

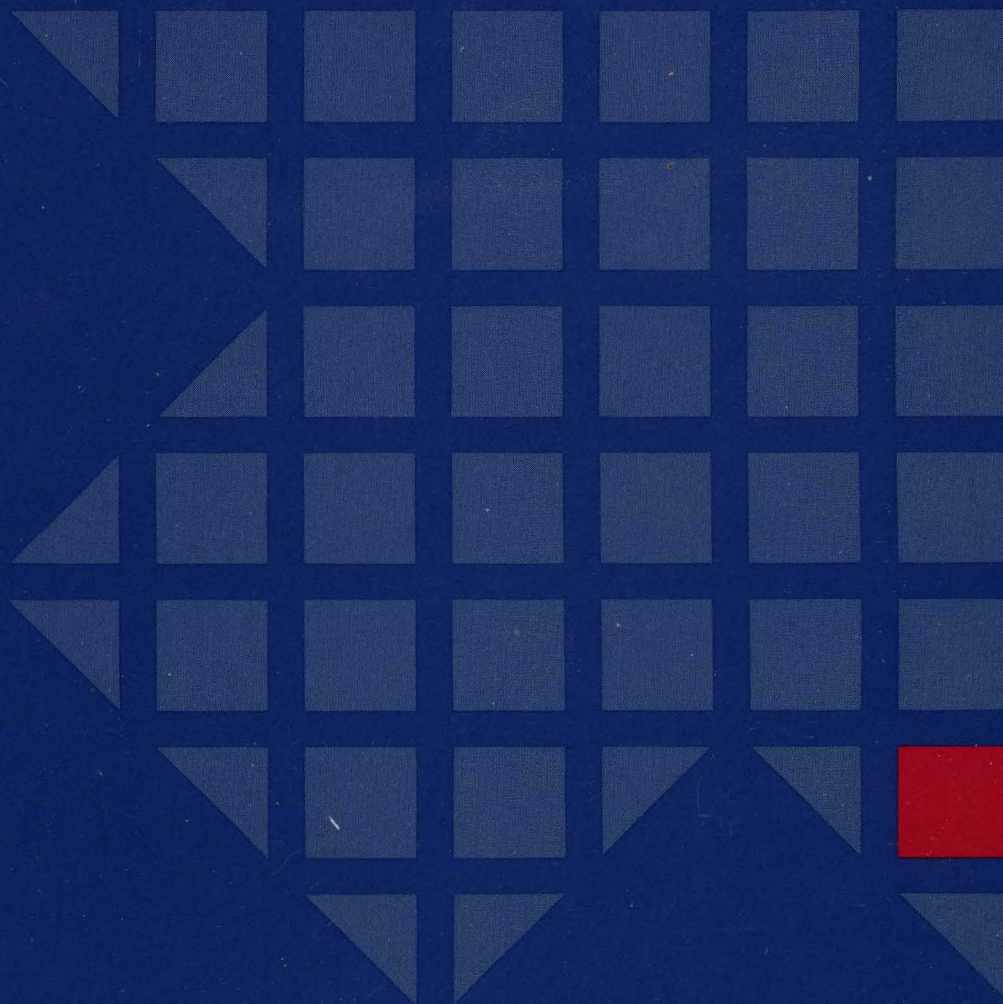


Virtual Machine/Extended Architecture™
System Product

SC23-0358-1

CP Command Reference

VM/XA™ SP Release 2





Virtual Machine/Extended Architecture™
System Product

SC23-0358-1

CP Command Reference

VM/XA™ SP Release 2

Second Edition (November 1988)

This is a major revision of, and obsoletes, SC23-0358-0. Technical changes throughout this manual are indicated by vertical bars to the left of the changed material.

This edition applies to Release 2 of the Virtual Machine/Extended Architecture System Product (VM/XA SP) Licensed Program 5664-308. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370, 30xx, 4300, and 9370 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

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

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

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to International Business Machines, Department 52Q/MS 458, Neighborhood Road, Kingston, N.Y. 12401. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Virtual Machine/Extended Architecture and VM/XA are trademarks of the International Business Machines Corporation (IBM).

Preface

Purpose

This publication describes the command formats, syntax rules, operands, and options for the Virtual Machine/Extended Architecture System Product (VM/XA SP) control program (CP) commands.

Audience

This manual is for anyone who uses CP commands.

Related Publications

Please refer to the Bibliography at the back of this publication.

Contents

Chapter 1. Introduction	1
Environments	1
Using VM/XA SP CP Commands	1
VM/XA SP CP Command Structure	2
The Command Name	2
The Command Operands	3
How to Enter Comments in the CP Command Language	3
Character Sets	3
Using Truncations and Abbreviations	3
Notational Conventions	5
Privilege Classes	8
Chapter 2. VM/XA SP CP Commands	11
* (ASTERISK COMMAND)	17
#CP	18
ACNT	20
ATTACH	21
ATTN	30
AUTOLOG	31
BACKSPACE/BACKWARD	35
BEGIN	37
CHANGE	39
CHANGE: Options	41
CLOSE	48
CLOSE: Options	49
COMMANDS	58
COUPLE	60
CP	62
CPFORMAT	63
CPTRAP	70
CPU	71
DEDICATE	73
DEFINE	75
DEFINE CONSOLE	77
DEFINE CPU	78
DEFINE CHANNEL-TO-CHANNEL ADAPTER	80
DEFINE GRAF	81
DEFINE MDISK	82
DEFINE (spooling device)	83
DEFINE STORAGE	87
DEFINE (temporary disk)	89
DEFINE (virtual device)	91
DEFINE VECTOR	93
DEFSEG	95
DEFSYS	101
DETACH	105
DETACH CPU	107
DETACH (real device)	108
DETACH VIRTUAL (devices)	112
DETACH VECTOR	116
DETACH XSTORE	118
DIAL	120

DISABLE	122
DISCONNECT	124
DISPLAY	126
DISPLAY GUEST STORAGE	130
DISPLAY HOST STORAGE	135
DISPLAY PREFIX	139
DISPLAY PROGRAM STATUS WORDS	140
DISPLAY REGISTERS	141
DISPLAY SCHIB	144
DISPLAY VECTOR	146
DRAIN	150
DUMP	156
DUMP GUEST STORAGE	161
DUMP HOST STORAGE	166
DUMP PREFIX	170
DUMP PROGRAM STATUS WORDS	171
DUMP REGISTERS	173
DUMP SCHIB	176
DUMP VECTOR	178
ECHO	182
ENABLE	183
EXTERNAL	184
FLUSH	186
FORCE	188
FORWARD	190
FREE	192
FREE LOGON	193
FREE PRINTER/PUNCH	194
HALT	196
HOLD	198
HOLD LOGON	199
HOLD PRINTER/PUNCH	200
INDICATE	202
INDICATE I/O	204
INDICATE LOAD	205
INDICATE PAGING	209
INDICATE QUEUES	211
INDICATE USER	213
INDICATE VECTOR	218
IPL	219
LINK	223
LOADBUF	228
LOADVFCB	233
LOCATE	236
LOCK	238
LOGOFF	241
LOGON	244
MESSAGE	248
MESSAGE ALL	250
MESSAGE (user)	251
MONITOR	253
MONITOR EVENT	256
MONITOR SAMPLE	262
MONITOR START/STOP	267
MSGNOH	269
NOTREADY	270

ORDER	271
PER	275
PURGE	276
PURGE	278
PURGE IMG	281
PURGE NLS	283
PURGE NSS	285
PURGE TRFILES	287
PURGE UCR	289
QUERY	290
QUERY ABEND	297
QUERY ACCOUNT	298
QUERY ALLOC	299
QUERY CACHE	301
QUERY CHPID	305
QUERY CMDLIMIT	307
QUERY COMMANDS	308
QUERY CPLANGLIST	310
QUERY CPLANGUAGE	311
QUERY CPLEVEL	312
QUERY CPTRACE	313
QUERY CPTRAP	316
QUERY CPUID	317
QUERY DASD RESERVE	318
QUERY (real device)	320
QUERY DUMP	335
QUERY FILES	337
QUERY FRAMES	340
QUERY HOLD	343
QUERY IMG	345
QUERY IOASSIST	347
QUERY LINKS	348
QUERY LOGICAL	349
QUERY LOGMSG	350
QUERY MAXSPOOL	351
QUERY MAXUSERS	352
QUERY MITIME	353
QUERY MONDATA	355
QUERY MONITOR	356
QUERY NAMES	364
QUERY NLS	366
QUERY NSS	368
QUERY PASSWORD	375
QUERY PATHS	377
QUERY PER	378
QUERY PF _{nn}	379
QUERY PROCESSORS	381
QUERY QUICKDSP	382
QUERY READER/PRINTER/PUNCH	383
QUERY RECORDING	390
QUERY SCREEN	392
QUERY SDF	393
QUERY SECUSER	394
QUERY SET	396
QUERY SHARE	401
QUERY SRM	402

QUERY (real) STORAGE	404
QUERY SYSTEM	405
QUERY TAG	407
QUERY TDISK	409
QUERY TERMINAL	410
QUERY TIME	413
QUERY TRACE	414
QUERY TRFILES	417
QUERY TRSAVE	420
QUERY TRSOURCE	423
QUERY UCR	428
QUERY USERID	430
QUERY USERS	431
QUERY VECTOR	433
QUERY VIRTUAL ALL	435
QUERY VIRTUAL CONSOLE	436
QUERY VIRTUAL CPUS	440
QUERY VIRTUAL CTCA	442
QUERY VIRTUAL DASD	444
QUERY VIRTUAL DEVICE	447
QUERY VIRTUAL GRAF	449
QUERY VIRTUAL PRINTER	451
QUERY VIRTUAL PUNCH	455
QUERY VIRTUAL READER	458
QUERY VIRTUAL STORAGE	460
QUERY VIRTUAL TAPES	461
QUERY VIRTUAL UR	462
QUERY VIRTUAL VECTOR	463
QUERY VIRTUAL XSTORE	464
QUERY V=R	465
QUERY XSTORE	467
READY	471
RECORDING	472
REDEFINE	474
REPEAT	476
REQUEST	478
RESET	479
RETAIN XSTORE	481
REWIND	484
SAVESEG	485
SAVESYS	487
SCREEN	489
SEND	491
SET	493
SET ABEND	498
SET ACCOUNT	499
SET AUTOPOLL	500
SET CACHE	501
SET CCWTRAN	503
SET CMDLIMIT	504
SET CONCEAL	505
SET CPCONIO	507
SET CPLANGUAGE	508
SET CPTRACE	509
SET CPUID	518
SET DUMP	519

SET EMSG	522
SET IMSG	523
SET IOASSIST	524
SET LINEDIT	525
SET LOGMSG	526
SET MACHINE	528
SET MAXUSERS	529
SET MIH	530
SET MITIME	531
SET (error recording) MODE	534
SET MONDATA	535
SET MSG	536
SET NOPDATA	537
SET NOTRANS	538
SET PAGEX	539
SET PASSWORD	540
SET PFnn	542
SET PFnn COPY	545
SET PFnn RETRIEVE	547
SET PFnn TAB	549
SET QUICKDSP	550
SET RECORD	552
SET RESERVED	553
SET RUN	554
SET SHARE	555
SET SHARED	557
SET SMSG	558
SET SRM	559
SET SVC76	564
SET TIMER	565
SET VMCONIO	566
SET WNG	567
SHUTDOWN	568
SLEEP	569
SMSG	571
SPACE	572
SPOOL	574
SPOOL READER/PRINTER/PUNCH/CONSOLE	577
SPOOLING: Options	579
SPTAPE	586
SPTAPE CANCEL	594
SPTAPE DUMP	595
SPTAPE LOAD	598
SPTAPE SCAN	601
SPTAPE STOP	602
START	603
START: Options	604
STOP	612
STORE	613
STORE GUEST STORAGE	615
STORE HOST STORAGE	617
STORE PSW/CAW/CSW	619
STORE REGISTERS	620
STORE STATUS	621
STORE VECTOR	622
SYSTEM	625

TAG	627
TERMINAL	630
TERMINAL APL	633
TERMINAL ASCIITBL	634
TERMINAL ATTN	635
TERMINAL BREAKIN	636
TERMINAL BRKKEY	637
TERMINAL CHARDEL	639
TERMINAL CONMODE	640
TERMINAL CNTL	642
TERMINAL ESCAPE	643
TERMINAL HILIGHT	644
TERMINAL LINEDEL	645
TERMINAL LINEND	646
TERMINAL LINESIZE	647
TERMINAL MODE	648
TERMINAL PROMPT	649
TERMINAL SCROLL	650
TERMINAL TABCHAR	651
TERMINAL TEXT	652
TERMINAL TYPE	653
TRACE	654
TRACE: Common Options	659
TRACE: Command Responses	663
TRACE ALL	672
TRACE APPEND	673
TRACE BRANCH	674
TRACE CLEAR	676
TRACE COUNT	677
TRACE END	678
TRACE EXTERNAL/PROGRAM	679
TRACE GOTO/CALL/RETURN	680
TRACE GPRs	682
TRACE INSTRUCTION	684
TRACE I/O	686
TRACE MCH	688
TRACE mnemonic1	689
TRACE mnemonic2	691
TRACE RIO/RSCH/SIO/SIOF/SSCH	693
TRACE STORE	695
TRACE SVC/DIAGNOSE/MC	697
TRACE TABLE	699
TRACE TPI	700
TRACE TRAP	702
TRANSFER	704
TRSAVE	710
TRSOURCE	715
TRSOURCE DISABLE	718
TRSOURCE DISPLAY	719
TRSOURCE DROP	720
TRSOURCE ENABLE	721
TRSOURCE ID (Definition Operands)	722
UNDEDICATE	731
UNLOCK	732
VARY	734
VARY PATH	737

VARY VECTOR	743
VMDUMP	745
WARNING	749
XAUTOLOG	751
Appendix A. Restricted Userids	757
Summary of Changes	759
Second Edition	759
Glossary	763
Bibliography	769
VM/XA System Product Microfiche	769
VM/XA System Product Publications	769
Index	773

Chapter 1. Introduction

The Virtual Machine/Extended Architecture™ System Product (VM/XA™ SP) is a flexible operating system that will help your installation to gain the benefits of the IBM System/370 Extended Architecture (370-XA). This means the removal of storage constraints and an enhancement to your use of IBM's largest 370-XA processors. With VM/XA SP, you also get a high-capacity interactive environment—CMS—that supports System/370 and 370-XA programs.

Environments

VM/XA SP uses two command languages which correspond to the two environments it creates. You should:

- Use the control program (CP) command language if:
 - You are a VM/XA SP system operator and you want to control the resources of the real machine located in your computer room.
 - You are a virtual machine user and you want to control your virtual machine's configuration and environment.
- Use a virtual machine command language if:
 - You are communicating with the operating system you loaded into your virtual machine.

To perform production or test work, load your virtual machine with one of the operating systems supported by the VM/XA SP system. Your virtual machine command language is the command language of the operating system you load. This command language is described in the library that documents that particular operating system.

To perform service, installation, and maintenance tasks, along with editing and text creation, communicating with others, and problem solving, load your virtual machine with the conversational monitor system (CMS). CMS is a single user, conversational operating system. It runs under CP in a System/370 mode or a 370-XA virtual machine. The *VM/XA SP CMS Command Reference* describes the CMS commands used to install and to service VM/XA SP.

Using VM/XA SP CP Commands

You can use CP commands whenever:

- Your virtual machine is in the control program (CP) command environment. Your virtual machine is in the CP environment when you log on to VM/XA SP and "CP READ" is displayed in the lower right-hand corner of the screen.

- You press the break key while in full-screen mode before entering a command. (The break key may be PA1, the VM/XA SP default break key, or another key that you have defined as the break key using the `TERMINAL BRKKEY` command. Also, the break key may be totally disabled by some application programs or when in the protected application environment.)
- You are in a virtual machine command environment, not running in full-screen mode, and issue the `#CP` command (and `#` is your logical linend character).
- You are in the CMS virtual machine environment and issue the `CP` command.
- You are in the CMS virtual machine environment and have the `IMPCP` function set `ON`.

To determine your current command environment, look at the status area in the lower right-hand corner of your display screen. “`CP READ`” indicates the `CP` environment and “`VM READ`” indicates the virtual machine environment. If you’re running CMS in your virtual machine, “`VM READ`” indicates the CMS environment. When “`RUNNING`” appears in the status area, enter a null input line to determine your environment. (To enter a null line, press the enter key but do not enter any data.) When you enter the null line, the status area displays either “`CP READ`” or “`VM READ`.” Also, if you’re in a read state in either the `CP` command environment (with `RUN` set `OFF`) or the CMS command environment and you enter a null line, the system responds with the name of your command environment (“`CP`” or “`CMS`”) in the system output area.

You enter `CP` commands using any combination of uppercase and lowercase letters. When you have typed the command and its operands, press the enter key to execute the command.

You can also issue `CP` commands from CMS EXEC procedures. *VM/XA SP Real System Operation* and the *VM/XA SP System Product Interpreter User’s Guide* contain information about issuing `CP` commands from CMS EXEC procedures.

VM/XA SP CP Command Structure

A VM/XA SP command line consists of a command name, usually followed by one or more positional operands. The general form for the `CP` command line is:

command name	operands...
--------------	-------------

You must use one or more blanks to separate each entry in the command line unless otherwise indicated.

The Command Name

The command name is an alphanumeric symbol, not more than twelve characters long. For example, you can enter:

`DISCONNECT`

as a command.

In general, command names are verbs that describe the function that you want the system to perform. For example, if you want to attach a real device to your virtual machine configuration, you would issue the `ATTACH` command.

The Command Operands

Command operands are keywords and positional symbols, usually no longer than eight alphanumeric characters.

You use operands to specify the information on which the system will operate when it performs the command function. For example, you use the `TERMINAL` operand in conjunction with the `CP QUERY` command to find out what terminal options are in effect for your virtual machine.

Some commands require no operands; others require several. You can find a description of each command with all of its operand requirements in Chapter 2, "VM/XA SP CP Commands."

Unless otherwise indicated, you must enter the operands that you select in the order in which they appear in the command formats in Chapter 2, "VM/XA SP CP Commands."

How to Enter Comments in the CP Command Language

You can use the `CP *` command to annotate your display screen and console spool file. Within this console spool file will be your input to the system and the system's responses back to you. Refer to the description of the `CP *` command in Chapter 2, "VM/XA SP CP Commands" for complete details.

Character Sets

You can enter VM/XA SP CP commands using a combination of characters from six different character sets. The contents of each of these sets is summarized in Table 1.

Character Set	Names	Symbols
Separator	Blank	
National	Dollar Sign Pound Sign At Sign	\$ # @
Alphabetic	Uppercase Lowercase	A - Z a - z
Numeric	Numeric	0 - 9
Alphanumeric	National Alphabetic Numeric	\$, #, @ A - Z, a - z 0 - 9
Special		All other characters

Using Truncations and Abbreviations

To make entering commands on the keyboard more convenient, VM/XA SP allows you to enter many commands and operands in a shortened form. This shortened form can be either a truncation or an abbreviation.

You truncate a command or operand name by dropping one or more letters from the end of the name. The format box for each command in Chapter 2, “VM/XA SP CP Commands” shows the truncations that you can use for each command and its operands. If a letter appears in uppercase in the format box, you must enter it when you issue the command from your display; if a letter is in lowercase, you may, if you wish, omit it when you issue the command.

For example, the format box for the QUERY command shows the command as:

Query

This means that you can enter the QUERY command in any of the following forms:

query
quer
que
qu
q

The minimum acceptable truncation is Q, but CP will accept any of the above forms of the command.

Abbreviations are also shorter forms of commands and operands, but unlike truncations, are not formed by simply dropping letters from the name of the command.

The format boxes in Chapter 2, “VM/XA SP CP Commands” show acceptable abbreviations for command names below the full name of the command. Abbreviations for operands are listed in the description of the operands following the format box.

For example, the format box for the MESSAGE command shows:

Message
Msg

This means that you can:

- Truncate the command to a minimum of “M”
- Use the abbreviation “MSG”
- Truncate the abbreviation to “MS” or “M”.

