name[,start[,end]])]
[,NUCMOD=(name[,start[,end]])]
[,PVTEP=(name[,start[,end]])]
[,PVTMOD=(name[,start[,end]])]
[,RANGE=(start[,end])

[,ASID=(asid[,asid]...)]

[,DATA=(comparison[,comparison]...)]

[,JOBNAME={userid | jobname}]

[,JSPGM=name]

[,MODE= (mode[,mode]...[,ANY | EVERY])]

[,PSWASC=(mode[,mode]...)]

[,ACTION=[(IGNORE[,RECOVERY] ) ]
          [(RECOVERY[,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(STDUMP[,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(STRACE[,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(SVCD[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(SYNCSVCD[,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(TRACE[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(TRDUMP[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(WAIT[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]

[,ENABLE | ,DISABLE]

[,IDGROUP=idgroup]

[,MATCHLIM=m ]
[,MATCHLIM=1 for ACTION=SVCD or ACTION=SYNCSVCD
[,MATCHLIM=50 for ACTION=STDUMP or ACTION=STRACE

[,PRCNTLIM=p | ,PRCNTLIM=10]

[,DEBUG]

[,ID=trapid]

[,OK]

,END
```

```

SLIP SET,{SA }
        {SAS}

[,ADDRESS=(start[,end]          )
  [,LPAEP=(name[,start[,end]])
  [,LPAMOD=(name[,start[,end]])
  [,NUCEP=(name[,start[,end]])
  [,NUCMOD=(name[,start[,end]])
  [,PVTEP=(name[,start[,end]])
  [,PVTMOD=(name[,start[,end]])

[,RANGE=(start[,end])]
[,ASID=(asid[,asid]...)]
[,ASIDSA=(asid | 'jobname'[,asid | , 'jobname']...)]
[,DATA=(comparison[,comparison]...)]
[,DSSA=(asid.name | 'jobname'.name[,asid.name | , 'jobname'.name]...)]
[,JOBNAME={userid | jobname}]
[,JSPGM=name]
[,MODE= (mode[,mode]...[,ANY | EVERY])]
[,PSWASC=(mode[,mode]...)]

[,ACTION=[(IGNORE[,RECOVERY]                                )]]
          [(RECOVERY[,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(STDUMP[,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(STRACE[,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(SVCD[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(SYNCSVCD[,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(TRACE[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(TRDUMP[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
          [(WAIT[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]

[,ENABLE | ,DISABLE]

[,IDGROUP=idgroup]

[,MATCHLIM=m ]
  [,MATCHLIM=1   for ACTION=SVCD or ACTION=SYNCSVCD
  [,MATCHLIM=50 for ACTION=STDUMP or ACTION=STRACE

[,PRCNTLIM=p | ,PRCNTLIM=10]
[,DEBUG]
[,ID=trapid]
[,OK]

,END

```

Example: The following SLIP command modifies a SLIP trap.

```

SLIP MOD{,ENABLE | ,DISABLE}
        {,ALL | ,ID=trapid}

```

Example: The following SLIP command deletes one or all SLIP traps.

```

SLIP DEL{,ALL | ,ID=trapid}

```

START Command (S)

Purpose: Use the START command to start started tasks, which support system functions such as IMS, CICS, and RACF. Started tasks are defined in cataloged procedures (residing in procedure libraries) or in jobs residing in a partitioned data set defined in master JCL.

Notes:

- For any variation of the START command, if the subsystem that processes the task is JES2 or JES3 and you omit *devicetype* (or *devnum*), or *classes*, or *volumeserial*, you must supply a comma for each one of these parameters that you leave out. Do not enter any commas, however, after the last parameter that you specify.
- *devicetype* and *devnum* are mutually exclusive. You cannot specify both parameters on a START command.
- JOBNAME and *identifier* are mutually exclusive. You cannot specify both parameters on a START command.
- If you specify an option in apostrophes, use uppercase.
- If you are overriding a data set name that is 44 characters long, use DSN=, not DSNAME=.
- If you are overriding a symbolic parameter, do not use any DD keywords.

Example: The following START command starts a system task from the console.

```
S membername[.identifier][,devicetype|[/]devnum][,volumeserial]
  [,parameters][,JOBNAME=jobname][,JOBACCT=acct_info]
  [,SUB=subsystemname][,keyword=option[,keyword=option]...]
```

Example: The following START command starts the advanced program-to-program communication/MVS (APPC/MVS) address space.