Thus, CP will accept all of the following forms of the MESSAGE command:

message
messag
messa OR msg
mess ms
mes m
me
m

Notational Conventions

The command format descriptions in Chapter 2, “VM/XA SP CP Commands” use the following notation to define command syntax:

- **Truncations and Abbreviations of Commands**

Where truncation of a command name or operand is permitted, the shortest acceptable version of the command is indicated in uppercase letters. (Of course, you can enter the command from your display using any combination of uppercase and lowercase letters.) If no truncation is permitted (that is, the entire command name or option is in uppercase letters), you must enter the entire command name or operand.

Abbreviations are shown below the full command name in the format box, and abbreviations for operands are shown in the description of the operands that follows the format box.

- Do not enter the following symbols when you are entering CP commands. These symbols are used to define command formats and are not part of the CP commands they describe:

Character	Symbol	Definition Usage
Underscore	_	Default
Braces	{ }	Must select one operand
Brackets	[]	Optionally select operand
Ellipsis	...	Repeat operand

- Always enter uppercase letters and words, and the following symbols when specified in the format box:

Character	Symbol
Asterisk	*
Comma	,
Hyphen	-
Equal sign	=
Parenthesis	()
Period	.
Colon	:

- Lowercase letters, words, and symbols serve two purposes in the command format box.

Lowercase letters in a command or operand name indicate that you do not have to enter that letter when entering the command or operand from your display. (See “Using Truncations and Abbreviations” above for more information.)

Other lowercase words and symbols that appear in the format box represent variable information. You must supply the specific information that VM/XA SP system will need to perform the command function. For example, “hexloc”

indicates that the system needs a hexadecimal storage location in order to process the command. You would substitute the appropriate value when you enter the command.

Table 2 includes “variable” names that are commonly used in Chapter 2, “VM/XA SP CP Commands” to express command syntax:

Table 2 (Page 1 of 2). Variable Names Commonly Used to Express Command Syntax	
Variable	Definition
cpuaddr	A virtual processor address, in hexadecimal.
hexloc	A one- to eight-digit hexadecimal number that represents a location in storage. If your virtual machine is operating in System/370 mode, storage locations can range from 0 through FFFFFFFF. For 370-XA mode virtual machines, the maximum value is 7FFFFFFF. Of course, when you substitute a value for hexloc, it can never be larger than the size of the real or virtual machine storage that it refers to.
ldev	An “L” followed by a one- to four-digit hexadecimal number that represents a logical device number. Note that L1, L01, L001, and L0001 all represent the same logical device number.
ldev-ldev	A range of logical device numbers. The second logical device number must be greater than or equal to the first. For example, L2A0-L2A2 refers to the logical device numbers L2A0, L2A1, and L2A2.
luname	A 1- to 8-character logical unit name that identifies an SNA/CCS terminal or printer. It is the name from the VTAM definition of the device as an SNA logical unit.
rdev	A one- to four-digit hexadecimal number that represents a real device number. This is the hardware configured physical device number.
rdev-rdev	A range of real device numbers. The second device number must be greater than or equal to the first. For example, 2A0-2A2 refers to real device numbers 2A0, 2A1, and 2A2.
spoolid	The spool file identification number that CP assigns to each spool file created on the system.
userid	The one- to eight-character name that identifies the user to the VM/XA SP system. This is essentially the name by which the system knows you.

Table 2 (Page 2 of 2). Variable Names Commonly Used to Express Command Syntax	
Variable	Definition
vdev	A one- to four-digit hexadecimal number that represents a virtual device number.
vdev-vdev	A range of virtual device numbers. The second device number must be greater than or equal to the first. For example, 1D0-1D2 refers to virtual device numbers 1D0, 1D1, and 1D2.

- Choices are represented in the command format boxes by stacking.

For example, the representation:

A
B
C

indicates that you have three choices: A, B, or C.

- An underscore indicates a default. If you select an underscored choice, you need not specify it when you enter the command.

For example, the representation:

A
B
C

indicates that you may select A, or B, or C. If you select B, you need not specify it when you enter the command. If you do not enter A, or B, or C, the system will assume that B was your choice.

- When a list of choices is enclosed by braces, you must select one of the choices when you enter the command.

For example, the representation:

{
A
B
C
}

indicates that you must specify one of A, or B, or C.

- When a list of choices is enclosed by brackets, you may select one of the choices when you enter the command; your selection is optional.

For example, the representation:

[
A
B
C
]

indicates that you may enter A, or B, or C, or you may omit the field.

- An ellipsis (...) indicates that you can repeat the preceding item or group of items more than once in succession.

For example, the representation:

```
{spoolid1 ... }
```

indicates that you can enter more than one spoolid.

Privilege Classes

The set of CP commands that you can use depends upon the privilege class or classes that have been assigned to you.

VM/XA SP CP commands are divided into eight groups, each represented by a privilege class. The privilege class indicates the type of user from whom the commands will be accepted.

In general, the system programmer who creates your system directory will assign you one or more privilege classes as part of your entry in the directory.

Privilege classes are denoted by the letters A through Z, the numbers 1 through 6, or the word "Any." These classes, and the type of user who can use the commands belonging to the privilege class set, are summarized in Table 3. Classes I through Z and numbers 1 through 6 are reserved for redefinition through user class restructure (UCR) by each installation for its own use.

Users who have been assigned the password NOLOG have no privilege class and can only receive spooled output as punched cards or printed forms. For more information about NOLOG users, see *VM/XA SP Installation and Service*.

Class	User and Function
A	<p><i>System Operator:</i> The class A user controls the VM/XA SP system. The system operator is responsible for the availability of the VM/XA SP system and its resources. In addition, the system operator controls system accounting, broadcast messages, virtual machine performance options, and other options that affect the overall performance of VM/XA SP.</p> <p>Note: The class A user who is automatically logged on during CP initialization is designated as the primary system operator.</p>
B	<p><i>System Resource Operator:</i> The class B user controls all the real resources of the VM/XA SP system, except those controlled by the system operator and the spooling operator.</p>
C	<p><i>System Programmer:</i> The class C user updates certain functions of the VM/XA SP system.</p>
D	<p><i>Spooling Operator:</i> The class D user controls spool files and the system's real reader, printer, and punch equipment allocated to spooling use.</p>

Table 3 (Page 2 of 2). Privilege Classes	
Class	User and Function
E	<i>System Analyst:</i> The class E user examines and saves system operation data in specified VM/XA SP storage areas.
F	<i>Service Representative:</i> The class F user obtains, and examines in detail, data about input and output devices connected to the VM/XA SP system.
G	<i>General User:</i> The class G user controls functions associated with a particular virtual machine.
Any	Commands belonging to class "Any" are available to any user, regardless of his privilege class. These commands are primarily those used to gain access to or relinquish access from the VM/XA SP system.
H	Reserved for IBM use.
I - Z 1 - 6	Classes I through Z are reserved for redefinition through user class restructure (UCR) by each installation for its own use.

Chapter 2. VM/XA SP CP Commands

This section contains format descriptions for all of the VM/XA SP CP commands. These command formats are listed in alphabetical order except for the * and #CP commands, which appear first.

Each of the command formats is presented in the following order:

- **Command name:** identifies the name of the command. This appears on the top of the page for easy reference.
- **Privilege class:** indicates which users can issue the command, unless user class restructure (UCR) is in use.
- **Description of function:** explains how you can use the command.
- **Syntax diagram:** lists the syntax of the command with all the possible operands that you can use.
- **Operand description:** describes the function of each operand and any values that you can include in the operand.
- **Usage Notes:** describe special uses of the command and general considerations that you should be aware of before you issue the command.
- **Responses:** describe the CP command responses that you may receive on your display device. Responses are normal operation output and provide information about the execution and effect of the command. Unlike system messages, command responses are not prefixed with an identification and are not contained in the *VM/XA SP System Messages and Codes Reference*.

Note, however, that execution of commands may produce system messages for abnormal situations. Refer to the *VM/XA SP System Messages and Codes Reference* for complete information about these messages and suggested actions that you can take.

- **Migration Notes:** describe for both VM/SP HPO users and VM/XA System Facility users what specific changes they may expect to see from their systems to VM/XA SP. For more information, both general and specific, these users should refer to the *VM/XA SP Conversion Notebook*.

To help you see at a glance which commands are discussed in this book, the following table lists each command alphabetically, shows its privilege classes, and gives a brief description of what the command accomplishes.

Command Summary

Table 4 (Page 1 of 5). List of CP Commands and Their Privilege Classes		
CP Command	IBM-Defined Privilege Class	Command Description
*	Any	Display a comment on the virtual operator's console.
#CP	Any	Enter CP commands when the virtual machine command environment is in effect and/or switch from the virtual machine command environment to the CP command environment.
ACNT	A	Create accounting cards for one or more logged-on users and reset accounting data.
ATTACH	B	Allocate a real or logical device to a virtual machine. Allocate a direct access device for CP control (CP-owned volume). Allocate a partition of Expanded Storage to a virtual machine.
ATTN	G	Signal one attention interruption from a virtual operator's console.
AUTOLOG	A,B	Automatically log on a virtual machine to operate with its virtual operator's console disconnected.
BACKSPACE (BACKWARD)	D	Reposition the output of a real spooling device.
BEGIN	G	Resume execution of the virtual machine at a specific virtual storage location or at the address in the current PSW on all or selected virtual CPUs.
CHANGE	D,G	Alter one or more external attributes of a closed spool file. Alter the user hold status of an active output spool file.
CLOSE	G	Terminate spooling operations on a virtual card reader, card punch, printer, or console.
COMMANDS	Any	Display the commands and diagnose codes which you are authorized to use.
COUPLE	G	Connect two virtual machines via virtual channel-to-channel adapter.
CP	Any	Treated as a null command if entered when the CP command environment is in effect in a virtual machine.
CPFORMAT	G	Format, allocate, and/or label a direct access storage volume for CP use.
CPTRAP	N.A.	VM/XA SP does not support CPTRAP. See the TRSOURCE, TRSAVE, and SET CPTRACE commands.
CPU	G	Select the virtual CPU on which CP commands will be executed or on which console function reads will be issued.

Table 4 (Page 2 of 5). List of CP Commands and Their Privilege Classes		
CP Command	IBM-Defined Privilege Class	Command Description
DEDICATE	A	Dedicate a real processor to a virtual CPU belonging to the virtual=real virtual machine.
DEFINE	A, G	Reconfigure a virtual machine. Define a minidisk on a system volume for recovery purposes.
DEFSEG	E	Define a skeleton system data file for a saved segment.
DEFSYS	E	Define a skeleton system data file for a named saved system.
DETACH	B,G	Remove a real or logical device from a virtual machine. Remove a direct access storage device from CP. Remove Expanded Storage from a virtual machine. Remove a virtual CPU, virtual Vector Facility, or virtual device from a virtual machine.
DIAL	Any	Connect a display to a virtual communication line of a multiple-access virtual machine.
DISABLE	A,B,F	Prevent displays and VTAM service machines from accessing VM/XA SP.
DISCONNECT	Any	Disconnect the virtual operator's console from a virtual machine.
DISPLAY	C,E,G	Display real storage on the virtual operator's console. Display the contents of specified virtual storage areas, certain registers, and/or certain control words on the virtual operator's console.
DRAIN	D	Halt operations on specified real spooling devices after processing of the current spool file is complete.
DUMP	C,E,G	Dump real storage to a spooled virtual printer. Dump to a spooled virtual printer the contents of specified virtual storage areas, certain registers, and/or certain control words on the virtual operator's console.
ECHO	G	Test display hardware.
ENABLE	A,B,F	Allow displays and VTAM service machines to access VM/XA SP.
EXTERNAL	G	Simulate an external interruption for a virtual machine and return control to that virtual machine.
FLUSH	D	Immediately terminate transcription by a specified real unit record device of the current output spool file.
FORCE	A	Cause the logoff of a specified user.
FORWARD	D	Reposition the current spool file on a real printer or card punch.
FREE	B,D	Remove spool files belonging to a specified user from system hold status. Remove a specified user from logon hold status.
HALT	A	Terminate the active channel program on the specified real device.

Table 4 (Page 3 of 5). List of CP Commands and Their Privilege Classes		
CP Command	IBM-Defined Privilege Class	Command Description
HOLD	B,D	Place the spool files of a specified user in system hold status. Prevent a specified user from logging on.
INDICATE	B,C,E,G	Display on the virtual operator's console the use of and contention for the resources of the real machine.
IPL	G	Simulate initial program load for a virtual machine.
LINK	G	Provide access for a virtual machine to a minidisk that is owned by another virtual machine.
LOADBUF	D	Load a real UCS or FCB buffer with a specified image.
LOADVFCB	G	Establish a virtual forms control buffer for a spooled virtual printer.
LOCATE	C,E	Display the address of CP modules and certain CP control blocks on the virtual operator's console.
LOCK	A	Bring pages into real storage and permanently lock them.
LOGOFF LOGOUT	Any	Disable access to CP and terminate the existence of a virtual machine.
LOGON LOGIN	Any	Provide access to CP and establish the existence of a virtual machine.
MESSAGE	A,B,Any	Send messages from user to user.
MONITOR	A,E	Initiate and terminate data collection and recording for performance measurement.
MSGNOH	B	Send a message without the standard message header.
NOTREADY	G	Simulate "not ready" for the specified virtual spooling device.
ORDER	D,G	Rearrange closed spool files in a specific order.
PER	G	See the TRACE command.
PURGE	A,B,C,D,E,G	Remove closed spool files and system data files from the system.
QUERY	Any	Request information about the configuration or status of a virtual machine or the real machine.
READY	G	Simulate a device-end interruption for the specified virtual device.
RECORDING	A,B,C,E,F	Change the processing parameters for CP accounting and error recording facilities.
REDEFINE	G	Change the virtual device number of any existing virtual device in a virtual machine configuration.
REPEAT	D	Repeat printing or punching of the current spool output file.
REQUEST	G	Signal one attention interruption from a virtual operator's console.

Table 4 (Page 4 of 5). List of CP Commands and Their Privilege Classes

CP Command	IBM-Defined Privilege Class	Command Description
RESET	B,G	Release a real direct access storage device that holds a reserve. Clear and reset all pending interruptions for a specified virtual device and reset all error conditions.
RETAIN	B	Reserve Expanded Storage for CP use.
REWIND	G	Perform an operation equivalent to rewinding and readying a tape drive.
SAVESEG	E	Save a saved segment in a previously defined system data file.
SAVESYS	E	Save a named saved system in a previously defined system data file.
SCREEN	G	Change either the color definition, the highlighting definition, or both definitions for a virtual machine console.
SEND	G	Enter console input to a disconnected virtual machine for which you provide secondary console services.
SET	A,B,C,E,F,G	Establish system parameters. Control various functions within a virtual machine.
SHUTDOWN	A	Checkpoint the system for a warm start and terminate real machine operations.
SLEEP	Any	Place the virtual machine in a dormant state for a specified period of time.
SMSG	G	Send a special message to a virtual machine programmed to accept and process the message.
SPACE	D	Force single spacing on a real printer that is being used for spooling.
SPOOL	G	Change spooling control options or transfer spool output files to another virtual machine.
SPTAPE	D,E	Save on tape spool files and system data files and restore to the system previously saved spool files and system data files.
START	D	Start a real spooling device after it has been drained or change the processing options currently in effect for the device.
STOP	G	Place all virtual CPUs in a virtual machine in soft-stop or place selected virtual CPUs in hard-stop.
STORE	C,G	Alter specified real storage locations. Alter specified virtual storage locations, registers, and/or control words or perform the STORE STATUS function.
SYSTEM	G	Simulate for a virtual machine the following hardware functions: CLEAR STORAGE, RESET, RESTART, and STORE STATUS.
TAG	G	Assign tag data to a spooled virtual unit record output device, insert tag data in the tag field of a spool output file, or display the tag information assigned to a spooled output device or a spool output file.

Table 4 (Page 5 of 5). List of CP Commands and Their Privilege Classes

CP Command	IBM-Defined Privilege Class	Command Description
TERMINAL	G	Redefine the virtual operator's console I/O processing, message handling, and attention interruption handling.
TRACE	G	Monitor events that occur in the virtual machine, including instruction execution, storage alteration, register alteration, and I/O activity.
TRANSFER	D,G	Send spool files to or retrieve previously transferred spool files from the specified user's reader, printer, or punch queue.
TRSAVE	A,C	Control the saving of traces on tape or DASD.
TRSOURCE	C	Define and control I/O, data, and guest tracing.
UNDEDICATE	A	Remove sole use of a real processor from a virtual CPU belonging to the virtual = real virtual machine.
UNLOCK	A	Unlock previously locked pages or release the pages in the V = R or RIO370 area.
VARY	B	Mark available or unavailable for use a real device, real channel path, real processor, or real Vector Facility.
VMDUMP	G	Dump to a spooled virtual reader all or selected pages of a virtual machine's virtual storage.
WARNING	A,B,C	Send a high-priority message to a specified user or to all users.
XAUTOLOG	A,B,G	Automatically log on a virtual machine to operate with its virtual operator's console disconnected.

* (ASTERISK COMMAND)

Privilege Class: Any

Use the asterisk command (*) to annotate your console sheet or your display screen.

*	[any comment]
---	---------------

Usage Notes

1. CP treats any statement that you prefix with an asterisk as a comment line. Thus, all of the following are comments:
 - * any comment
 - *any comment
 - *** any comment
2. If you have invoked the console spooling function for your virtual machine, the commentary that you write will also appear in your console spool file.

Responses

None.

Migration Notes

VM/SP HPO: VM/XA SP does not require a blank after the * command.

#CP

Privilege Class: Any

Use the #CP command to execute one or more CP commands if you are not in a full-screen environment. After the CP command or commands have been executed your virtual machine will continue execution.

#CP	[command line ...]
-----	--------------------

where:

command line

is the name and operands for the CP command or commands you want to issue. You must precede the first command line with at least one blank. If you are specifying multiple CP commands, separate them with your logical line end symbol (see Usage Note 2).

Usage Notes

- The #CP command is a handy method to reduce the number of operations you must perform when you want to issue a CP command from a virtual machine command environment. For example, you can display the number of files you have on your virtual machine system by issuing the command:
#CP QUERY FILES
This is equivalent to performing the following sequence of operations:
 - Pressing the PA1 key to enter the CP command environment
 - Entering the QUERY FILES command
 - Entering the CP BEGIN command.
- The pound sign (#) shown in the preceding format represents the logical line end symbol currently in effect for your virtual machine. If you have redefined the logical line end symbol, you must substitute your line end symbol for the pound sign.
- Before you issue the #CP command, be sure that:
 - Your virtual machine is operating with the CP SET LINEDIT function set ON. This is the default.
 - The first three characters you enter in the command line are #CP (uppercase or lowercase); the “#” represents the currently defined logical line end symbol.
 - At least one blank separates the #CP from any command line.
 - Your virtual machine issued a READ for at least four bytes of data.
- You can enter multiple command lines as operands of the #CP command provided that:
 - Your virtual machine’s console READ is for at least the number of bytes required to enter the entire line.
 - You separate all the command lines that you want to execute with the current logical line end symbol.
- If you enter #CP without operands, your virtual machine enters the CP environment. You may then enter any number of CP commands. (If you have SET RUN ON, you may issue only one command.) To return to virtual machine operation, enter the CP BEGIN command.

If you enter #CP during a virtual machine read (that is, when “VM READ” appears in the lower right corner), CP terminates your virtual machine’s console read by returning a unit exception for the virtual console. The operating system that is running in your virtual machine then reissues the console READ to allow you to key in the appropriate response to a previous message from that machine’s operating system.

6. The #CP command has no meaning if:
 - You are running an operating system in your virtual machine that has a virtual console defined as a 3270
 - You are using DIAGNOSE Code X'58' – 3270 VIRTUAL CONSOLE INTERFACE, CCW operation codes 29 or 2A.
7. See Table 5 for a summary of system responses to various #CP commands.

Command	System Action
#CP	Your virtual machine enters the CP environment.
#CP QUERY FILES	The QUERY FILES command is executed.
#CP QUERY FILES#QUERY USERS	Two separate QUERY commands are executed.
#CP data entered	You enter the CP environment and CP interprets the “data entered” as a CP command, which may or may not be a valid command.

Responses

If you enter #CP without a command line, you receive this response:

CP

and the status area of your display device will indicate “CP READ.” If you enter #CP with a command line, you receive the responses that correspond to the individual CP commands you entered.

ACNT

Privilege Class: A

Use the ACNT command to create accounting records for logged-on users and to reset accumulated accounting data.

ACNT	$\left\{ \begin{array}{l} \text{userid1 [userid2... userid10]} \\ \text{SYSTEM} \\ \text{ALL} \end{array} \right\}$
------	---

where:

userid1 [userid2... userid10]

is the list of users for whom accounting records are to be created. You can specify up to 10 userids on the command input line.

SYSTEM

is used to account for all time not charged to a specific user.

ALL

specifies that accounting records are to be created for all logged-on users, and for SYSTEM.

Usage Notes

1. When you issue the ACNT command, an accounting record is created for each user that you specify on the command line, including accounting records for any dedicated devices or T-disks. The record indicates the amount of time that user's virtual machine has used since the last ACNT command was issued for the user, or since the user logged on.

Vector time is a subset of CPU time; that is, using a second of vector CPU time results in incrementing both the CPU time and the Vector Facility time counts. This applies to both virtual (guest) and total time. The reason for this is that using the VF implies using the CPU; all VF usage arises from executing instructions on that CPU.

2. CP processes the list of userids in the same order that you entered them on the command line. If any userid in the list is invalid, accounting records are created only for those userids that precede the invalid one.
3. To provide an ACNT command syntax that is compatible with VM/SP, VM/XA SP accepts CLOSE as an option following ALL or ACNT (for example, ACNT ALL CLOSE or ACNT CLOSE). However, this option does not change the operation of the ACNT command in VM/XA SP.

Responses

None.

Migration Notes**VM/SP HPO:**

1. VM/XA SP accepts but ignores CLOSE if specified.
2. VM/XA SP does not send a response to the user indicating accumulated accounting data.
3. VM/XA SP supports an additional operand: SYSTEM.

ATTACH

Privilege Class: B

Use the ATTACH command to logically connect:

- A real or logical device to a virtual machine for that virtual machine's exclusive use
- A DASD device to CP
- A partition of real Expanded Storage to a virtual machine for that virtual machine's exclusive use.

ATTach	<pre> {rdev} [TO] {userid} [[AS] vdev] [R[/O]] [[WITH] NOASSIGN SYSCTL DEVCTL NOCTL } {ldev} [TO] {userid} [[AS] vdev] [R[/O]] } {rdev} [TO] SYSTEM [[AS] valid] } {rdev ... } [TO] {userid} [R[/O]] [[WITH] NOASSIGN {rdev-rdev ... } {*} SYSCTL DEVCTL NOCTL } {ldev ... } [TO] {userid} [R[/O]] {ldev-ldev ... } {*} {rdev ldev ... } } {rdev ... } [TO] SYSTEM {rdev-rdev} } XSTore [TO] {userid} [nnnnnnnM] XStorage {*} [ALL] </pre>
--------	--

where:

rdev

rdev ...

rdev-rdev ...

specifies a real device number, a list of real device numbers or a range of real device numbers to be attached to the specified user or to the CP system. Specify a real device number as four hexadecimal digits, noting that leading zeros may be omitted. The maximum number of devices allowed in a range is 256.

If you are not specifying a virtual device number or volume label for the real device that you are attaching, the virtual device number defaults to the real device number. You may combine device numbers, lists of device numbers, and ranges of device numbers on a single command line.

If you are specifying a virtual device number for the device, you can specify only one real device number when you issue the ATTACH command.

ldev**ldev ...****ldev-ldev ...**

specifies a logical device number, a list of logical device numbers or a range of logical device numbers to be attached to the specified user. Specify a logical device number as an "L" followed by up to four hexadecimal digits, noting that leading zeros may be omitted.

If you are not specifying a virtual device number for the logical device that you are attaching, the virtual device number defaults to the logical device number. You may combine device numbers, lists of device numbers, and ranges of device numbers on a single command line.

If you are specifying a virtual device number or volume label for the device, you can specify only one logical device number when you issue the ATTACH command.

rdev ldev ...

specifies a list of both real and logical device numbers.

[TO] userid
[TO]*

is the userid of the virtual machine that is to receive the device or devices or Expanded Storage. That virtual machine must be currently logged on to the system. If you are attaching either a device or Expanded Storage to your own virtual machine, you can specify your userid as an asterisk (*).

You must use TO when:

- You specify a userid with 3 numbers or less in a range of addresses or multiple device addresses
- You specify a userid that could also be a valid hexadecimal device address (for example, CE)
- You specify a userid as "to" or "t."

[AS] vdev

is the virtual device number to be assigned to the device.

Note: If your virtual machine is operating in System/370 mode, vdev is the System/370 virtual device address (channel, control unit, and device) and may consist of up to four digits. If your virtual machine is operating in 370-XA mode, vdev specifies the virtual device number. It is assigned to an available virtual subchannel number, which may consist of up to four digits. If the device is attached to the preferred guest, then an effort is made to assign the real subchannel number as the virtual subchannel number.

R/O

is optionally used for DASD and tape devices that are attached to a user. This allows the system operator to apply read-only protection to a user's disk or tape.

CP ignores the R/O operand for any device number that does not correspond to a TAPE or DASD device.

[WITH] NOASSIGN

specifies that the tape device that is to be attached to the user is not to be assigned to the user's system.

CP ignores the NOASSIGN operand for any real tape device that does not support the ASSIGN/NOASSIGN CCWs.

Warning: CP does not REWIND/UNLOAD when the tape is detached.

[WITH] SYSCTL
[WITH] DEVCTL
[WITH] NOCTL

determine the CCWs, and options within CCWs, that are accepted by this device.

SYSCTL means that the device accepts CCWs that have a direct global effect on subsystem resources and functions, not just CCWs that are related to the device. For example, subsystem storage may be turned on or off. SYSCTL is the default for V=R and V=F machines.

DEVCTL means that the device accepts CCWs that have an effect on resources and functions directly related to the device. For example, caching may be turned on or off for the device. DEVCTL is the default for the V=V (nonpreferred) guest.

NOCTL means that the device accepts no CCWs that can exercise control of subsystem resources or functions, regardless of whether they directly relate to the device. For example, subsystem storage may not be turned on or off and caching may not be turned on or off for the device.

[TO] SYSTEM

specifies that the DASD device is to be made available to CP for accessing users' virtual disks and/or for paging, spooling and temporary disk functions.

[AS] valid

is the volume label of the volume on the device being attached. The volume label on the device will be verified and must be unique when compared to all other volumes attached to the system.

XSTORE

XSTORAGE

indicates that Expanded Storage is to be attached to the specified user.

XSTORE [TO] {userid}
XSTORE [TO] {*}

specifies the userid of the virtual machine that is to receive the real Expanded Storage. The virtual machine must be currently logged on to the system. The userid of SYSTEM is not valid. * specifies the userid of the virtual machine that issues the ATTACH XSTORE command.

nnnnnnnM

specifies the size of real Expanded Storage that is to be assigned to the virtual machine. If the Expanded Storage exceeds the maximum available for guest virtual machine use, then the maximum contiguous Expanded Storage is assigned.

ALL

specifies that all of Expanded Storage that has not been reserved for CP use with the RETAIN XSTORE command is attached to the user's virtual machine for exclusive use. This is the default.

Usage Notes

1. If you are attaching a device with logical addresses, both the real device number and the virtual device number you specify must be the base address.
2. When you attach a device to a 3880 Model 11 or 21 with SYSCTL or DEVCTL specified, the base address and the paging addresses are processed. If you specify NOCTL, only the base address is attached.

3. If you mount and ready a device that contains virtual minidisks after you have loaded VM/XA SP onto the system, you must ATTACH that device to the system if you want to use it. If you log on after the device has been attached, you will gain access to the disk as provided by your entry in the system directory. If you were logged on before the volume was attached to the system, you must use the CP LINK command to gain access to the virtual disks.
4. All the device numbers you specify will be processed even if errors are encountered when attempting to attach the devices. Thus, if you issue the following command:

```
attach 108-10F user1 r
```

and real devices 10A and 10D are not ready and online, the following devices would be successfully attached:

```
108, 109, 10B, 10C, 10E, 10F
```

Messages tell you which devices were successfully attached. Error messages identify the devices that were not attached.

5. Attaching a cache device to a userid while any other cache is in use by a system results in an error message.
6. You can use the system directory DEDICATE statement to attach real devices to their virtual machine. Then, when they have successfully logged on to the system, the devices (if online and available) are automatically attached to their virtual machine. This method can save them the trouble of issuing multiple ATTACH commands for the devices.
7. The ATTACH XSTORE command allows a class B operator to assign Expanded Storage to a guest virtual machine without authorization in the user directory. The command size parameter allows the operator to attach any size up to the maximum Expanded Storage allowed for guest use on the system, regardless of the user directory limit. The maximum allowed for a guest virtual machine is the total XSTORE less what is retained by the system. For more information, see the RETAIN command.

The size parameter must be in units of 1 megabyte. The size of the Expanded Storage feature dedicated to the guest is displayed on the response to the originator of the command as well as to the owner of the Expanded Storage. When the user issues QUERY XSTORE, the dedicated Expanded Storage for a user appears as an Expanded Storage feature equal to the size provided in the ATTACH XSTORE command. The actual size of Expanded Storage depends on whether any non-configured Expanded Storage exists within the user's Expanded Storage partition.

8. The system directory XSTORE statement directs CP to attach Expanded Storage to a virtual machine during logon. When logon is complete, Expanded Storage (if available) is automatically attached to the user's virtual machine.
9. When running on a partitioned processor, the amount of Expanded Storage attached to a user may appear larger than the amount requested. For example, a 3090-400E with 128 megabytes of XSTORE may be partitioned with 64 megabytes of XSTORE on each side. In this example, the lower block addresses are on one side, the 'A' side, and the higher block addresses are on the other side, the 'B' side. With CP running on the 'B' side, attaching 4 megabytes of XSTORE to a user results in CP responding with 4 megabytes attached. However, the guest operating system sees 68 megabytes attached with 64 megabytes not configured.

10. Issuing the ATTACH XSTORE command at initialization suspends virtual machine execution until XSTORE initialization completes. To avoid this, you can (a) wait until XSTORE initialization completes to issue ATTACH XSTORE or (b) issue ATTACH XSTORE from a disconnected virtual machine that has been automatically logged on.
11. SYSCTL/DEVCTL/NOCTL apply only to DASD on a cached control unit. If one of these operands is specified for a noncached DASD, the device is attached and a response is issued stating that the operand was ignored.
12. If you attempt to use DEVCTL or NOCTL to attach cached DASD to a V=R or V=F guest, the DASD is attached with SYSCTL and a response is issued stating that an attach is forced to be SYSCTL for a preferred virtual machine. This is necessary because a V=R or V=F guest may run with the start interpretive execution assist or without CCW translation (see the SETCCW command), and thus bypass CP's enforcement of the level of control.
13. Specifying a level of control (NOCTL, DEVCTL, or SYSCTL) for a DASD device attached to a cache storage control unit authorizes CP to accept particular control CCWs. Table 6 shows you the additional control CCWs CP accepts for a level of control. The DEVCTL column shows the CCWs CP accepts in addition to those it accepts for the NOCTL level of control. The SYSCTL column shows the CCWs CP accepts in addition to those it accepts for the DEVCTL and NOCTL levels of control.

Table 6. 3880 Control Levels

3880 Model	Additional Control CCWs Accepted for DEVCTL	Additional Control CCWs Accepted for SYSCTL
3880-11	Discard block (not all blocks) Set paging parameters	Discard block (all blocks)
3880-13	Set subsystem mode (device option)	Set subsystem mode (subsystem option) Set high performance storage limits
3880-21	Set subsystem mode (device option) Discard block (not all blocks) Set paging parameters	Set subsystem mode (subsystem option) Discard block (all blocks)
3880-23	Set subsystem mode (device option)	Set subsystem mode (subsystem option)

14. Note that the ATTACH command does not execute synchronously when you are attaching real devices to another userid. That is, you (the issuer of the command) may receive control BEFORE all processing has been completed on behalf of the other user. Any later attempt you make to use that device before it becomes attached may yield unexpected results.

For example, suppose the following command sequence exists in a CMS EXEC procedure:

```
CP DETACH 250 USERA
CP ATTACH 250 SYSTEM
```

CP will begin to ATTACH device 250 to the system before the DETACH command completes. USERA has not completed all processing when the DETACH is issued. Since the device is still attached to USERA, you receive an error message for the ATTACH command.

Thus, if you intend to issue commands using the same real device owned by another user, do not string the commands together (using the logical line end character) and do not issue the commands one right after the other in a CMS EXEC.

15. You cannot be certain that the device has been attached until you receive the appropriate response from the system. If after a reasonable amount of time you haven't received the expected response, issue the QUERY command to determine the status of the device. Attachment may be pending (see Usage Note 14) or may have been unsuccessful. Note that if you issued the command from the CMS environment and received a return code of 0, you cannot assume that the device has been attached. A return code of 0 indicates only that your request has been accepted.

Response 1

The following responses go to the receiver of the device:

```
type rdev ATTACHED TO userid vdev [R/0] [WITH SYSCTL
                                     WITH DEVCTL
                                     WITH NOCTL]
```

informs the user that the device has been attached to his or her virtual machine. If more than one device number was entered, a response is displayed for each device attached.

```
DASD vdev vdev vdev vdev ATTACHED [R/0] [WITH SYSCTL
                                         WITH DEVCTL]
```

occurs when attaching a DASD on a 3880 Model 11 or 21 subsystem. Notice that WITH NOCTL has been deleted. Only the base address gets attached if NOCTL is specified on the ATTACH command.

Note: In the above and all following ATTACH responses, "type" is one of the following:

Type	Meaning
DASD	Direct access storage device
TAPE	Magnetic tape
RDR	Card reader
PRT	Printer
PUN	Card punch
GRAF	Display device
CTCA	Channel-to-channel adapter
CTLR	3704 or 3705 Communications Controller
MSC	Mass Storage Controller
DEV	Any other device

Response 2

Depending on the command format the class B user issued, this response indicates the device(s) attached to a specified user's virtual machine (and, if appropriate, the virtual device number access mode and the level of control for the device) or indicates that the device has been attached to the CP system. You receive this response for each device that is attached.

The following responses go to the originator of the ATTACH command:

type {rdev} ATTACHED TO userid vdev [R/0]
 {ldev}

- or -

DASD rdev ATTACHED TO userid vdev [R/0] [WITH SYSCTL]
 [WITH DEVCTL]
 [WITH NOCTL]

- or -

type rdev ATTACHED TO SYSTEM valid

- or -

DASD rdev rdev rdev rdev ATTACHED TO SYSTEM valid

occurs when attaching a DASD on a 3880 Model 11 or 21 subsystem to VM/XA SP.

- or -

DASD rdev rdev rdev rdev ATTACHED TO userid vdev vdev vdev vdev [R/0] [WITH SYSCTL]
 [WITH DEVCTL]

occurs when attaching a DASD on a 3880 Model 11 or 21 subsystem. Note that WITH NOCTL has been deleted. Only the base address gets attached if NOCTL is specified on the ATTACH command.

Response 3

The responses in this category are the responses which can be sent to the primary system operator (if this operator did not issue the command) indicating that the device has been attached either to the specified user's machine (at virtual device number vdev) or to the CP system. The "originator" field designates the userid of the user who issued the command.

The following responses go to the system operator:

type {rdev} ATTACHED TO userid vdev BY originator [R/0]
 {ldev}

- or -

DASD rdev ATTACHED TO userid vdev BY originator [R/0] [WITH SYSCTL]
 [WITH DEVCTL]
 [WITH NOCTL]

- AND -

type rdev ATTACHED TO SYSTEM valid BY originator

ATTACH

- or -

DASD rdev rdev rdev rdev ATTACHED TO SYSTEM valid BY originator

occurs when attaching a DASD on a 3880 Model 11 or 21 subsystem to VM/SP.

DASD rdev rdev rdev ATTACHED TO userid vdev vdev vdev BY originator [R/O] [WITH SYSCTL]
[WITH DEVCTL]

occurs when attaching a DASD on a 3880 Model 11 or 21 Subsystem. WITH NOCTL has been deleted. Only the base address gets attached if NOCTL is specified on the ATTACH command.

Response 4

ctllevel IGNORED - INVALID FOR rdev

informs the originator of the command that control options have been specified for a noncached DASD.

where:

ctllevel

is the level of control: SYSCTL, DEVCTL, or NOCTL.

rdev

is the real device number.

Response 5

ctllevel IGNORED - SYSCTL REQUIRED FOR PREFERRED VIRTUAL MACHINE

informs the originator of the command that invalid control options have been specified for a cached DASD being attached to a preferred virtual machine.

where:

ctllevel

is the level of control: DEVCTL or NOCTL

Note: For Responses 6, 7, and 8:

The owner of Expanded Storage, the user who issued the command (if different from the owner), and the primary system operator (if different from the user who issued the command), each receives responses indicating that the ATTACH was successful.

Response 6

[XSTORE MIGRATION STARTED, MAY TAKE SEVERAL MINUTES TO COMPLETE]
XSTORE CLEARING STARTED
XSTORE ATTACHED, SIZE=nnnnnnnM

informs the guest that Expanded Storage has been attached to the virtual machine. Line 1 is issued when migration of pages from Expanded Storage to DASD is necessary, for example, when CP has already used some of the area to be attached to the guest.

Response 7

XSTORE ATTACHED TO userid

informs the Class B user who issued the ATTACH XSTORE command that Expanded Storage has been attached to the specified virtual machine.

Response 8

XSTORE ATTACHED TO userid BY originator

is the response sent to the primary system operator (if not the person issuing the command) indicating that Expanded Storage has been attached to the specified user's virtual machine. The word "originator" designates the userid of the issuer of the command.

Migration Notes*VM/SP HPO*

1. VM/XA SP supports neither the 3330V nor the CHANNEL operands.
2. In VM/XA SP if a person issues an ATTACH command on behalf of another user, the attachment does not take place immediately (it is asynchronous). The attachment occurs just prior to the user being dispatched.
3. VM/XA SP supports an additional operand: XSTORE.
4. VM/XA SP supports the attachment of logical devices.
5. VM/XA SP limits the maximum number of devices allowed.
6. VM/XA SP does not accept 'A' for 'AS'.
7. VM/XA SP accepts rdev * R/O, where rdev is a tape device to allow users to put in a check simulating a tape ring (instead of doing actual I/O to the device).

VM/XA SF: Multiple users may have Expanded Storage attached to their virtual machines.

ATTN

Privilege Class: G

Use the ATTN command to make an attention interruption pending for your virtual console.

ATTN	
------	--

Usage Notes

1. The CP REQUEST command performs the same function as ATTN; you can use the two commands interchangeably.
2. The CP BEGIN command is not required after you issue ATTN.
3. CP ignores the ATTN and REQUEST commands if an I/O operation is in progress at the console, or if another interrupt is pending for your virtual machine. This condition may also result if you issue the ATTN or REQUEST commands by means of DIAGNOSE code X'08'.

Responses

None.

AUTOLOG

Privilege Class: A, B

Use the AUTOLOG command to log on another virtual machine automatically. Before a virtual machine can be logged on automatically, it must be defined in the system directory and you must know its password. The virtual machine you automatically log on operates in disconnected mode.

Note: You can also use the XAUTOLOG command to log on another virtual machine. XAUTOLOG has parameters for defining the virtual machine and changing the command synchronization. You do not need a password with the XAUTOLOG command if you have class A or class B privileges. If you have class G privileges, you can use the XAUTOLOG command when you are authorized in the other virtual machine's system directory entry. Read the XAUTOLOG command description for more information about the additional functions available.

AUTOLOG	userid [password] [console-input-data]
---------	--

where:

userid

is the user identification of the virtual machine you are logging on.

password

is the password of the virtual machine you are logging on.

If your installation has password suppression active, all the data following the userid is console-input-data. Wait for the "ENTER PASSWORD ..." prompt before you type in the password. The password is automatically masked, thus improving system security.

If your installation allows a password to be specified on the AUTOLOG command line, the first data item following the userid is taken as the password. Any data following the password is console-input-data.

To determine whether password suppression is active, use the QUERY PASSWORD command.

console-input-data

is data that you want used as virtual console input in response to the first read issued by the user's virtual machine. You may use the remainder of the command line to enter console data; the entire line including any logical-line-end characters (#) are passed to the virtual machine.

Note: If you are using CMS and you specify console-input-data containing the logical-line-end character (#) on the AUTOLOG command line, CMS truncates the AUTOLOG command at the first logical-line-end character. To avoid this situation, precede the logical-line-end character with a logical-escape-character or precede the AUTOLOG command line with #CP, where # is your logical-line-end character.