```
S APPC,SUB=MSTR[,APPC=(nn[,nn]...[,L])]
```

Example: The following START command starts the APPC/MVS transaction scheduler (ASCH) address space.

```
S ASCH,SUB=MSTR[,ASCH=(nn[,nn]...[,L])]
```

Example: The following START command starts the generalized trace facility (GTF).

```
S {GTF|membername}[.identifier][,devicetype|[/]devnum][,volumeserial]
  ([,MODE={INT|DEFER|EXT}][,TIME=YES][,DEBUG=YES])
  [,BLOK={numpages|nnk|nnM}][,MEMBER=xxxxxxx][,REGION=nnnnk]
  [, {SADMP|SA}={nnM|nnk}][, {SDUMP|SD}={nnM|nnk}]
  [, {NOPROMPT|NP}][, {ABDUMP|AB}={nnM|nnk}]
  [,keyword=option[,keyword=option]...]
```

Example: The following START command starts the library lookaside (LLA) address space.

START Command

```
S LLA[,SUB=MSTR][,LLA=xx]
```

Example: The following START command starts the object access method.

```
S {OAM[membername]}.identifier[,OAM=xx]
```

Example: The following START command starts resource recovery services.

```
S RRS|membername[,CTMEM=CTnRRSxx][,GNAME=|grpname][,JOBNAME=jobname]
```

Example: The following START command starts TSO/VTAM time-sharing.

```
S membername|.identifier[,devicetype|,[/]devnum]
  [,volumeserial][([,MEMBER={name|nn})[,USERMAX=nnnnn]
  [,GNAME=[name|NONE]])
  [,keyword=option[,keyword=option]...]
```

Example: The following START command starts the virtual lookaside facility (VLF) or the data lookaside facility (DLF).

```
S {VLF|DLF},SUB=MSTR[,NN=xx]
```

Example: The following START command starts an external writer.

```
S {XWTR|membername}.identifier[,devicetype|,[/]devnum]
  [,volumeserial][,classes][,keyword=option[,keyword=option]...]
```

STOP Command (P)

Purpose: Use the STOP command to stop system functions and jobs in execution.

Example: The following STOP command stops a started task.

```
P [jobname.]identifier[,A=asid]
```

STOP Command

Example: The following STOP command stops the APPC/MVS transaction scheduler (ASCH) address space.

```
P {ASCHINT,A=asid}
```

Example: The following STOP command stops the data lookaside facility (DLF) address space.

```
P DLF
```

Example: The following STOP command stops the library lookaside facility (LLA) address space.

```
P LLA
```

Example: The following STOP command stops the object access method (OAM).

```
P [OAM.|jobname.]identifier
```

Example: The following STOP command stops the virtual lookaside facility (VLF).

```
P VLF
```

STOPMN Command (PM)

Purpose: Use the STOPMN command to stop displays initiated in response to the MONITOR command or the MONITOR parameters on the CONSOLE and INIT statements in the CONSOLxx member of SYS1.PARMLIB.

```
PM {JOBNAMES[,L={a|cc|cca|name|name-a}]}  
   {DSNAME                               }  
   {SPACE                                 }  
   {STATUS[,L={a|cc|cca|name|name-a}]} }  
   {SESS[,L={a|cc|cca|name|name-a}]} }
```

STOPTR Command (PT)

Purpose: Use the STOPTR command to stop or reduce displays resulting from the TRACK command.


```
PT {TS }[,L={a  }]
   {JOBS}      {cc  }
   {J  }      {cCa }
   {A  }      {name }
                   {name-a}
```

SWAP Command (G)

Purpose: Use the SWAP command to initiate an operator request for dynamic device reconfiguration (DDR) or to activate or deactivate system-initiated DDR.

Example: The following SWAP command initiates an operator request for DDR.

```
G [/]devnum1,[/]devnum2
```

Example: The following SWAP command activates or deactivates system-initiated DDR.

```
G {OFF}
   {ON }
```

SWITCH Command (I)

Purpose: Use the SWITCH command to switch the recording of SMF data from one data set to another and to switch certain console attributes from one console to another.