AUTOLOG

Usage Notes

1. The virtual machine that you log on using the AUTOLOG command must have an IPL statement in its system directory entry.
2. The virtual machine that you log on using the AUTOLOG command operates in disconnected mode. The same restraints that apply to any disconnected machine also apply to virtual machines that are logged on with the AUTOLOG command.
3. CP disables the AUTOLOG command for the rest of a user's logon session if a user enters excessive incorrect passwords to automatically log on another userid. The initial system limit is 4 incorrect passwords. Use the QUERY CMDLIMIT command to see the system limit value set by your installation.
4. A secondary userid may be specified on the user's system directory CONSOLE statement. All console messages are routed to the secondary userid, if functional, once the user is automatically logged on.
5. If AUTOLOG is used in an EXEC or module, you must either put the password in the command or set the DIAGNOSE code X'08' control bit to indicate that you want a prompt. See *VM/XA SP CP Programming Services* for details about DIAGNOSE X'08'.

Response 1

ENTER PASSWORD (IT WILL NOT APPEAR WHEN TYPED):

is the response if you must enter a password. Enter the password of the userid you want to automatically log on.

Response 2

AUTO LOGON *** userid USERS = nnnnn

is the response to issuer of the AUTOLOG command (if different from the system operator) upon successful completion of the LOGON process of the user specified on the AUTOLOG command.

where:

indicates there is no assigned line address for the virtual console.

userid

is the identification of the virtual machine initiated by the AUTOLOG command.

USERS = nnnnn

is the number of logged-on users.

Response 3

```
hh:mm:ss AUTO LOGON ***      userid  USERS = nnnnn BY {SYSTEM }
                                {userid2}
```

is the response to the system operator upon successful completion of the LOGON process of the user specified on the AUTOLOG command.

where:

indicates there is no assigned line address for the virtual console.

userid

is the identification of the virtual machine initiated by the AUTOLOG command.

USERS = nnnnn

is the number of logged-on users.

SYSTEM

indicates that the AUTOLOG process has occurred as part of VM/XA SP system initialization. Typically, your primary system operator's virtual machine is logged on as part of the system initialization process.

userid2

identifies the user who issued the AUTOLOG command.

Response 4

```
hh:mm:ss USER DSC LOGOFF AS  userid  USERS = nnnnn FORCED
```

is the response to the system operator if an invalid IPL statement is processed for the user specified on the AUTOLOG command.

where:

userid

is the identification of the virtual machine initiated by the AUTOLOG command.

USERS = nnnnn

is the number of logged-on users.

Migration Notes

VM/SP HPO

1. VM/XA SP gives additional messages for VM/XA SP related situations.
2. If the number of incorrect passwords exceeds the system limit value, then **all** AUTOLOG and XAUTOLOG commands are disabled for the remainder of the session.

VM/XA SP

1. The former VM/XA SP1 AUTOLOG command (syntax and function) is now the XAUTOLOG command. The AUTOLOG command is now compatible with VM/SP HPO.
2. No additional functions can be specified on the AUTOLOG command using keyword parameters. If additional functions are required or you do not want to supply a password, use the XAUTOLOG command.

AUTOLOG

3. The AUTOLOG command requires the use of a password (either in the command or in response to a prompt).
4. The console-input-data is **not** preceded by the logical-line-end character (#).
5. The userid you are logging on must have an IPL statement defined in the system directory.
6. The AUTOLOG command operates synchronously; this means that VM returns control to you when the target userid is logged on and the IPL statement has been verified.

BACKSPACE/BACKWARD**Privilege Class:** D

Use the BACKSPACE command to restart or reposition the current output on a real punch or printer. The BACKSPACE command is identical to the BACKWARD command.

Printer BACKSPACE Format

BAckspace BAckward	rdev	[File nnn [EOF] <u>1</u>]
-----------------------	------	-----------------------------------

Punch BACKSPACE Format

BAckspace BAckward	rdev	[File]
-----------------------	------	--------

where:

rdev

is the real device number of the punch or printer that is to be backspaced.

FILE

specifies that the output spool printer or punch file is to be restarted from the beginning.

nnn

specifies the number of pages (in decimal, up to a maximum of 999) that you want to backspace on the printer. If you do not specify a value, the printer is backspaced to the beginning of the page being printed when the command is received.

EOF

specifies that backspacing is to start from the end of the spool file and will then continue backward the requested number of pages. Printing begins again from this new position.

Usage Notes

1. If you issue a BACKSPACE command while an intervention required condition is pending on your printer or punch, the command will not take effect immediately. Once the intervention required condition is cleared and the current outstanding I/O operation completes, then the BACKSPACE command will be performed.
2. In BACKSPACE operations, pages are counted by looking for skip to channel 1 specifications (that is, page ejects). If a file does not contain skip to channel 1 specifications, 66 output lines (including blank lines) are treated as a "page."

BACKSPACE/BACKWARD

Responses

$\left. \begin{matrix} \{ \text{PRT} \} \\ \{ \text{PUN} \} \end{matrix} \right\}$ rdev BACKSPACE userid FILE spoolid RECS $\left. \begin{matrix} \{ \text{nnnn} \} \\ \{ \text{nnnK} \} \\ \{ \text{nnnM} \} \end{matrix} \right\}$ COPY $\left. \begin{matrix} \{ * \text{nnn} \} \\ \{ \text{nnn} \} \end{matrix} \right\}$ SEQ sss

where:

PRT

PUN

indicates the type of device that has been backspaced.

rdev

is the real device number.

userid

is the identification of the user whose file was active and backspaced.

FILE spoolid

is the system-assigned spoolid number by which VM/XA SP refers to the file.

nnnn

nnnK

nnnM

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY $\left. \begin{matrix} \{ * \text{nnn} \} \\ \{ \text{nnn} \} \end{matrix} \right\}$

is the remaining number of copies of the file to be produced. The asterisk (*) is for the 3800 printer and indicates that the printer will copy each page nnn times before going on to the next page.

SEQ sss

is the sequence number for the file on the printer or punch.

Migration Notes

VM/SP HPO

1. VM/XA SP provides a one-line response instead of a two-line response as in VM/SP HPO.
2. VM/XA SP allows you to spell out the entire command BACKSPACE.
3. VM/XA SP does not support server virtual machines for the real spooling commands.

VM/XA SF: None.

BEGIN

Privilege Class: G

Use the **BEGIN** command to resume execution on a virtual multiprocessor configuration, or on a specific processor or processors.

Begin	<pre> * hexloc CPU { cpuaddr { cpuaddr1 cpuaddr... { cpuaddr1-cpuaddrn ALL } } } </pre>
-------	---

where:

 terminates hard-stop for the processor on which it is issued. No other virtual processors are affected.

hexloc
 is the hexadecimal storage location where execution is to begin for the processor on which the command is issued. It will terminate hard-stop for the processor on which it is issued without affecting any other processors. To begin multiple processors at specified hexadecimal locations, use the **CP CPU** command in conjunction with the **BEGIN** command.

CPU cpuaddr
 terminates hard-stop for the specified processors without affecting any other processors.

CPU cpuaddr1 cpuaddr...
 is the list of addresses and/or ranges delimited by blanks.

CPU cpuaddr1-cpuaddrn
 specifies a nonwrapping pair of addresses that is separated by a hyphen (“-”) and not separated by any blanks.

CPU ALL
 specifies all virtual processors currently defined.

Usage Notes

1. If you enter the **BEGIN** command without operands, it will terminate soft-stop on all virtual processors in your virtual machine configuration. If all of your processors are in hard-stop, the **BEGIN** command will terminate hard-stop for the processor on which it is executed.
2. If you enter the **BEGIN** command without designating an address where execution is to begin (hexloc), execution will begin at the location pointed to by the current virtual machine program status word (PSW). Unless the PSW has been altered since **CP** command mode was entered, the location stored in the PSW is the location where the virtual machine stopped.
3. If you enter the **BEGIN** command and specify a starting storage location, execution will begin at that address. The address you specify will replace the instruction address in the PSW, and then the PSW will be loaded.

BEGIN

4. **BEGIN** terminates **SLEEP** status. For more information about **SLEEP** status, see the **CP SLEEP** command.
5. The **BEGIN** command cannot be used to resume execution on check-stop processors. Refer to the **SYSTEM RESET** and **SYSTEM CLEAR** commands.

Responses

None.

Migration Notes

VM/SP HPO: *VM/XA SP* supports additional operands: * and CPU.

VM/XA SF: None.

CHANGE**Privilege Class: D,G**

Use the CHANGE command to alter one or more of the external attributes of a closed spool file or files, or to alter the user HOLD status of an active spool file (one currently being printed or punched).

If you are a class D user, use the CHANGE command to change any spool files on the system, regardless of origin, using the SYSTEM or userid options.

If you are a class G user, you cannot use the SYSTEM option; however, you can use the userid option of the CHANGE command to manipulate your own files or files that you created and subsequently transferred (as long as the transferred files are unchanged).

Change	[SYSTEM]	$\left. \begin{array}{l} \text{Reader} \\ \text{Printer} \\ \text{PUnch} \end{array} \right\}$	$\left. \begin{array}{l} \text{Class } c \\ \text{FORM form} \\ \text{ALL} \end{array} \right\}$	(options)
	[userid]	$\left. \begin{array}{l} \text{Reader} \\ \text{Printer} \\ \text{PUnch} \end{array} \right\}$	$\left. \begin{array}{l} \text{Class } c \\ \text{spoolid} \\ \text{FORM form} \\ \text{ALL} \end{array} \right\}$	(options)

where:

SYSTEM

allows the class D user to manipulate any files in the system, regardless of ownership. This is not allowed with the spoolid option.

userid

*

is the user identification for the user whose spool files are to be changed. If you are changing your own files, you may specify your userid as an asterisk (*), which is also the default.

READER**RDR**

changes reader spool file(s).

PRINTER**PRT**

changes printer spool file(s).

PUNCH**PCH**

changes punch spool file(s).

CLASS c

designates a spool file class, where *c* is a 1-character alphanumeric field from A to Z or 0 to 9.

FORM form

designates a 1- to 8-character form name. Use this form name to select those files you want to change. For the class G user, the form is the user form name. For the class D user, *form* is the user form name when the "*" option is specified (or accepts the default). Otherwise, *form* is the operator form number.

CHANGE

ALL

specifies the files on which the CHANGE command is to operate.

If you want to change all files, regardless of their class, form, or spoolid, specify ALL.

spoolid

is the spool file identifier by which you may select a particular file. *spoolid* is not valid with the SYSTEM keyword.

options

See "CHANGE: Options" on page 41 for a complete description of these options.

CHANGE: Options

Options:	<pre> [HOld] [KEep] [MSG] [SYS] [NOHOld] [NOKEep] [NOMsg] [NOSYS] [Class c] [COpy [*]nnn] [Dist {distcode} {OFF *}] [[To] {userid2} [Reader *] [Printer PUnch]] [FORM {form} {OFF}] [FLASH {name [ALL] name [nnn] {OFF} }] [MODify {name [n] {OFF} }] [FCB {name {6 {8 {10 {12 {OFF} } }] [[CHars {name0 [name1 [name2 [name3]]} {OFF} }]] [[CHars name0 [CHars name1 [CHars name2 [CHars name3]]]]] [NAME {fname [ftype] {dsname} }] NOName </pre>
----------	--

Note: At least one of these options must be chosen. If you specify more than one, you can combine them in any order, except for the NAME option which must be specified last.

HOLD
NOHOLD

HOLD prevents the punching, printing, or reading of a file by placing it in a user HOLD status. NOHOLD releases the specified spool file(s) currently in user HOLD status.

KEEP
NOKEEP

KEEP specifies that the spool file is to be retained in user HOLD status after processing. Specify NOKEEP to reset files that are in KEEP status.

CHANGE: Options

MSG

NOMSG

indicates whether messages are sent to the file owner when the file begins processing and again when the file completes processing. Although you may enter these options for any type of spool file, no message is sent when a virtual reader processes the file. Specify NOMSG to reset the MSG option.

If you have issued the CP SET IMSG OFF command, messages are not sent.

SYS

NOSYS

SYS prevents the punching, printing, or reading of a file by placing it in a system HOLD status. NOSYS releases the specified spool file(s) currently in system HOLD status. User HOLD status can coexist with system HOLD status.

This option is available to class D users only.

CLASS c

designates the new class ("c") for the file(s).

COPY [*]nnn

specifies the number of file copies you want printed or punched when the file is produced. The value of nnn (number of copies) must be a number from 1 through 255. Leading zeros do not need to be specified.

When you specify the optional asterisk (*), the 3800 printer prints each page from the spooled output files *nnn* times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (*) is ignored and duplication occurs normally.

DIST distcode

specifies the new distribution code for the spool file(s). *distcode* is a 1-to 8-character alphanumeric identifier that appears on the output separator of punch or print files.

DIST { OFF }
 { * }

resets the distribution code of the spool file(s) to the default specified in your entry in the system directory.

[TO] userid2

*

sends the spool file to the virtual machine specified by *userid2*. If no device type is specified immediately after *userid2*, the spool file is placed on *userid2*'s reader queue. Specify an * if you want to direct the output to your own virtual machine.

READER

RDR

PRINTER

PRT

PUNCH

PCH

specify, in conjunction with the TO operand, the file queue to which the spool file(s) are to be directed.

FORM form

designates the new form name for the spool file(s). *form* is 1 to 8 characters. For the class G user, *form* is the user form name. For the class D user, *form* is the user form name when the "*" option is specified. Otherwise, *form* is the operator form number.

FORM OFF

resets the form for the spool file(s) to the system default as specified by the installation.

FLASH name $\left[\begin{array}{l} \text{ALL} \\ \text{nnn} \end{array} \right]$

signifies that a form overlay contained in the 3800 printer is to be superimposed (or flashed) onto the specified number of copies of the output. The name, to be 1 to 4 characters in length, identifies the forms overlay to be used in the projection. The number, nnn, must be in the range from 0 to 255. The number specifies that the first nnn copies of output are to be flashed.

For example, if you specify COPY 4 and FLASH name 2, the first two copies of your file are flashed, while the last two copies remain unflashed. If ALL is specified, the flash count always matches the copy count. ALL is the default.

FLASH OFF

resets the flash parameters. Blanks are inserted into the flash name field of the specified spool file(s), and the count is set to 0.

MODIFY name [n]

assigns a copy modification module to be used when printing the spool file(s) on the 3800. This function makes it possible to alter text by either preventing printing of certain information or adding labels to output. The name must be 1 to 4 characters.

n selects one of the character sets specified by the CHARS operand to be used for the copy modification text. The values 0, 1, 2, or 3 correspond to the order in which the character sets have been specified. If no *n* is specified, the first character set (0) is assumed.

MODIFY OFF

resets the modify parameters. Blanks are inserted into the modify name field of the spool file(s) and, when printed, no modification occurs. The modify number (*n*) is set to 0.

FCB $\left(\begin{array}{l} \text{name} \\ 6 \\ 8 \\ 10 \\ 12 \end{array} \right)$

specifies the forms control buffer to be used when printing the file on the 3800 printer. *name* is a 1- to 4-character name of an FCB image, or the number 6, 8, 10, or 12. The number tells the 3800 to print the entire spool file at 6, 8, 10, or 12 lines per inch, regardless of the size of the paper that is currently loaded. Use of these unique numbers assures that printed material will fit onto a specified page size. Note, however, that FCB 10 prints only on a 3800 Model 3 printer.

FCB OFF

resets the FCB parameter. Blanks are inserted into the FCB name field of the spool file(s) and, when printed, a default FCB controls the vertical spacing of the output text.

CHARS name0 [name1 [name2 [name3]]]**CHARS name0 [CHARS name1 [CHARS name2 [CHARS name3]]]**

specifies the name of the character set(s) used when printing the spool file(s) on a 3800 printer. The name(s) must be from one to four characters with a maximum of four names specified. The multiple use of CHARS need only be used if *nameN* would conflict with an option name (for example, FORM) or with a userid. If more than one name is specified, and modification is also specified, the order in which the names appear determines which table is used for copy modification text.

CHARS OFF

resets the CHARS parameters. Blanks are inserted into the name fields of the character sets used for output generation. The spool file(s), therefore, use a default character set when actually printed.

NAME { **fname** [**ftype**] }
 { **dsname** }

NONAME

indicates the name is to be assigned to the spool file for identification. For output files, this name appears on the separator page or cards. Use the *fname* *ftype* option to name files in the CMS filename and filetype format. Each of these fields is a 1- to 8-character alphanumeric name. If you omit the *ftype* field, blanks are substituted. If you want to name files in a format suitable for MVS use (for example, SYS1.SYSLIB.MAC1), use the *dsname* option. You may specify the *dsname* up to 24 characters in length; however, if the *dsname* is more than 16 characters in length, it is truncated to 16 characters.

Specify NONAME to reset the name information to blanks.

If specified, the NAME option must be the last operand you enter on the command line.

Usage Notes

1. You can use the CP QUERY RDR/PRT/PUN command to determine the attributes of a spool file before and after the CHANGE command.
2. You may change the status of the file to HOLD in order to stop the processing of the file that is currently being printed or punched. No other attributes of an active file may be changed.
3. Class G users may change files that they own or that they have created.
4. If you use the CHANGE command to transfer a file to another user and you do not specify a distribution code, the distribution code is set to the receiving user's default distribution.
5. Although the CHANGE options FLASH, FCB, MODIFY, CHARS, and COPY *nnn may be specified for any spool file or files, they only cause system action when a spool file with these attributes is printed on a 3800 printer.
6. A user may also cancel settings of the CHARS, MODIFY, or FCB options by specifying NULL in the name fields of these operands. In these cases, NULL performs the same function as OFF.
7. The keyword OFF, which resets attributes to their default values, may not also be used to specify a FORM, FLASH, FCB, MODIFY, DISTcode, or CHARS name. Similarly, NULL may not be used to specify a CHARS, MODIFY, or FCB name.

8. If the NORETURN option was specified when the file was created by DIAGNOSE X'94', then the originator of the file cannot retrieve that file from the current owner. If you try, the following message is issued:

FILE NOT CHANGED; FILE CREATED WITH THE NORETURN OPTION

9. When a file changes ownership, the spool file id number also changes. The new number is obtained from the pool of available IDs of the recipient of the file. Thus, if user DORA uses the CHANGE command to send file #3 to user LILA, DORA gets the response:

RDR 0003 SENT TO LILA RDR AS 0018 RECS 1000 COPY 001 A NOHOLD NOKEEP

LILA would now have a file #18.

Response 1

{ nnnnnn } FILE(S) CHANGED
 { NO }

indicates how many files have been changed. This response is sent to the user who issued the CHANGE command but is suppressed if the virtual machine user has issued the SET IMSG OFF command.

Response 2

If you transfer a spool file to another user using the TO operand, the following response is sent to you and to the user receiving the file (if that user is active on the system):

{ RDR } FILE spd1 SENT { TO } userid { RDR } AS spd2 RECS { nnnn } COPY [*]nnn c { HOLD } { KEEP }
 { PRT } { FROM } { PRT } { nnnK } { NOHOLD } { NOKEEP }
 { PUN } { PUN } { nnnM }

where:

RDR
PRT
PUN

indicates the spool file queue. If you are transferring a file to another user's virtual machine, this field indicates the queue from which the file originated. If you are receiving a file from another user, this field indicates the file queue in which the file has been added.

spd1

is the spool identification number (spoolid) at the time the command is issued.

TO userid

indicates to whom the file has been sent.

FROM userid

indicates from whom the file came.

RDR
PRT
PUN

indicates the spool file queue. If you are transferring a file to another user's virtual machine, this field indicates the queue in which the file has been added. If you are receiving a file from another user, this field indicates the file queue in which the file originated.

spd2

is the spool identification number of the file after the command is complete.

CHANGE: Options

RECS { nnnn
 nnnK
 nnnM }

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999,499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

specifies the number of file copies you want printed or punched when the file is produced.

When the optional asterisk (*) is specified, the 3800 printer prints each page from the spooled output files nnn times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (*) is ignored and duplication occurs normally.

c

is the spool file class assigned to the file.

HOLD

NOHOLD

is the user HOLD status associated with the file.

KEEP

NOKEEP

is the KEEP status associated with the file.

This response is suppressed if the virtual machine user has issued the SET IMSG OFF command.

Response 3

{ PRT } rdev { PRINTING } userid FILE spoolid RECS { nnnn } COPY [*]nnn SEQ sss
{ PUN } { PUNCHING }

is the message you receive when the system begins processing a spool file that has the MSG attribute.

where:

PRT

PUN

indicate the type of device on which the file is being processed.

rdev

is the real device number.

userid

is the name of the spool file owner.

FILE spoolid

is the spool file spoolid number.

RECS { nnnn
nnnK
nnnM }

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

specifies the number of file copies you want printed or punched when the file is produced.

When the optional asterisk (*) is specified, the 3800 printer prints each page from the spooled output files nnn times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (*) is ignored and duplication occurs normally.

SEQ sss

is the sequence number for the file on the printer or punch.

Migration Notes

VM/SP HPO

1. VM/XA SP truncates a name identification for dsnames to sixteen characters (if longer than sixteen).
2. VM/XA SP does not allow users to specify SYSTEM and spoolid on the same command line. VM/SP HPO (Release 5) users do not see this difference.
3. The VM/XA SP response includes the number of records and KEEP setting.
4. If the file is changed to another user, the VM/XA SP response includes the new spoolid.
5. VM/XA SP supports additional options: TO, KEEP/NOKEEP, and MSG/NOMSG.
6. VM/XA SP does not support the DEST or the UNCONV options.
7. VM/XA SP sets DIST to what is in the OPTIONS directory when * (asterisk) is specified.
8. VM/XA SP displays a 7-digit response for the number of files changed.

VM/XA SF

1. VM/XA SP does not allow users to specify SYSTEM and spoolid on the same command line;
2. If the file is changed to another user, the VM/XA SP response includes the new spoolid.

CLOSE**Privilege Class: G**

Use the CLOSE command to terminate spooling activity on a virtual spooling device (reader, printer, punch, or console). Options on the CLOSE command override the options set on the virtual device or spool file(s).

Close	<pre>{Reader } [OptionA] {Printer } [OptionA] [OptionB] PUnch {CONsole } {vdev }</pre>
-------	--

where:

READER**RDR****PRINTER****PRT****PUNCH****PCH****CONSOLE**

closes all reader, printer, punch or console devices. Once a virtual console spool file is closed, it becomes a printer spool file and can be manipulated in the same way as any printer spool file.

vdev

is the virtual device number of the device to be closed. The device number may represent a console, reader, printer, or punch.

OptionA**OptionB**

See "CLOSE: Options" on page 49.

The options of the CLOSE command override the corresponding options in effect for the device that you are closing. If you do not specify any of the optional operands, then the options in effect for the device or the status of the file that you are processing will determine the file's disposition.

CLOSE: Options

OptionA:	{ [H0ld] [KEep] [EOF] [PUrge] } { [NOH0ld] [NOKeep] }
OptionB:	{ [Class c] [MSG] [NOMsg] [COpy [*]nnn] [Dist(distcode)] { OFF } * } [[To] { userid } { <u>Reader</u> } Printer PUnch]] [FORM { form } OFF] [FLash { name [ALL] name [nnn] } OFF] [MOdify { name [n] } OFF] [FCB (name) 6 8 10 12 OFF] [[CHars { name0 [name1 [name2 [name3]] }] OFF] [[CHars name0 [CHars name1 [CHars name2 [CHars name3]]]]] [NAME { fname [ftype] } dsname] NOName]

Note: You can specify one or more of these options. If you specify more than one, you can combine them in any order, except for the NAME option which must be specified last.

where:

HOLD**NOHOLD**

controls the user HOLD status.

For output devices, HOLD prevents files from being punched or printed on a real device by putting a user HOLD status on the file. If you specify NOHOLD, you will override the spool device HOLD option and prevent the file from being placed in HOLD status.

For input devices, the HOLD/NOHOLD options override the setting for the virtual device or the spool file. The file is not put in HOLD status unless the KEEP option is also specified. If you specify HOLD, the file will not be purged after processing and will remain eligible for the device. If you specify NOHOLD, the file will be purged unless the KEEP option is also specified.

KEEP

NOKEEP

controls the KEEP status, which indicates whether the current spool file is to be placed in user HOLD after processing.

For output files, the KEEP status will be placed on the spool file. After it is printed or punched, the file will be assigned user HOLD status, and will not be purged. The file's KEEP status is retained. Specify NOKEEP to override the current KEEP setting for the device so that KEEP is not set on the spool file.

For input files, KEEP overrides the KEEP setting on either the virtual reader or the spool file. After processing, the file is placed in HOLD status, and the KEEP option on the file remains unchanged. Specify NOKEEP to override either file or device KEEP characteristics. The disposition of a file closed NOKEEP (whether it is purged or retained) is controlled by the HOLD status of the virtual reader. If it were closed KEEP, it would never be purged.

EOF

closes the spool file on the specified device regardless of the CONT setting for the device. This option is not meaningful for a reader.

PURGE

closes and purges the spool file on the specified device regardless of the CONT, KEEP, and HOLD setting for the device.

CLASS c

specifies the spool class to be assigned to the file, where c is a one-character alphanumeric field from A to Z or 0 to 9.

MSG

NOMSG

indicates whether you want to be informed about file printing and punching. If you specify MSG, you will be informed of file printing and punching. The message is sent to you when the file begins either printing or punching on a real device. This message is not sent if you have issued CP SET IMSG OFF command. Specify NOMSG to reset the MSG option.

COPY [*]nnn

specifies the number of file copies you want printed or punched when the file is produced. The value of nnn (number of copies) must be a number from 1 through 255. You do not need to specify leading zeros.

When the optional asterisk (*) is specified, the 3800 printer will print each page from the spooled output files nnn times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (*) is ignored and duplication occurs normally.

DIST distcode

specifies the distribution code to be assigned to the resulting spool file. *distcode* is a one-to-eight character alphanumeric identifier which will appear on the output printer/punch separator.

DIST { **OFF** }
 { * }

resets the distribution code for the resulting spool file to the default specified in the system directory for your userid.

[TO] { **userid** }
 { * }

directs the spool file from the device to the virtual machine specified by the userid. Specify an asterisk (*) if you want to direct the file to your own virtual machine.

READER
RDR
PRINTER
PRT
PUNCH
PCH

are used with the **TO** operand to specify the file queue to which the spool file is to be directed. If you do not specify a queue, the file will be directed to the reader.

FORM form

overrides the form name of a spool file. The form name is 1-to-8 characters. If **FORM** is not given for the **CLOSE** command, the file created has one of the following forms: (1) form as specified by the last **SPOOL** command for the device, or (2) a default name specified during system generation.

FORM OFF

resets the form to the default specified during system generation.

FLASH name { **ALL** }
 { **nnn** }

signifies that a form overlay contained in the 3800 printer is to be superimposed (or flashed) onto the specified number of copies of the output. The name, to be 1-to-4 characters in length, identifies the forms overlay to be used in the projection. The number, **nnn**, must be in the range from 0 to 255. The number specifies that the first **nnn** copies of output are to be flashed.

For example, if you specify **COPY 4** and **FLASH name 2**, the first two copies of your file are flashed, while the last two copies remain unflashed. If **ALL** is specified, the flash count will always match the copy count. **ALL** is the default.

FLASH OFF

resets the flash parameters. Blanks are inserted into the flash name field of the specified spool file, and the count is set to 0.

MODIFY name [n]

assigns a copy modification module to be used when printing the spool file on the 3800. This function makes it possible to alter text by either preventing printing of certain information or adding labels to output. The name must be one to four characters.

n selects one of the character sets specified by the **CHARS** operand to be used for the copy modification text. The value 0, 1, 2, or 3 corresponds to the order in which the character sets have been specified. If no *n* is specified, the first character set (0) is assumed.

MODIFY OFF

resets the modify parameters. Blanks are inserted into the modify name field of the spool file and, when printed, no modification will occur. The modify number (*n*) is set to 0.

FCB { **name** }
 { 6 }
 { 8 }
 { 10 }
 { 12 }

specifies the forms control buffer to be used when printing the file on a 3800 printer. *name* is the 1- to 4-character name of an FCB image, or the number 6, 8, 10, or 12. This number tells the 3800 to print the entire spool file at 6, 8, 10, or 12 lines per inch, regardless of the size of the paper that is currently loaded. Please remember, however, that FCB 10 will print only on a 3800 Model 3 printer.

FCB OFF

resets the FCB parameter. Blanks are inserted into the FCB name field of the spool file and, when printed on a 3800 printer, a default FCB will control the vertical spacing of the output text.

CHARS name0 [name1 [name2 [name3]]]

CHARS name0 [CHars name1 [CHars name2 [CHars name3]]]

specifies the name of the character set(s) used when printing the spool file on a 3800 printer. The name(s) must be from one to four characters with a maximum of four names specified. The multiple use of CHARS need only be used if *nameN* would conflict with an option name (for example, FORM) or with a userid. If more than one name is specified, and modification is also specified, the order in which they appear determines which table will be used for copy modification text.

CHARS OFF

resets the *chars* parameters. Blanks are inserted into the name fields of character sets to be used for output generation. The resulting spool file, therefore, will use a default character set when actually printed on a 3800 printer.

NAME { **fname [ftype]** }
 { **dsname** }

NONAME

allows you to assign a name to the spool file for identification.

The *fname*, *ftype* specification permits you to name a file in CMS filename and filetype format. The fields *fname* and *ftype* are one to eight character alphameric names. If you omit the *ftype* field, blanks will be substituted.

The *dsname* specified may consist of up to 24 characters; however, if more than 16 characters are specified, the name will be truncated to 16 characters. An example could be: SYS1.SYSLIB.MAC1.

Specify NONAME to reset the name information to blanks.

If specified, the NAME option must be the last operand you enter on the command line.

Usage Notes

The optional operands of the CLOSE command override the corresponding options in effect for the device that you are closing. If you do not specify any of the optional operands, then the options in effect for the device or the status of the file that you are processing determine the file's disposition.

1. Although the CLOSE options FLASH, FCB, MODIFY, and CHARS may be specified for any virtual device except a reader, they only cause system action when a spool file with these attributes is printed on a 3800 printer.
2. A user may also cancel settings of the CHARS, MODIFY, or FCB options by specifying NULL in the name fields of these operands. In these cases, NULL performs the same function as OFF.
3. The keyword OFF, which resets attributes to their default values, may not also be used to specify a FORM, FLASH, FCB, MODIFY, DISTcode, or CHARS name. Similarly, NULL may not be used to specify a CHARS, MODIFY, or FCB name.
4. If you close an output device (printer, punch, console) and the file you created is in hold, that file will not be available for processing. If the file is not in hold, it will be available for processing. You can transfer a file to another user by using the TO operand on the CP TRANSFER or CHANGE commands.
5. You can specify a filename and filetype to help you identify a file and its contents. The filename and filetype you assign will have no effect on how CMS names the file if it is read using the CMS READCARD or CMS DISK LOAD commands.
6. If you are closing a reader, the file that is being processed will be purged unless HOLD or KEEP is in effect for the device or for the file. Use the appropriate CLOSE command option to override the effect of HOLD, NOHOLD, KEEP, or NOKEEP established for the reader or for the file.
7. If an input spool file has been only partially read (that is, your virtual machine did not get an end-of-file indication), you must issue a CLOSE READER command.
8. When a file changes ownership, as when it is closed to a new user, the spool file id number changes. The new number is obtained from the pool of available ids of the recipient of the file. Thus, if user SAM closes his reader to user HARRY, he would get the response:


```
RDR 0003 SENT TO HARRY RDR AS 0018
```

HARRY would now have a file #18.
9. The disposition of a spool file that is to be processed on a real unit record device depends upon the resolution of HOLD, CONT, and KEEP status as implied by the device using the SPOOL and CLOSE commands.

For example, suppose your virtual printer is spooled CONT. You issue the following command:

```
close prt class z
```

Since you did not specify the CLOSE operand EOF, the file will not become available for processing on a real printer.

CLOSE: Options

After completion of the CLOSE command, CP checks the HOLD status of the file to determine whether the files you closed can be processed. Thus, if HOLD is in effect for any of the closed files because you specified HOLD on the CLOSE command or because HOLD was set for the device using the CP SPOOL command, the file will not be processed.

For readers only, the net effect of file status, device options, and the CLOSE command overrides is in Table 7.

In all cases, the original KEEP/NOKEEP status of the file is unchanged by the CLOSE operation.

Likewise, the disposition of a spool file that is to be processed on a real unit record device depends upon the resolution of HOLD, CONT, and KEEP status as implied by the device using the SPOOL and CLOSE commands.

Options in Effect For the Input File	System Action
NOKEEP/NOHOLD	File is purged.
NOKEEP/HOLD	File is not purged and remains available for processing in HOLD status.
KEEP/NOHOLD	File is not purged. It will be set in NOHOLD/KEEP status and will not be available for processing.
KEEP/HOLD	File is not purged. It will be set in HOLD/KEEP status and will not be available for processing.

Response 1

The CLOSE command is used to terminate the spooling activity on any virtual unit record or console device. Below is the response issued when the system begins processing a spool file that has the MSG attribute.

```
{PRT} rdev {PRINTING} userid FILE spoolid RECS {nnnn} COPY [*]nnn SEQ sss
{PUN}      {PUNCHING}                                {nnnK}
                                                    {nnnM}
```

where:

PRT

PUN

indicates the type of device on which the file is being processed.

rdev

is the real device number.

userid

is the identification of the user whose file was active.

FILE spoolid

is the unique, system-assigned spoolid number by which VM/XA SP refers to the file.

RECS { $\left. \begin{array}{l} \text{nnnn} \\ \text{nnnK} \\ \text{nnnM} \end{array} \right\}$

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

is the remaining number of copies of the file to be created.

When the optional asterisk (*) is specified, the 3800 printer will print each page from the spooled output files nnn times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (*) is ignored and duplication occurs normally.

SEQ sss

is the sequence number for the file on the printer or punch.

Response 2

If you are using the TO operand with the CLOSE command, the following response will be sent to you and to the operator receiving the file (assuming the operator is active on the system):

Note: This response is not displayed if the SET IMSG OFF command has been issued.

$\left. \begin{array}{l} \text{RDR} \\ \text{PRT} \\ \text{PUN} \\ \text{CON} \end{array} \right\}$ FILE spd1 SENT $\left. \begin{array}{l} \text{TO} \\ \text{FROM} \end{array} \right\}$ userid $\left. \begin{array}{l} \text{RDR} \\ \text{PRT} \\ \text{PUN} \end{array} \right\}$ AS spd2 RECS $\left. \begin{array}{l} \text{nnnn} \\ \text{nnnK} \\ \text{nnnM} \end{array} \right\}$ COPY [*]nnn c $\left. \begin{array}{l} \text{HOLD} \\ \text{NOHOLD} \end{array} \right\}$ $\left. \begin{array}{l} \text{KEEP} \\ \text{NOKEEP} \end{array} \right\}$

where:

RDR
PRT
PUN
CON

indicates the spool file queue. If you are transferring a file to another user's virtual machine, this field indicates the queue from which the file originated. If you are receiving a file from another user, this field indicates the file queue in which the file has been added.

spd1

is the spool identification (spoolid) of the file as seen by the owner of the file.

TO userid

is the recipient of the spool file.

FROM userid

indicates from whom the file came. If the PURGE option is used with the CLOSE command, this field is replaced by **** PURGED **** to indicate that the file has been deleted.

RDR
PRT
PUN

indicates the spool file queue. If you are transferring a file to another user's virtual machine, this field indicates the queue in which the file has been added. If you are receiving a file from another user, this field indicates the file queue in which the file originated.

spd2

is the spool identification of the file when it is associated with the receiver of the file.

RECS $\left\{ \begin{array}{l} \text{nnnn} \\ \text{nnnK} \\ \text{nnnM} \end{array} \right\}$

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

specifies the number of file copies you want printed or punched when the file is produced.

When you specify the optional asterisk (*), the 3800 printer will print each page from the spooled output files nnn times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (*) is ignored and duplication occurs normally.

c

is the spool file class.

HOLD
NOHOLD

is the user HOLD status associated with the file.

KEEP
NOKEEP

is the KEEP status associated with the file.

Migration Notes

VM/SP HPO:

1. VM/XA SP truncates a name identification to 16 characters (if longer than 16).
2. The VM/XA SP response includes the number of records and KEEP setting.
3. If the file is closed to another user, the VM/XA SP response includes the new spoolid.
4. VM/XA SP supports additional options: TO/FROM, KEEP/NOKEEP and MSG/NOMSG.
5. VM/XA SP does not allow the specification of SYSTEM or userid.
6. VM/XA SP does not support the DEST and UNCONV operands.

7. VM/XA SP sets DIST to what is in the OPTIONS directory when * (asterisk) is specified.

VM/XA SF: If the file is closed to another user, the VM/XA SP response includes the new spoolid.

COMMANDS

Privilege Class: Any

Use the **COMMANDS** command to display the commands and **DIAGNOSE** codes which you are authorized to use.

COMMANDs	[cmd1 cmd2 cmd3 ...]
----------	----------------------

where:

cmd1 cmd2 cmd3 ...

are the names of specific commands to be checked for authorization. The response will show which default IBM privilege classes you have that correspond with a specified command.

Usage Notes

1. The same function is available via the **QUERY COMMANDS** command.
2. Some commands have more than one version, but the user may not be authorized to use all the versions of a particular command. If the system rejects a command that is listed in the response, it is probably because the user is not authorized to use the version that includes the operand or operands that were entered.
3. For an indication of which versions of a command you are authorized to issue, use "**COMMANDS cmd,**" where "cmd" is the name of the command or commands you are interested in. You will be shown the default IBM privilege classes of the command that are authorized to you.
4. You may specify any number of commands on the "**COMMANDS cmd...**" command. If a command name is entered twice, it will be displayed twice in the response. No checking is made for duplicate entries.

Response 1

```

comm1  comm2  comm3  comm4  comm5  comm6
comm7  .....
.
.
diag1  diag2  diag3  diag4  diag5  diag6
diag7  .....
```

indicates the commands and diagnose codes that the user is authorized to issue.

where:

commx

is the name of a command you are authorized to use.

diagx

identifies a **DIAGNOSE** code you are authorized to use.

Response 2

```
cmd      IBMCLASS={c1,c2,c3}
           {
           ANY
           NONE
           }
```

is the response when a user specifically enters commands to be checked for authorization.

where:

cmdx

is the name of the command you entered on the command line.

IBMCLASS = c1,c2,c3

are the IBM-defined privilege classes of this command.

ANY

indicates that the command is a class ANY command, and all users may issue the command.

NONE

indicates that you are not authorized to issue any version of the command.

Migration Notes*VM/SP HPO*

1. Although VM/XA SP does have the * (asterisk) command, it does not include the * (asterisk) in the list of commands.
2. VM/XA SP allows the specification of a particular command.
3. VM/XA SP allows commands up to 12 characters long (HPO only 8 characters long).
4. For VM/XA SP only, equivalent function exists for QUERY COMMANDS.

COUPLE
Privilege Class: G

Use the COUPLE command to connect two virtual channel-to-channel adapters (CTCA).

COUPLE	vdev1 [TO] userid vdev2
--------	-------------------------

where:

vdev1

is the virtual device number of your virtual CTCA.

[TO] userid

is the user identification of the virtual machine to which vdev1 is to be connected.

vdev2

is the virtual device number of the virtual CTCA that you want to connect to vdev1.

Usage Notes

1. The user you specify in the command line must be logged on and have a virtual channel-to-channel adapter defined for his or her virtual machine configuration. The virtual machine may be operating disconnected.
2. If you want to connect the CTCA you designated as vdev1 to another CTCA in your own virtual machine, you may specify userid as an asterisk (*).
3. If you omit the keyword TO from the command line, the userid you specify cannot be T or TO.
4. If a vdev is already connected to another vdev, that connection is broken.
5. The other user must have authorized the connection using the CP DEFINE CTCA command or the SPECIAL directory entry.

Responses

The originator of the command and the owner of the coupled-to CTCA (if different from the originator) receive messages indicating that the CTCAs have been coupled.

Response 1

CTCA vdev1 COUPLED TO userid vdev2

is the response sent to the originator of the COUPLE command.

where:

vdev1

is the device number of the originator's virtual CTCA.

userid

is the identification of the receiving virtual machine.

vdev2

is the device number of the virtual CTCA of the receiving user. This may also be a different channel-to-channel adapter in your own virtual machine.