```
I {SMF                }
   {                  }
   {CN={{(consname1)  }}
   {  {{(consname1,consname2)}}
   {  {{(consname1,consname1)}}}
```

TRACE Command

Purpose: Use the TRACE command to start, stop, modify, or display the status of a system-, master-, or component-trace. Specify component trace operands by using the appropriate form of the REPLY command.

TRACE Command

```
TRACE [STATUS ]
      [ST[,nnnK][,BR={ON|OFF}] ]
      [ST[,OFF] ]
      [MT[,nnnK|,OFF] ]
      [CT{,WTRSTART=membername[,WRAP|NOWRAP] } ]
      [CT{,WTRSTOP=jobname} ]
      [CT{[,ON ],COMP=name[,SUB=(sub)][,PARM=mem]} ]
      [ ,nnnK ]
      [ ,nnnM ]
      [ ,OFF ]
      [TT{[,COLL|C=collection name } ]
        {,CON=connection type }
        {,COR=correlation info }
        {,LU=logical unit name }
        {,LVL=level }
        {,NET=netid }
        {,PKG=package name }
        {,PLAN=|PL=plan name }
        {,PRF=perform }
        {,PROC|PR=proc name }
        {,PRS=process }
        {,SUB=subsystem }
        {,TC=transaction class }
        {,TRAN|T=transaction name}
        {,USER|U=userid }
      [,WTR=membername|STOP ]
      [,LATENT=Y|N ]
      [,BUFSIZ=nnnK|nnM] ]
      [,OFF={nn|ALL} ]
```

TRACK Command (TR)

Purpose: Use the TRACK command to request a periodic display of job information on display consoles.

```
TR {TS|JOBS|J|A}[,LIST|,L][,USERID=userid][,L={a } ]
                                     {cc }
                                     {cca }
                                     {name }
                                     {name-a}
```

Note: To stop the TRACK display, use the STOPTR command.

UNLOAD Command (U)

Purpose: Use the UNLOAD command to unload mounted tape or DASD volumes, where *devnum* is the device number of the I/O device containing the volume to unload.

```
U [/]devnum
```

VARY Command (V)

Purpose: Use the VARY command to control certain system components such as the online or offline status of consoles and I/O devices.

Example: The following VARY command places the system console in problem determination mode.

```
VARY CN(conspec1|*),{ACTIVATE|ACT}
```

Example: The following VARY command removes the system console from problem determination mode.

```
VARY CN(conspec1|*),{DEACTIVATE|DEACT}
```

Example: The following VARY command changes the indicated authority for the console id(s) or name(s) specified.

```
V CN{(*|conspec1[,conspec1]... )}
    [,ALTGRP={name|{*NONE*}}]
    [,AMSCOPE=([*][,name[,name]...])]
    [,AUTH={ALL|INFO|MASTER|([SYS][,IO][,CONS])}]
    [,AROUT=(rtcode[,rtcode]...)]
    [,DMSCOPE=([*][,name[,name]...])]
    [,DROUT=(rtcode[,rtcode]...)]
    [,MSCOPE=(*ALL)|{([*][,name[,name]...)}]}
    [,OFFLINE|,ONLINE[,SYSTEM=sysname][,FORCE]]
    [,ROUT={ALL|NONE|(rtcode[,rtcode]...)}]}
    [,UD={Y|N} ]
```

Example: The following VARY command places a secondary console online or offline.

```
V (conspec[,conspec]...),{OFFLINE}
    {ONLINE }
```

Example: The following VARY command assigns and controls multiple console support (MCS) consoles.

```
V {conspec2|(conspec2[,conspec2]...)},CONSOLE
    [,ALTCONS=conspec2]
    [,AROUT=(rtcode[,rtcode]...)]
    [,AUTH={ALL|MASTER|INFO|([SYS][,IO][,CONS])}]
    [,DROUT=(rtcode[,rtcode]...)]
    [,ROUT={ALL|NONE|(rtcode[,rtcode]...)}]}
```

Example: The following VARY command switches the master console.

VARY Command

```
V {[/]devnum},MSTCONS
  {conname }
```

Example: The following VARY command controls the hardcopy message set and the hardcopy medium.