Response 2

CTCA vdev2 COUPLED BY userid vdev1

is the response sent to the owner of the coupled-to virtual CTCA, if he is not the originator of the command.

where:

vdev2

is the device number of the virtual CTCA of the receiving user.

userid

is the identification of the user who issued the COUPLE command.

vdev1

is the device number of the originator's virtual CTCA.

Response 3

CTCA vdev1 DROPPED FROM userid vdev2

is the response that the originator receives if the virtual CTCA vdev1 was already coupled. This response is also sent to the owner of the vdev from which the connection was broken. The previous connection is terminated.

Response 1

where:

vdev1

is the virtual device number of the CTCA.

userid

is the user identification of the virtual machine to whose CTCA your CTCA had been coupled.

vdev2

is the virtual device number of the other virtual machine's CTCA.

Migration Notes

VM/SP HPO: The VM/XA SP response uses CTCA, COUPLED, and DROPPED instead of CTC, COUPLE, and DROP as in VM/SP HPO.

CP

Privilege Class: Any

Use the CP command as an optional preface to CP commands that you issue from the CP command environment.

CP	[command line1 [#command line2 #...]]
----	--

where:

command line1 [#command line2 #...]

is the name and operands for the CP command or commands you want to issue. You must precede the first command line with at least one blank.

Usage Notes

1. The pound sign (#) shown in the format block represents the logical line end symbol currently in effect for your virtual machine. If you have redefined the logical line end symbol, you must substitute your line end symbol for the pound sign when using this command. Before you issue the CP command, be sure that your virtual machine is operating with the CP SET LINEDIT function set ON. This is the default.
2. The CP command is treated as a null command by the control program and therefore can precede any other command if one or more blanks separate CP from the other command.
3. In the CP environment, you can enter multiple command lines as operands of the CP command, but you must separate each command line by the logical line end (#) symbol.

Responses

If you enter CP without a command line in the CP environment, there is no response and CP READ appears in the status area.

If you enter CP with command lines, you receive the responses appropriate to the individual commands you entered.

CPFORMAT

Privilege Class: G

Use the CPFORMAT command to format, label, and/or allocate DASD volumes for CP uses, such as paging, spooling, temporary disk, directory, nucleus, and minidisk space.

Note: To use CPFORMAT, your virtual machine must be a System/370 mode machine.

CPFORMAT	vdev	volid	[startcyl	[-endcyl]]
				-END		
				.ncyl		
				.END		
			Allocate			
			Label			

Note: The CMS command, CPFMTXA, may be found in the *VM/XA SP: CMS Command Reference*. The CPFMTXA command performs a function similar to CPFORMAT but does not reset the user's virtual machine.

where:

vdev

is the virtual device number of the DASD device to be formatted. It is specified as a 1- to 4-character hexadecimal number between X'0' and X'1FFF'. The device type associated with this vdev must be supported as a CP-owned volume and the user must have write access to the device.

volid

is the volume identification label of the virtual device to be formatted, labelled, and/or allocated. The volid is specified as a 1- to 6-character name (the characters do NOT have to be alphanumeric). The volid is written if startcyl is 0 or if you specify the "LABEL" option. Otherwise, the volid on the virtual device is read and compared to the one that you specify on the command line.

startcyl

is the starting cylinder that is to be formatted. It is a 1- to 4-digit decimal number; and must be within the size of the virtual device. If you do not specify startcyl, LABEL, or ALLOCATE, then 0-END is assumed. If the starting cylinder is all that is specified, then only the one cylinder specified is formatted.

-endcyl

-END

is the last of the range of cylinders that are to be formatted. If specified, endcyl is a 1- to 4-digit decimal number and must be within the size of the virtual device. If specified, endcyl must be equal to or greater than startcyl. Specify END to format all the cylinders on the virtual device, starting with the startcyl. If you do not specify an *endcyl* or END, the default is one cylinder.

.ncyl
.END

is a 1- to 4-digit decimal number designating the number of cylinders to be formatted. It must not be zero. The sum of startcyl and ncyl must not exceed the size of the virtual device. Specify END to format all cylinders on the virtual device, starting with the startcyl. If you do not specify an *ncyl* or END, the default is one cylinder.

ALLOCATE

indicates that you only want to reallocate a previously formatted CP volume.

LABEL

indicates that you only want to change the volume identification label. The volume must have already been formatted.

Note: If you do not specify any of the startcyl, LABEL, or ALLOCATE options, then you will, by default, format, allocate, and label the device. See Usage Note 8 on page 66 for more information.

Changing the Allocation Map

Whenever you format a disk, or specify the "ALLOCATE" option, the current allocation of the disk will be displayed as follows:

```
CURRENT ALLOCATION
TYPE CYLINDERS
.....
type nnnn-xxxx
type nnnn-xxxx
.
.
.
type nnnn-xxxx
```

where:

type

is the designated usage of the extent listed. The usage type is what was specified when the disk was last allocated (DRCT, PAGE, PERM, SPOL, or TDSK) or is BAD. BAD indicates an allocation type that CPFORMAT cannot process, which could result either from the pack not being formatted or from errors that overlay the allocation map. If "type" is BAD, the disk has not been formatted by CPFORMAT, and you should consider reformatting the entire disk, using CPFORMAT.

nnnn-xxxx

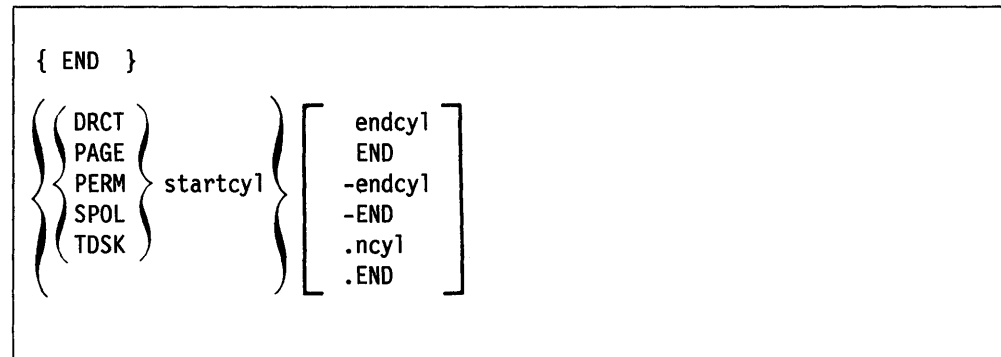
is a range of cylinders of this type.

After the current extents are displayed, you will be prompted to update the allocation map as follows:

```
ENTER ALLOCATION DATA
TYPE CYLINDERS
.....
```

For each cylinder range that you want to re-allocate, you enter one line of allocation data.

The format for allocation data is:



where:

END

is entered to indicate that you have completed entering your allocation data.

DRCT

indicates that the corresponding cylinders are to be allocated as space for directory files.

PAGE

indicates that the corresponding cylinders are to be allocated as space that will be used solely for system paging activity.

PERM

indicates that the corresponding cylinders are to be allocated as space for CP user minidisks, nucleus, checkpoint, warm start and other static space.

SPOL

indicates that the corresponding cylinders are to be allocated as space for spooling, dump space or overflow paging activity.

TDSK

indicates that the corresponding cylinders are to be allocated as temporary disk space, which will become available to users after they log on to the VM/XA SP system.

startcyl

is the starting cylinder that is to be allocated. It is a 1- to 4-digit, decimal number and must be within the size of the real device where the vdev is located. For example, if the vdev is a 3330 Model 1 (which has 404 cylinders), the starting cylinder must be any number from 0 to 403. If the starting cylinder is all that you specify, only that cylinder will be allocated.

endcyl

END

-endcyl

is the last of the range of cylinders whose allocation is specified. It is specified as a 1- to 4-digit decimal number. If specified, it must be equal to or greater than startcyl. It must be within the size of the real device where the vdev is located. For example, if the vdev is a 3330 Model 1 (which has 404 cylinders), the endcyl must be a number from 0 to 403. Specify "END" to allocate all cylinders as the specified type. For example, if you are allocating a 3330 Model 1 and have entered "SPOL 0-END", then cylinders 0-403 will be allocated as SPOL.

If you do not specify the dash (–) before endcyl or END, then you must leave at least one blank between startcyl and endcyl or END.

.ncyl

.END

is a 1- to 4-digit decimal integer designating the number of cylinders being allocated. If specified, ncyl must not be zero and the sum of startcyl and ncyl must not exceed the size of the real DASD upon which the specified vdev is located. For example, if the vdev is a 3330 Model 1 (which has 404 cylinders), and the startcyl was 0, the ncyl must be a number from 1 to 404. Specify END to allocate all cylinders as the specified type. If you do not specify ncyl or END, one cylinder is the default.

Usage Notes

1. In order to use CPFORMAT, your virtual machine must be a System/370 mode machine. Also, your virtual console address must be a 4-digit device address no higher than X'1FFF'.
2. Execution of the CPFORMAT command resets your virtual machine and clears virtual storage. This means that, after issuing the CPFORMAT command, you will have to re-IPL any system (such as CMS) that was running in your virtual machine.

The CMS command, CPFMTXA, performs the same function as the CP CPFORMAT command but does **not** reset your virtual machine.

3. The CPFORMAT command terminates all IUCV connections and releases the IUCV buffer.
4. CPFORMAT runs in your virtual machine. You can interrupt CPFORMAT processing by pressing the PA1 key.
5. A V=R or V=F user cannot invoke the CPFORMAT command processor.
6. You must format a volume using CPFORMAT or CPFMTXA prior to using the ALLOCATE option. Otherwise, an error message will be displayed.
7. You may use CPFORMAT to relabel a volume that was formatted by another operating system. Only the volid is changed, but the rest of the label record remains unchanged.
8. To prevent you from accidentally erasing a disk, CPFORMAT will prompt you indicating the cylinders that will be formatted, and ask you if you want to continue. If the cylinders are incorrect, you can reply "NO", and the disk will remain unchanged. If they are correct, you can reply "YES", and processing will continue.
9. If you issue the CPFORMAT command without the LABEL or ALLOCATE options, it is assumed that you want to format the DASD. For example, assume you enter the following command:

```
CPFORMAT 192 TEST1
```

CPFORMAT assumes that you want to format the entire device. All cylinders will be formatted, and all cylinders will be initially allocated as "PERM" space. Then, you will be prompted to update the DASD allocation, and the volume on 192 will be labelled "TEST1".

Assume, however, that you enter:

```
CPFORMAT 192 TEST2 0.3
```

CPFORMAT will then format cylinders 0, 1, and 2, and ALL cylinders will be initially allocated as "PERM" space. Then, you will be prompted to update the DASD allocation, and the volume on 192 will be labelled "TEST2".

If you enter:

```
CPFORMAT 192 TEST3 1-3
```

CPFORMAT will first check to see if the volid on 192 is "TEST3". If it isn't, you will receive an error message indicating what the correct label is, and CPFORMAT processing will terminate. If the volid is "TEST3", CPFORMAT will format cylinders 1, 2, and 3, and you will be prompted to update the DASD allocation. Note that in this case, the label is NOT written, and the cylinders are NOT initially allocated as "PERM" because cylinder 0 is not being formatted.

10. When allocating cylinders, you may enter the cylinders to be allocated in any order, using any combination of the valid allocation types of DRCT, PAGE, PERM, SPOL, and TDSK. Each entry in turn overlays the allocation record, and any space not reallocated will remain the same. If an allocation function overlays the previous cylinder allotment, then the previous cylinder space allotment is truncated to the beginning of the next cylinder allotment. For example, assume that the following is the current allocation of a vdev:

```
CURRENT ALLOCATION
TYPE CYLINDER
.....
PERM 0000-0554
```

and you enter the following allocation data:

```
TDSK 1-100
DRCT 100-200
SPOL 150-500
PAGE 400.50
END
```

The following allocation will result:

```
CURRENT ALLOCATION
TYPE CYLINDERS
PERM 0000-0000
TDSK 0001-0099
DRCT 0100-0149
SPOL 0150-0399
PAGE 0400-0449
SPOL 0450-0500
PERM 0501-0554
```

11. You can allocate a minidisk as if it were a real disk. For example, if you are allocating a 3350 minidisk, you can allocate 555 cylinders, even though your minidisk may not have 555 cylinders.
12. PERM is the only type of space that should be allocated beyond the extent of the virtual device.
13. Space for system dumps, named saved systems, saved segments, image library files, and UCR class system data files needs to be allocated as SPOL.
14. You should not allocate cylinder 0 as TDSK.

Response 1

FORMAT WILL ERASE CYLINDERS range ON DISK vdev
DO YOU WANT TO CONTINUE? (YES | NO)

where:

range

is inclusive, and specifies those cylinder numbers that will be formatted.

vdev

is the virtual DASD that will be formatted.

This prompt is issued whenever you format a disk. If you reply "YES", processing will continue, and the following response is issued:

FORMAT STARTED

If you reply "NO", you will receive messages indicating that the disk remains unchanged, and that CPFORMAT has been terminated.

Response 2

CURRENT ALLOCATION

TYPE CYLINDERS

.....

type nnnn-nnnn

type nnnn-nnnn

.

.

.

type nnnn-nnnn

ENTER ALLOCATION DATA

TYPE CYLINDERS

.....

where:

type

can be BAD, DRCT, PAGE, PERM, SPOL, or TDSK.

nnnn-nnnn

is the cylinder range for the corresponding type.

These responses are issued whenever you want to update the allocation map. The "CURRENT ALLOCATION" portion of the response displays how all cylinders on the virtual device are currently allocated. The "ENTER ALLOCATION DATA" portion of the response prompts you to update the allocation map. Enter your allocation data from the console; this data must contain both (1) the allocation type, and (2) the cylinders to be allocated as that type. When you enter END, the updated allocation map will be displayed on your console using the "CURRENT ALLOCATION" response.

Refer to the "Changing the Allocation Map" section in this command description for further details.

Response 3

REENTER: TYPE AND CYLINDERS OR 'END'

This prompt is issued whenever you make an error entering allocation data. An error message will precede this prompt explaining the error. This prompt asks you to correct and re-enter the line of allocation data that contained the error.

Response 4

FORMAT COMPLETE

This response is issued when all the specified cylinders have been formatted.

Response 5

ALLOCATE COMPLETE

This response is issued when the allocation record has been written on the DASD volume.

Response 6

VOLUME LABEL IS NOW 'XXXXXX'

This response is issued when the label record has been written on the DASD volume. 'XXXXXX' is the six-character valid with which the volume has been labelled.

Migration Notes

VM/SP HPO: In HPO, this function is provided by the stand-alone VM/370 Format/Allocate program (DMKFMT).

CPTRAP

VM/XA SP does not support the CPTRAP command. Equivalent function is provided by the following CP commands:

- TRSOURCE — Supports definitions and control of I/O, data, and guest tracing.
- TRSAVE — Specifies where CP trace data or data from traces defined by the TRSOURCE command are to be saved. CP trace data may be saved either in system trace files or on tape. Data from traces defined by the TRSOURCE command may be saved only in system trace files.
- SET CPTRACE — Activates or deactivates CPTRACE entry recording of real machine events in the CP TRACE tables.

In addition, the following commands may be helpful:

- QUERY TRSAVE — Displays the destination of traces defined by TRSOURCE or displays the current status of recording for CP system tracing (traces controlled by the SET CPTRACE command).
- QUERY TRSOURCE — Displays the current status of the various traces that have been defined using the TRSOURCE command.
- QUERY CPTRACE — Displays the current setting of the tracing of real system events in the CP trace tables.

Migration Notes

VM/SP HPO: Using any abbreviation for the QUERY CPTRAP command may result in the QUERY CPTRACE command being invoked. This results in syntax errors on that command.

CPU

Privilege Class: G

Use the CPU command to:

- Designate the virtual processor on which a specified CP command will be executed
- Select the virtual processor on which console function reads will be issued
- Choose the processor on which #CP commands will be executed.

CPU	{	<div style="border: 1px solid black; padding: 2px;"> cpuaddr1 cpuaddr1 cpuaddr... cpuaddr1-cpuaddrn ALL </div>	[CMD] command line	}
		cpuaddr2		

where:

cpuaddr1

is the processor address of the processor on which a CP command is to be executed.

cpuaddr1 cpuaddr...

is the list of addresses and/or ranges delimited by blanks.

cpuaddr1-cpuaddrn

specifies a nonwrapping pair of addresses separated by a hyphen (“-”) and not separated by any blanks.

ALL

specifies that the CP command executes on all virtual processors currently defined.

CMD

must be specified if VM/XA SP would interpret the first token of the command line as a hexadecimal address. For example, if you wish to specify that the DISPLAY PROGRAM STATUS WORDS command be executed on the CPU at address 1, do *not* specify:

```
CPU 1 D PSW.
```

VM/XA SP would interpret the “D” as an address. Instead, specify:

```
CPU 1 CMD D PSW
```

command line

is the CP command you want to execute on the specified virtual processors. When processing the CPU command, VM/XA SP does not check the validity of the CP command you enter here. Only the processor that receives the command checks it for validity and errors. CP commands that you enter as part of the CPU command are not checked for validity or errors.

cpuaddr2

is the processor address of the processor on which console function mode is to be entered and on which #CP commands are to be executed. Commands are processed on cpuaddr2 until another CPU command is issued specifying another cpuaddr.

CPU

Usage Notes

If a command line is specified, the command applies only to the virtual processors specified. If the `cpuaddr` is the only operand, all future CP commands will apply to the specified virtual processor until the next CPU command is issued.

Responses

XX:

where:

XX

is the processor in console function mode on which #CP commands are to be executed.

Migration Notes

VM/SP HPO: This command does not exist in HPO.

DEDICATE

Privilege Class: A

Use this command to dedicate (grant the sole usage of) a real processor to any virtual machine user's virtual CPU.

DEDICATE	[USER {userid } *]	[CPU cpuaddr] [CPU ALL]
----------	-------------------------	--------------------------------

where:

USER {userid }
 *

specifies the user whose virtual CPU is to be dedicated to a real processor. If the USER operand is specified, then either a userid or * must be specified. If the USER operand is not specified, then the V=R user is the default.

CPU cpuaddr

is the virtual CPU address of the virtual machine guest's virtual CPU to which a real processor is to be dedicated.

CPU ALL

will attempt to dedicate a real processor to every virtual CPU in the user's virtual configuration. If the CPU operand is not specified, CPU ALL is the default.

Usage Notes

1. The defaults for the DEDICATE command are CPU ALL and the userid of the V=R machine. An attempt will be made to dedicate a real processor to every virtual CPU defined in the V=R virtual configuration.
2. When DEDICATE USER userid CPU ALL is issued and there are more virtual CPUs defined than available real processors, all the real processors (except the master processor) are dedicated to those virtual CPUs starting at the lowest virtual CPU address.
3. If you have a virtual Vector Facility defined for a virtual CPU being dedicated, the system attempts to dedicate a real processor that has a Vector Facility associated with it. If another nondedicated processor having no Vector Facility associated with it is available, and the master processor has a Vector Facility, then this may result in the master processor being reassigned to that processor.

If a matching real processor cannot be found, the real processor most closely matching the characteristics of the virtual CPU being dedicated is assigned to that virtual CPU.

Note: If the last nondedicated Vector Facility is being taken by a user issuing a DEDICATE command, a message stating that no real Vector Facility is available to service nondedicated users is issued back to that user.

Any guest that has a virtual Vector Facility defined when the last real Vector Facility becomes unavailable subsequently receives a Vector Facility failure message if the guest issues any vector instructions. Guests that define virtual Vector Facilities after the last real Vector Facility has gone away receive a warning message.

DEDICATE

If a processor with a real Vector Facility is dedicated to a virtual CPU that has no virtual Vector Facility defined, the real Vector Facility will be unavailable for use by anyone until an UNDEDICATE is issued or a virtual Vector Facility is defined for the dedicated guest.

If “DEDICATE USER userid CPU ALL” is specified, all the currently defined virtual CPUs for the userid are affected. In addition, if automatic dedication is not in effect for this user and this user is V=R, then automatic dedication is enabled for this user. To avoid enabling automatic dedication, use “DEDICATE USER userid CPU nn” instead.

4. The CPU and USER operands can appear in any order.
5. If you issue the DEDICATE command to DEDICATE a processor to a V=R machine that has automatic dedication disabled, and the dedicated processor is subsequently undedicated for any reason (such as a CP abend), the dedicated processor is not automatically rededicated.

Responses

COMMAND COMPLETE

is the response issued upon the successful completion of a DEDICATE CPU command specifying a CPU address on the command line.

Migration Notes

VM/SP HPO: This command does not exist in VM/SP HPO.

DEFINE

		Class
DEFine	$\left. \begin{array}{l} 3800 \\ 3800-1 \\ 3800-3 \end{array} \right\} [AS] \text{ vdev [Option]}$ STORage [AS] $\left\{ \begin{array}{l} \text{nnnnnnnk} \\ \text{nnnM} \end{array} \right\}$ $\left. \begin{array}{l} T2305 \\ T2314 \\ T2319 \\ T3310 \\ T3330 \\ T3340 \\ T3350 \\ T3370 \\ T3375 \\ T3380 \\ TFB-512 \end{array} \right\} [AS] \text{ vdev [CYL] nnnn}$ vdev1 [AS] vdev2 VECTor $\left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{nn} \\ \text{nn1-nn2} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{nn} \\ \text{nn1-nn2} \end{array} \right\} \dots \right] \\ \text{ALL} \end{array} \right\}$	G
Option	$\left[[Size \ width \ length] \left[\begin{array}{l} 4Wcgm \\ 2Wcgm \end{array} \right] \left[\begin{array}{l} CFS \\ BTS \end{array} \right] \left[\begin{array}{l} NODatck \\ DATck \end{array} \right] \right]$	

General Usage Notes for DEFINE

1. When you use the DEFINE command to alter the configuration of your virtual machine, the changes are temporary and are only in effect for the current session.
2. If your virtual machine is operating in 370-XA mode, the virtual device numbers you specify can be no higher than FFFF. If your machine is operating in System/370 mode, you can define virtual device numbers through FFFF; however, your virtual machine can use only the virtual devices numbered 0 through 1FFF.
3. Informational responses are sent to your display to confirm that the configuration change you requested has taken place. These responses do not appear if you have issued the CP SET IMSG OFF command.
4. The temporary disks T2314, T2319, T3310, T3370, and TFB-512 are accepted for compatibility but will result in a message indicating that the temporary disk is not defined because space is not available.

Migration Notes

VM/SP HPO

1. VM/XA SP does not support the TIMER, CHANNELS, LINE, and SYSVIRT operands.
2. VM/XA SP does not allow the A abbreviation for AS.

DEFINE CONSOLE

Privilege Class: G

Use DEFINE CONSOLE to add a virtual system console to your virtual machine configuration.

DEFine	CONsole [AS] vdev	<u>3215</u> 3270
--------	-------------------	---------------------

where:

CONSOLE

indicates that you are defining a virtual system console for your virtual machine.

[AS] vdev

is the virtual device number of the console you are defining.

3215

3270

is the virtual console device type. 3215 is the default.

Usage Notes

1. If a terminal is controlled by a VTAM service machine with a level lower than 3.1.1., the console cannot be defined as a 3270. If you attempt to do so, the console is defined as a 3215 and an error message is displayed.
2. Only one virtual console is permitted. You will receive an error message if you already have a virtual console defined.
3. The default spool file class given to spool files created on the virtual console is class T.
4. Any time you issue the DEFINE CONSOLE command, you define the console mode for your display. This becomes the console mode for the duration of the session or until you reissue DEFINE CONSOLE or TERMINAL CONMODE. For more information on the console mode setting and its significance to the programs that you run in your virtual machine, see "TERMINAL CONMODE" on page 640.
5. For additional information, refer to the "General Usage Notes for DEFINE."

Responses

CONSOLE vdev DEFINED

confirms the virtual console definition. This response will not appear if you have issued the CP SET IMSG OFF command.

Migration Notes

VM/SP HPO

1. VM/XA SP supports an additional option: the virtual console device type (3215 or 3270).
2. VM/XA SP does not accept 'A' for 'AS'.

DEFINE CPU

Privilege Class: G

Use DEFINE CPU to add one or more virtual processors to your virtual machine configuration, or to change the address of a currently defined virtual processor.

DEFine	CPU { cpuaddr cpuaddr1 cpuaddr... cpuaddr1-cpuaddrn cpuaddr2 [AS] CPU cpuaddr3 }
--------	---

where:

CPU cpuaddr

defines a virtual processor for your virtual machine configuration at the address you specify for cpuaddr. You may specify multiple addresses in a list or in range form. If you specify a list of addresses, separate each address from the others by at least one blank. If you specify a range of addresses, specify the first and last addresses in the range and put a hyphen between them. The processor addresses can be any value from 0 to 3F; the initial address of your base processor either is specified in your entry in the system directory or defaults to 0.

CPU cpuaddr1 cpuaddr...

is a list of addresses and/or ranges delimited by blanks.

CPU cpuaddr1-cpuaddrn

specifies a nonwrapping pair of addresses separated by a hyphen (“-”) and not separated by any blanks.

CPU cpuaddr2 [AS] CPU cpuaddr3

changes the address of the processor currently defined at cpuaddr2 to the address cpuaddr3.

Usage Notes

1. Your entry in the system directory specifies the maximum number of processors you can define for your virtual machine.
2. If your entry in the system directory contains multiple CPU statements which specify a virtual multiprocessing configuration, then it will be created for you at logon time. If your directory entry does not specify additional processors, then your virtual machine logs on as a uniprocessor machine. Once logged on, you can use the DEFINE CPU command to modify your virtual configuration.
3. All future CP responses will be prefaced with the CPU number, with CPU issuing the responses. This continues as long as you have the virtual multiprocessor configuration.
4. When a virtual Vector Facility is associated with a virtual processor, a change of the virtual processor address (via “DEFINE CPU nn AS CPU mm”) also changes the address of the virtual Vector Facility.
5. If the V = R user issues the DEFINE CPU command and has automatic dedication enabled, the newly created CPU is dedicated to a real processor if one is available. If you do not want this CPU dedicated to a real processor: (a) use the DEFINE CPU command to get your new CPU and then use the UNDEDICATE command to release the dedication, or (b) disable automatic

dedication (by issuing UNDEDICATE CPU ALL beforehand). Note that UNDEDICATE is a privilege class A command.

If a V=R machine (with its OPTION directory statement set or defaulted to DEDICATE) issues DEFINE CPU, CP attempts to dedicate a real processor.

6. For additional information, refer to the "General Usage Notes for DEFINE."
7. The protected application environment is supported only for virtual machines consisting of a single processor. If you have previously established the protected application environment by issuing the SET CONCEAL ON command or by logging on with an OPTION CONCEAL directory statement, that environment is terminated when you define additional CPUs. If you revert to a uniprocessor configuration at a later time, you may reestablish the protected application environment by issuing SET CONCEAL ON.

Responses

00: CPU nn DEFINED

confirms that the processor has been defined.

where:

00

is the number of the CPU responding to you.

nn

is the number of the CPU currently defined.

DEFINE CHANNEL-TO-CHANNEL ADAPTER

Privilege Class: G

Use DEFINE CHANNEL-TO-CHANNEL ADAPTER (CTCA) to define a virtual channel-to-channel adapter for your virtual machine.

DEFine	CTCa [AS] vdev [USER] userid
--------	------------------------------

where:

CTCA

indicates that you are defining a channel-to-channel adapter for your virtual machine.

[AS] vdev

is the virtual device number of the CTCA you are defining.

[USER] userid

is the user that you will permit to connect to this CTCA using the CP COUPLE command.

Usage Notes

1. You cannot specify an asterisk in the command line to indicate your own userid. This is an exception to the rule which states that an asterisk is a substitute for your userid.
2. For more information, refer to the "General Usage Notes for DEFINE."

Responses

CTCA vdev DEFINED

confirms the definition of the virtual channel-to-channel adapter. This response will not appear if you have issued the CP SET IMSG OFF command.

Migration Notes

VM/SP HPO: On the VM/XA SP command line, you must specify the userid that you will permit to connect to the virtual CTCA.

DEFINE GRAF

Privilege Class: G

Use DEFINE GRAF to add a virtual display station to your virtual machine configuration.

DEFine	GRAF [AS] vdev [3270]
--------	-----------------------

where:

GRAF

indicates that you are defining a display station for your virtual machine.

[AS] vdev

is the virtual device number of the display you are defining.

3270

is the virtual device type of the display you are defining. This is the default.

Usage Notes

1. Once the display station is defined, you must issue the CP DIAL command from a real display station in order to logically connect the display station to your virtual machine. The device from which you issue the DIAL command must be supported by the operating system that is running in your virtual machine and by VM/XA SP.
2. For additional information, refer to the “General Usage Notes for DEFINE.”

Responses

GRAF vdev DEFINED

confirms the definition of the virtual display station. This response will not appear if you have issued the CP SET IMSG OFF command.

Migration Notes

VM/SP HPO: VM/XA SP does not support GRAF as vaddr (3036/3066/3138/3148/3158).

VM/XA SF: The VM/XA SP response uses GRAF instead of DISP.

DEFINE MDISK

Privilege Class: A (primary system operator only)

Use DEFINE MDISK to define a minidisk on a system volume or DASD volume attached to the system.

DEFine	MDISK [AS] vdev scyl {cnt} volser {END}
--------	--

where:

MDISK

indicates that you are designating an area as a minidisk on a system volume or a DASD volume attached to the system.

[AS] vdev

is the virtual device number of the minidisk you are defining.

scyl

is a 4-digit decimal starting cylinder number that specifies which cylinder on the real disk corresponds to cylinder 0 of the virtual disk.

cnt

is a 4-digit decimal number that specifies the number of cylinders that is to make up the MDISK definition.

END

specifies that the MDISK should be defined with the remaining available cylinders.

volser

is the volume serial (one to six alphanumeric characters) of the real DASD volume upon which the MDISK definition is to apply.

Usage Notes

1. The minidisk that you define remains available until you log off or detach the virtual device on which it resides. Use this command only in disaster recovery situations because it provides access to any area of a DASD volume without requiring additional authorization.
2. You can define a minidisk as a full-pack minidisk. You must define the starting cylinder number as zero and include all of a DASD volume's primary cylinders and some or all of its alternate cylinders.
3. Minidisks created using the DEFINE MDISK command have cache access if they are on a cached control unit.
4. Minidisks are not permitted on 3880 Model 11 and 21 subsystems.

Responses

DASD vdev DEFINED

where:

vdev

is the virtual device number of the minidisk you define.

DEFINE (spooling device)

Privilege Class: G

Use the DEFINE (spooling device) command to add a virtual spooling device to your virtual machine configuration.

DEFine	<pre> Reader Printer Punch 1403 2501 [AS] vdev 2540P 2540R 3203 3211 3262 3505 3525 4245 4248 </pre>
	<pre> (3800) [AS] vdev [Size width length] [2Wcgm] [BTS] [DATCK] (3800-1) [4Wcgm] [CFS] [NODATCK] (3800-3) </pre>

Note: If you specify a 3800-3 with 2WCGM, you will receive an error message.

where:

READER

RDR

specifies a 2540 card reader.

PRINTER

PRT

specifies a 1403 printer.

PUNCH

PCH

specifies a 2540 card punch.

1403

specifies a 1403 printer.

2501

specifies a 2501 card reader.

2540P

specifies a 2540 card punch.

2540R

specifies a 2540 card reader.

3203

specifies a 3203 printer.

DEFINE (spooling device)

3211
specifies a 3211 printer.

3262
specifies a 3262 printer.

3505
specifies a 3505 card reader.

3525
specifies a 3525 card punch.

4245
specifies a 4245 printer.

4248
specifies a 4248 printer.

3800
specifies a 3800 Model 1 printer.

3800-1
specifies a 3800 Model 1 printer.

3800-3
specifies a 3800 Model 3 printer in Model 1 compatibility mode.

[AS] vdev
is the virtual device number for the spooling device that you are defining.

SIZE width length

specifies the physical characteristics of the paper to be loaded into the 3800 printer. "width" indicates the hexadecimal width code of the paper. "length" indicates the decimal length of the paper. Specify "length" as a whole number using half-inches. If "width" and "length" are not specified, 14-7/8 x 11 inches is assumed.

The default value of "width" is X'0F'.

The default value of "length" is 22 (eleven inches).

The following is a list of available form width codes. All other values will be rejected.

CODE	WIDTH IN INCHES	WIDTH IN MILLIMETERS
01	6-1/2 in.	(165 mm ISO)
02	Reserved	(180 mm ISO)
04	8-1/2 in.	(215 mm ISO)
06	9-1/2 in.	(235 mm ISO)
07	9-7/8 in.	(250 mm ISO)
08	10-5/8 in.	(270 mm ISO)
09	11 in.	(280 mm ISO)
0A	12 in.	(305 mm ISO)
0B	Reserved	(322 mm ISO)
0D	13-5/8 in.	(340 mm ISO)
0E	14-3/10 in.	(363 mm ISO)
0F	14-7/8 in.	(378 mm ISO)

2WCGM

4WCGM

specifies the number of writable character generation modules (WCGM) assumed for the virtual 3800 printer. A WCGM is a 64-position portion of the 3800's character generation storage that holds the scan elements of one character set. A 3800 can have either two or four WCGMs. If neither is specified, a default of 4WCGM is assumed.

Note: Specifying 3800-3 with 2WCGM results in an error message.

BTS

CFS

designates the stacker for the virtual 3800 printer. You may specify either CFS (continuous forms stacker) or BTS (burster trimmer stacker). If neither is specified, CFS is assumed.

DATCK

NODATCK

specifies the desired handling of certain virtual 3800 data checks. If you specify DATCK, all 3800 data checks are reflected to the virtual machine (provided the 'BLOCK DATA CHECK' CCW has not been issued). If you specify NODATCK, only data checks that occur due to invalid translate table specifications or unmatched FCB codes are reflected to the virtual machine. NODATCK is the default condition.

Note: DATCK should be used only when absolutely necessary as it severely increases the overhead associated with simulation of WRITE and SKIP CCWs to the virtual 3800. In general, the reflection of data checks due to overprinting and invalid EBCDIC codes is not necessary.

Usage Notes

1. Avoid using device addresses that can be mistaken for device types (such as 1403, 2501, and 3203). When CP encounters a device address on the device command line that is the same as an IBM device type, CP assumes that you are specifying a device type and not a device address.
2. If you define a virtual 3211 printer, it supports the CP LOADVFCB command and the INDEX feature.
3. Default spool file classes given to spool files created on this device are: reader-class A, punch-class A, printer-class A.
4. Virtual printers, consoles, and punches are created with the default option "NOEOF."

Examples:

```
DEFINE Reader AS 00c
```

The parameters following the vdev specification may be given in any order and multiple times. However, the last specification of a multiply-defined parameter will be the one used. For example, if you issue:

```
DEF 3800 as 00A CFS DATCK SI 07 22 2W BTS
```

BTS is the value used instead of CFS because it was specified after CFS.

5. For more information, refer to the "General Usage Notes for DEFINE."

DEFINE (spooling device)

Responses

type vdev DEFINED

confirms that the device has been defined. This response will not appear if you have issued the CP SET IMSG OFF command. The possible values for type are:

Type	Meaning
RDR	Card reader
PRT	Printer
PUN	Card punch

Migration Notes

VM/SP HPO: VM/XA SP does not support DEFINE PRINTER As vaddr 1443/3289E.

DEFINE STORAGE

Privilege Class: G

Use the DEFINE STORAGE command to change the storage size of your virtual machine.

DEFine	STORage [AS] { nnnnnK } { nnnM }
--------	-------------------------------------

where:

STORAGE [AS] { nnnnnK }
 { nnnM }

indicates that you are changing the storage size of your virtual machine. You can specify the new storage size as nnnnnK (where K represents 1024 bytes) or nnnM (where M represents 1,048,576 bytes).

Usage Notes

1. Specify the new storage size as a multiple of 64K bytes. The new storage size is limited by the maximum size contained in your entry in the system directory. The minimum size you can specify is 64Kb and the maximum size is 999Mb.

If the storage size you specify is not a multiple of 1Mb and is not greater than 16Mb, it is rounded up to the next 64Kb boundary. If the storage size you specify is not a multiple of 1Mb and is greater than 16Mb, it is rounded up to the next 1Mb boundary. If the storage size you specify remains the same, you receive Response 1. No reset takes place if you try to define your storage to the same value you currently have.

The storage size specified is rounded up to the next higher megabyte boundary for both V=R and V=F users, even if the storage size specified is less than 16Mb.
2. The DEFINE STORAGE command causes the virtual machine to be reset and storage to be cleared. It also terminates all IUCV connections and releases the IUCV buffer. If the virtual machine was a member of a virtual machine group (from prior IPL of a VMGROUP NSS), that group membership is canceled. The virtual machine's IUCV connection to the signal system service is severed and a signal-out message is broadcast by the signal system service to all other members of the group who are enabled for signal-out notification.
3. If your virtual machine uses DIAGNOSE X'98' for locking virtual machine pages in real storage, issuing DEFINE STORAGE unlocks those pages. DEFINE STORAGE releases all NSSs and saved segments from your virtual machine. Locked shared pages are unlocked only by DIAGNOSE X'98' or by the release of the shared segment by the last user.
4. Whenever you change the virtual storage size, your virtual machine is reset as if a CP SYSTEM CLEAR command had been executed. Because a reset clears all virtual storage to binary zeros, you must repeat your IPL procedure to continue operation.
5. If the V=R or V=F storage amount requirement cannot be allocated within the V=R area, then this command is not processed and an error message is issued.
6. For more information, refer to "General Usage Notes for DEFINE" on page 76.

DEFINE STORAGE

Response 1

$$\text{STORAGE} = \left\{ \begin{array}{l} \text{nnnnnK} \\ \text{nnnM} \end{array} \right\}$$

confirms the new storage size of your virtual machine. This response does not appear on your display if you have issued the CP SET IMSG OFF command.

Response 2

STORAGE CLEARED - SYSTEM RESET

is issued whenever you change your storage size using DEFINE STORAGE.

Migration Notes

VM/SP HPO

1. VM/XA SP does not allow a V=R user to define a storage size larger than the V=R area;
2. VM/XA SP provides an additional response: STORAGE CLEARED - SYSTEM RESET;
3. VM/XA SP allows you to specify storage sizes larger than 16Mb.

DEFINE (temporary disk)

Privilege Class: G

Use DEFINE (temporary disk) to define temporary virtual disks.

DEFine	T2305 T2314 T2319 T3310 T3330 T3340 T3350 T3370 T3375 T3380 TFB-512	[AS] vdev [CYL] nnnn
--------	---	----------------------

where:

T2305

T3330

T3340

T3350

T3375

T3380

adds a temporary virtual disk to your virtual machine configuration. The four digits following the “T” correspond to the virtual device type.

T2314

T2319

T3310

T3370

TFB-512

are accepted for compatibility but will result in a message indicating that the temporary disk is not defined because space is not available.

[AS] vdev

specifies the virtual device number of the disk.

[CYL] nnnn

specifies the number of cylinders contained in the virtual disk.

Usage Notes

1. Temporary disk space is assigned from a pool of DASD resources; therefore, you should always format your temporary disk space before you use it.
2. Specify T3350 if a 3350 is used in native mode; specify T3330 if a 3350 is used in 3330 compatibility mode. Specify T3340 if a 3344 is used.
3. If you are defining a multiple-exposure device, the virtual device number that you specify must be the first (base) exposure. Once you define the multiple exposure device, all the exposures are defined.
4. TDISKs created via the DEFINE TDISK command will have cache access if they are on a cached control unit.
5. For more information, refer to the “General Usage Notes for DEFINE.”

DEFINE (temporary disk)

Responses

DASD vdev DEFINED

confirms the definition of the temporary disk. This response will not appear on your display if you have issued the CP SET IMSG OFF command.

Migration Notes

VM/SP HPO: VM/XA SP does not support DEFINE TFB-512/T3310/T3370 As vaddr BLK nnnnnn, or DEFINE T2314/T2319 As vaddr CYL nnn.

DEFINE (virtual device)

Privilege Class: G

Use DEFINE (virtual device) to change the virtual device number of any existing virtual device in your configuration.

DEFine	vdev1 [AS] vdev2
--------	------------------

where:

vdev1

is the virtual device number of an existing virtual device.

[AS] vdev2

is the new virtual device number.

Usage Notes:

1. If you are changing the virtual device number of a multiple-exposure device, specify the base exposure as the virtual device number in both vdev1 and vdev2. When you change the virtual device number of a multiple-exposure device, all exposures are redefined. Thus, if 1B0 is the device number of a 2305, the command:

```
define 1b0 as 1d0
```

redefines device numbers 1B0 through 1B7 as 1D0 through 1D7, respectively.