```
V [[/]devnum],HARDCPY[,CMDS|,NOCMDS|,STCMDS|,INCMDS]
  conname          [,AROUT=(rtcode[,rtcode]...)]
  SYSLOG           [,DROUT=(rtcode[,rtcode]...)]
  OPERLOG          [,ROUT={ALL|NONE }
                  {(rtcode[,rtcode]...)}]
                  [,UD={Y|N} ]
                  [,OFF]
```

Example: The following VARY command turns on or off the AUTOSWITCH attribute of a tape device.

```
V {(devspec[,devspec]...)},{[AUTOSWITCH|AS][,ON|OFF]}
```

Example: The following VARY command places an I/O device or range of devices online or offline.

```
V {(devspec[,devspec]...)},{ONLINE[,UNCOND][,FORCE]}
  {devspec } { ,SHR }
              { ,RESET }
              { }
              {OFFLINE[,FORCE]} }
```

Example: The following VARY command controls a global resource serialization (GRS) ring complex.

```
V GRS{({sysname|*|ALL}),{RESTART|R}}
  { }
  {({sysname|*}),{QUIESCE|Q} }
  { }
  {(sysname),{PURGE|P} }
```

Example: The following VARY command places an I/O path or paths online or offline.

```
V PATH
  {[/]devnum,chp[,[/]devnum,chp]...}
  {
  {([/]devnum[,[/]devnum]...),chp
  {  [,([/]devnum[,[/]devnum]...),chp]...}
  }
  {([/]lowdevnum-[/]highdevnum[,[/]lowdevnum-[/]highdevnum]...),chp
  {  [,([/]lowdevnum-[/]highdevnum[,chp]...)}
  }
  {(cfname,chp[,cfname,chp]...)}
  {
  {((cfname[,cfname]...),chp
  {  [, (cfname[,cfname]...),chp]...)}
  }
  {,ONLINE[,FORCE] }
  {,OFFLINE[,UNCOND | ,FORCE]}
```

Example: The following VARY command changes the state of coupling facility cache structures and volumes over a sysplex wide scope.

```
V SMS,{CFCACHE(cachename),{ENABLE|E }
  { {QUIESCE|Q}
  {CFVOL(volid),{ENABLE|E }
  { {QUIESCE|Q}
  {MONDS(dsname[,dsname...]),{ON|OFF}
  {SHCDS(shcdsname),{NEW }
  { {NEWSPARE}
  { {DELETE }
  {SMSVSAM,{ACTIVE
  { {SPHERE(spherename),{ENABLE|E}}
  { {FALLBACK
  { {TERMINATESERVER
  { {FORCEDELETELOCKSTRUCTURE }
```

Example: The following VARY command places an optical drive or optical library online or offline.

```
V SMS,{DRIVE|DRI|LIBRARY|LIB}(name),{ONLINE|ON }
  {OFFLINE|OFF}
```

Example: The following VARY command places a system-managed tape library online or offline.

```
V SMS,{LIBRARY}(name[, {systemid[,systemid]...}]),{ONLINE|ON }
  {LIB } {ALL } {OFFLINE|OFF}
  {*_ }
```

Example: The following VARY command changes the SMS status of a storage group.

VARY Command

```
V SMS,{{STORGRP|SG}(storgrp,[*|ALL|system[,system]...)}}, {QUIESCE|Q} [,NEW|,N]
      {
      {VOLUME|VOL}(volume,[*|ALL|system[,system]...)} } {ENABLE|E }
      {DISABLE|D} [,NEW|,N]
```

Example: The following VARY command controls an application environment.

```
V WLM,APPLENV=applenvname, {REFRESH }
                          {QUIESCE|Q}
                          {RESUME }
```

Example: The following VARY command activates a workload management service policy for a sysplex.

```
V WLM,POLICY=policyname[,REFRESH]
```

Example: The following VARY command causes the cross-system coupling facility (XCF) to remove the specified system from the sysplex.

```
V XCF{,systemname},{OFFLINE|OFF} [,RETAIN={YES|NO}] [,FORCE]
```

WRITELOG Command (W)

Purpose: Use the WRITELOG command to control the system log.

```
W [class|CLOSE|START]
```

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

NOTICES

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States and/or other countries:

BookManager	IBM
BookMaster	OS/390

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Readers' Comments — We'd Like to Hear from You

OS/390
MVS System Commands Summary

Publication No. GX22-0040-10

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

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

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY
12601-5400



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5647-A01



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GX22-0040-10