2. Avoid using device addresses that can be mistaken for device types (such as 1403, 2501, and 3203). When CP encounters a device address on the device command line that is the same as an IBM device type, CP assumes that you are specifying a device type and not a device address.

To avoid potential confusion and errors, use the REDEFINE command.

Note: If you redefine a System/370 device that is eligible for the start interpretive execution assist as a vdev with an address greater than FFF, you will lose the assist for that device.

3. Virtual paging subsystem devices cannot be redefined in order to protect the integrity of the user's virtual paging subsystem.
4. For more information, refer to the "General Usage Notes for DEFINE."

DEFINE (virtual device)

Responses

type vdev DEFINED

confirms that the device number has been changed. This response will not be displayed if you have issued the CP SET IMSG OFF command. The possible values for type are:

Type	Meaning
DASD	Direct access storage device
TAPE	Magnetic tape
RDR	Card reader
PRT	Printer
PUN	Card punch
GRAF	Display graphic device
CONS	Console
CTCA	Channel-to-channel adapter
CTLR	3704 or 3705 Communications Controller
MSC	Mass Storage Controller
DEV	Any other device

Migration Notes

VM/SP HPO

1. VM/XA SP does not support DEFINE 1443/3088/3289E As vaddr.
2. Within VM/XA SP, when CP encounters a device address on the device command line that is the same as an IBM device type, CP assumes that you are specifying a device type and not a device address.

VM/XA SF: VM/XA SP uses GRAF rather than DISP in its response.

DEFINE VECTOR

Privilege Class: G

Use DEFINE VECTOR to create a virtual Vector Facility (VF) on one or more CPUs of your virtual machine configuration. If VF use is not restricted by directory option, a virtual VF may be defined for each virtual CPU that has been defined. A virtual Vector Facility may be defined for each virtual CPU.

A virtual VF can only be defined if VM/XA SP is running on a machine with at least one real VF installed.

DEFine	VEctor $\left\{ \begin{array}{l} \left\{ \begin{array}{l} nn \\ nn1-nn2 \end{array} \right\} \left[\left\{ \begin{array}{l} nn \\ nn1-nn2 \end{array} \right\} \dots \right] \\ ALL \end{array} \right\}$
--------	--

where:

VECTOR

defines a virtual VF for the virtual machine configuration at the address or addresses specified. Multiple addresses may be specified in the form of a list of single numbers or ranges. The addresses may be any value from hex 0 to 3F, provided a CPU with the same address has been defined.

nn

is a single virtual CPU address in hexadecimal.

nn1-nn2

is a nonwrapping pair of addresses in hexadecimal separated by a hyphen and not separated by any blanks.

ALL

specifies that a virtual VF is to be defined for every virtual CPU in the configuration that does not already have one.

Usage Notes

1. If you define a new virtual Vector Facility for a virtual CPU that is currently dedicated to a real processor that doesn't have a Vector Facility installed, you will have to rededicate (UNDEDICATE and then DEDICATE) the virtual CPU. (The exception to this is if the processor is dedicated to a V = R user with automatic dedication set to on.)
2. When you redefine a virtual CPU with the DEFINE CPU *nn* AS *mm* command, you also cause the associated Vector Facility address to be redefined.
3. When a virtual VF is defined on a dedicated virtual CPU of the preferred (V = R) guest and automatic dedication is set on, VM/XA SP attempts to reassign that virtual CPU to a real processor with a real VF. If this attempt fails (because there is no nondedicated real processor with VF), then the virtual CPU remains dedicated to a processor without VF, and the virtual Vector Facility appears not operational to the guest. The system issues a warning message to this effect.

DEFINE VECTOR

4. When a user defines a virtual VF on a dedicated virtual CPU of a preferred (V=R) guest and automatic dedication is set on, this may leave no real VF for nondedicated virtual CPUs, including all V=V users. Therefore, this command should be used with care by the V=R guest.

Examples

To define a virtual VF for CPU 1, enter:

```
DEFINE VECTOR 1
```

To define virtual VFs for CPUs 0, 2, and 3, enter:

```
DEFINE VECTOR 0 2-3
```

Response 1

```
VECTOR FACILITY nn DEFINED
```

nn

is the virtual CPU address on which the virtual Vector Facility has been created.

Response 2

```
ALL VECTOR FACILITIES ALREADY DEFINED
```

is displayed when DEFINE VECTOR ALL is issued and all virtual CPUs already have virtual Vector Facilities defined on them.

DEFSEG
Privilege Class: E

Use the DEFSEG command to define a skeleton (class S) system data file (SDF) for the saved segment. The skeleton file consists only of a descriptor page, containing all the information necessary for keeping the saved segment when using the SAVESEG command. Users cannot access the saved segment until it has been saved.

DEFSEG	$dcssname \left\{ \begin{array}{l} \{hexpage1-hexpage2 \text{ type}\} \dots \\ \text{SAMErange} \end{array} \right\} \text{ [RSTD] [SPACE spacename]}$
--------	---

where:

dcssname

is the name of the saved segment (a 1- to 8-character alphanumeric string). When used with SPACE, the *dcssname* specified here is known as the member name and must be different from the SPACE name.

hexpage1-hexpage2

is a range of pages to be saved. When *hexpage1* and *hexpage2* are the same, only that page is saved. The range may be contained in a single 1-megabyte segment, or it may comprise a contiguous area in two or more segments. You may use multiple page ranges to define a 1-megabyte segment. However, the page descriptor codes of all page ranges in the same segment must have the same shared or exclusive attribute.

Define only those pages whose data must be saved. Any page or pages in a segment not explicitly defined in a page range on the DEFSEG command will default to the *no data saved* attribute. If other pages in the segment were defined with the shared attribute (page descriptor codes SR, SW, SN and SC), pages not specified on the command line become shared read-only pages whose data are not saved. No page descriptor code corresponds to this state; it has the same attributes as the state defined by page descriptor code SC, except that CP will not write into these pages.

If other pages in the segment were defined with the exclusive attribute (page descriptor codes ER, EW, and EN), any pages not specified on the command line become exclusive read/write pages whose data are not saved. This corresponds to page descriptor code EN.

The page ranges of a member saved segment cannot overlap with the specified page ranges of another member saved segment associated with the same segment space. Areas not specified (defaulted to *no data saved*) may be taken by another member saved segment. If the member uses these pages but doesn't need the pages' data saved, the range with the EN or SN attribute type must be specified. Also, this member saved segment's range descriptor code cannot conflict with those of an existing member that has ranges defined in the same segment of storage.

The page number is a hexadecimal value less than or equal to X'3E6FF'(999 Mb). The following list shows examples of how storage addresses translate into page numbers:

Hexadecimal Storage Address	Hexadecimal Page Number
00000xxx	0
00001xxx	1
00022xxx	22
00333xxx	333
04444xxx	4444
3E6FFxxx	3E6FF

type

refers to the page descriptor code of the page range in the saved segment. It indicates the type of virtual machine access permitted to pages in the range.

The first character is either "E" for exclusive access or "S" for shared access. Sharing of storage between virtual machines is based on one-megabyte segments, so all 256 pages in any one segment must be either exclusive or shared.

The second character of the page descriptor code defines the storage protection characteristics of the page. These character designators are as follows:

"R" indicates that page protection is used to make the page range read-only.

"W" means that the page is not protected, and users have read/write access.

"N" stands for "read/write, no data saved," meaning that the page is treated as a new page when the user references it. The contents of and storage keys for "no data saved" pages are not saved by the SAVESEG command.

"C" stands for "CP-writable, virtual machine read-only, no data saved," meaning virtual machines have read-only access to these pages, but CP services may change the data in the pages.

The contents of storage keys for "no data saved" pages are not saved by the SAVESEG command.

Valid page descriptor codes are:

```
EW -- exclusive read/write access
EN -- exclusive read/write access, no data saved
ER -- exclusive read-only access
SW -- shared read/write access
SN -- shared read/write access, no data saved
SR -- shared read-only access
SC -- CP writable pages, shared read-only access
      by virtual machine, no data saved
```

The user must define all pages in segment zero in ranges with an exclusive page descriptor code (EW, ER, or EN). The system will reject the DEFSEG command if the user defines any pages in segment zero with a shared access page descriptor code.

SAMERANGE

specifies that this member saved segment definition is the same as one specified by a previous DEFSEG. This operand is mutually exclusive with the *hexpage1-hexpage2* operands. When you specify SAMERANGE, you also must specify SPACE.

You don't have to specify SAVESEG if you use DEFSEG with the SAME operand for a class A member saved segment. However, if the member saved segment you're referencing is a class S file, you must still use SAVESEG to save the file.

SAMERANGE makes it easier to create overlaying segment spaces by using the same member for both segment spaces.

If you use this operand with a *spacename* that doesn't exist, the member that you specify must be a class S (skeleton) file.

You must have either a class S or class A file with the same *dcssname* when you use this operand. If you have both class A and class S files, the command uses the information in the class S file.

RSTD

indicates a restricted saved segment. The user must have a corresponding NAMESAVE directory statement to access this saved segment. If any member of a segment space is defined with the restricted operand (that is, the SPACE operand is used), the entire segment space is then restricted. This means that the restricted *spacename* is that which must be used on the NAMESAVE directory statement. If you didn't specify SPACE, then the *dcssname* is restricted. If you use the RSTD operand, make sure that you use the NAMESAVE directory statement to allow authorization.

If you used the SAMERANGE operand, you can use RSTD. This will affect the authorization of the segment space that you specified in the DEFSEG definition. The *spacename* is that which must be used on the NAMESAVE directory statement.

SPACE

specifies the definition of a segment space. Use this operand to make the *dcssname* that you are defining a member of this segment space.

spacename

is the name of the segment space (a 1- to 8-character alphanumeric string). You must specify this if you use the SPACE operand. The *spacename* must be different from any *dcssname* known to CP.

Usage Notes

1. Pages either defined as or defaulted to *no data saved* are initialized to a page of zeros when first referenced. If the pages are in an exclusive segment, changes will be seen only by the one user. If the pages are in a shared segment, any changes previously made by any sharing user (if allowed) will be visible to all other users sharing that same page.
2. Storage keys are not saved for pages either defined as or defaulted to *no data saved*. When first referenced, they have keys of zero.
3. A segment space has a range that may span one or more physical one megabyte segments. The beginning address of a segment space is rounded down to the nearest megabyte boundary of the member having the lowest page value. The ending address of a segment space is rounded up to the last address of the last

page within the megabyte occupied by the member having the highest page value.

4. Each member saved segment has a range that may span one or more pages. The beginning address of a member saved segment is determined by its lowest page value. The ending address is determined by the last address of the highest page value.
5. Segment zero must always have the exclusive access attribute "Ex." If a shared ("Sx") page descriptor code is used for any pages in segment zero, the command is rejected.
6. A CP DCSS (SC mode) contains CP-writeable pages, to which virtual machine users have just read-only access. CP will write data into these pages, which may then be referenced by virtual machines. Serialization of data in the CP-writeable pages, if desired, must be arranged between the CP function providing the data and the virtual machines referencing it.
7. A CP DCSS may not be loaded in non-shared (exclusive) mode by DIAGNOSE X'64'.
8. Saved segments are loaded via DIAGNOSE code X'64'. Named saved systems (NSS) are loaded via the IPL command.
9. Only one skeleton (class S) DCSS can exist with a specific name. If one already exists and a second DEFSEG command is issued specifying the same name, the command is rejected.
10. Each name may be used for only one of the following: a segment space, a member saved segment, or a DCSS.
11. Defining members which use undefined areas within other members of the same segment space is not recommended. However, in support of multiple products requiring both shared and exclusive segment types and to minimize storage use, definition of members with interleaved ranges is supported.
12. You can only use the SAMERANGE operand with the SPACE operand.
Note: You *cannot* use the SAME operand with a *dcssname* containing SW, SN, EW, or EN page types.
13. A DEFSEG with a SAMERANGE operand can be the first definition of a segment space only if the specified member is a class S file.
14. A DEFSEG with a SAMERANGE operand doesn't need a corresponding SAVESEG if the member saved segment is already saved. If the member saved segment isn't saved, you need to use the SAVESEG command.
15. A member of a segment space cannot overlap any of the ranges specified for an existing member within the same segment space. Also, the members cannot have conflicting descriptor codes for ranges defined within the same segment of storage.
16. You cannot specify a DEFSEG with the SAMERANGE operand for a segment space that already has a member saved segment with that name.
17. Putting a member name in a directory statement will have no effect; use the segment space name.

Security Notes

1. When RSTD is specified, (indicating a restricted saved segment) only those users whose directory entries contain a NAMESAVE statement specifying the name of the restricted system data file are allowed to load this saved segment. If RSTD was specified **without** the SPACE operand, the directory entries must contain a NAMESAVE statement specifying the *dcssname*. If RSTD was specified **with** the SPACE operand, the directory entries must contain a NAMESAVE statement specifying the *spacename*. This allows an installation to ensure data integrity by restricting access to saved segments that contain shared writable storage (defined with the SW and SN page descriptor codes) or to CP DCSSs (defined with the SC page descriptor code), which may contain sensitive data.
2. Virtual machines which share read/write pages, defined with the SW and SN page descriptor codes, are responsible for the integrity of the shared data. The operating system that uses these shared writable pages should ensure that these virtual machines are properly protected from one another.

Examples

```

DEFSEG USERSEG 100-1FF SR RSTD

DEFSEG SEG1 1000-1FFF EN 2000-2FFF SR 3000-3FFF ER

DEFSEG PPA 700-721 SR SPACE GROUP1

DEFSEG PPB 722-7A1 SR SPACE GROUP1

DEFSEG PPC 7A2-7F0 SR SPACE GROUP1

DEFSEG PPX 700-721 SR RSTD SPACE GROUP1A

DEFSEG PPB SAME SPACE GROUP1A

DEFSEG PPC SAME SPACE GROUP1A

DEFSEG PPD 700-740 SR SPACE GROUP3

DEFSEG PPE 741-779 SR SPACE GROUP3

DEFSEG PPE 7A0-7FF SR SPACE GROUP3

```

For more explanations and examples, please refer to *VM/XA SP Planning and Administration*.

Responses

None.

Migration Notes

VM/SP HPO

1. In HPO, this function is performed at system generation time by the NAMESYS entries in DMKSNT.
2. VM/XA SP allows this function to be done dynamically, at any time during system operation, so that an existing saved segment can be changed without regenerating the system.

VM/XA SF

1. The VM/XA SP DEFSEG syntax (without the SPACE operand) defines a DCSS that has no members. This DCSS contains the definition of the saved segment. When this is loaded (via DIAGNOSE X'64'), the range information is as follows:
 - The beginning address is equal to the lowest page value specified on the DEFSEG command.
 - The ending address is rounded up to the last address of the last page in the megabyte related to the highest page specified on the DEFSEG command.
2. If the same definition was specified with a DEFSEG command using the SPACE operand, then loaded (via DIAGNOSE X'64'), the range information is as follows:
 - The beginning address is equal to the lowest page value specified on the DEFSEG command.
 - The ending address equal to the highest page value specified on the DEFSEG command.
3. VM/XA SP supports a CP writable DCSS. This DCSS has restrictions against multiple copies. Saved segments not defined with an SC page range are unaffected, except that you can't define both a CP and non-CP DCSS with the same name.
4. VM/XA SP keeps saved segments over a cold start.

DEFSYS

Privilege Class: E

Use the DEFSYS command to define a skeleton (class S) system data file (SDF) for the named saved system (NSS). The skeleton file consists only of a descriptor page, containing all the information necessary for saving the NSS when using the SAVESYS command. Users cannot access the NSS until it has been saved.

DEFSYS	<pre>name {{hexpage1-hexpage2 type}...} MINSIZE={nnnnnK} [options] {nnnM } options: [RSTD] [PARMREGS={m[-n]}] [VMGROUP] { NONE }</pre>
--------	---

where:

name

is the name of the NSS that is a 1- to 8-character alphanumeric string.

Be careful not to specify a name that could be a hexadecimal device number because the device, if one exists, would be IPLed instead of the named saved system.

hexpage1-hexpage2

is a range of pages to be saved. When “hexpage1” and “hexpage2” are the same, only that page is saved. The range may be contained in a single 1Mb segment, or it may comprise a contiguous area in two or more segments. You may use multiple page ranges to define a 1Mb segment. However, the page descriptor codes of all page ranges in the same segment must have the same shared or exclusive attribute. (See the *type* parameter for information about page descriptor codes.)

Define only those pages whose data must be saved. Any page or pages in a segment not explicitly defined in a page range on the DEFSYS command defaults to the “no data saved” attribute. If other pages in the segment were defined with the shared attribute (page descriptor codes SR, SW, SN and SC), pages not specified on the command line become shared read-only pages whose data is not saved. No page descriptor codes correspond to this state; it has the same attributes as the state defined by page descriptor code SC, except that CP does not write into these pages.

If other pages in the segment were defined with the exclusive attribute (page descriptor codes ER, EW and EN), any pages not specified on the command line become exclusive read/write pages whose data is not saved. This corresponds to page descriptor code EN.

The page number is a hexadecimal value less than or equal to X'3E6FF' (999 Mb). The following list shows examples of how storage addresses translate into page numbers:

Hexadecimal Storage Address	Hexadecimal Page Number
00000xxx	0
00001xxx	1
00022xxx	22
00333xxx	333
04444xxx	4444
3E6FFxxx	3E6FF

type

refers to the 2-character page descriptor code of the page range in the NSS. It indicates the type of virtual machine access permitted to pages in the range.

The first character is either "E" for exclusive access or "S" for shared access. Sharing of storage between virtual machines is based on one-megabyte segments, so all 256 pages in any one segment must be either exclusive or shared.

The second character of the page descriptor code defines the storage protection characteristics of the page. These character designators are as follows:

"R" indicates that page protection is used to make the page range read-only.

"W" means that the page is not protected, and users have read/write access.

"N" stands for "read/write, no data saved," meaning that the page is treated as a new page when the user references it. The contents of and storage keys for "no data saved" pages are not saved by the SAVESYS command.

"C" stands for "CP-writable, virtual machine read-only, no data saved," meaning virtual machines have read-only access to these pages, but CP services may change the data in the pages.

The contents of and storage keys for "no data saved" pages are not saved by the SAVESYS command.

Valid page descriptor codes are:

- EW -- exclusive read/write access
- EN -- exclusive read/write access, no data saved
- ER -- exclusive read-only access
- SW -- shared read/write access
- SN -- shared read/write access, no data saved
- SR -- shared read-only access
- SC -- CP writable pages, shared read-only access
by virtual machine, no data saved

Any page ranges defined in segment zero must have an exclusive page descriptor code (EW, ER, or EN). If any pages in segment zero are defined with a shared access page descriptor code, the command is rejected.

$$\text{MINSIZE} = \left\{ \begin{array}{l} \text{nnnnnK} \\ \text{nnnM} \end{array} \right\}$$

indicates the minimum storage size of the virtual machine into which the NSS can be loaded.

nnnnnK is the storage size in decimal kilobytes

nnnM is the storage size in decimal megabytes

For example, MINSIZE = 256K.

RSTD

indicates a restricted NSS. The user must have a corresponding NAMESAVE directory statement to access this NSS.

PARMREGS = $\left\{ \begin{array}{l} m[-n] \\ \text{NONE} \end{array} \right\}$

specifies a range of general purpose registers to contain the IPL parameter string when the NSS receives control from CP.

m and n are decimal values from 0 to 15. If you specify two registers, n must be greater than or equal to m. If only one register is required, you may specify "PARMREGS = m".

NONE specifies that no parameters may be passed to this NSS.

VMGROUP

indicates that a user who IPLs this NSS becomes a member of a virtual machine group, identified by the name of the NSS.

Usage Notes

1. Page 0 should always be saved for an operating system. If not saved, certain information (such as the interval timer value for System/370 guests) may be lost. IBM recommends that page 0 be saved with the exclusive write (EW) page descriptor code.
2. Segment 0 should always have the exclusive access attribute (EW, EN, or ER). If a shared page descriptor code (SW, SN, SR, or SC) is used for any pages in segment 0, the command is rejected.
3. Pages either defined as or defaulted to "no data saved" are initialized to a page of zeros when first referenced. If the pages are in an exclusive segment, changes are seen only by the one user. If the pages are in a shared segment, any changes previously made by any sharing user are visible to all other users sharing that same page.
4. A CP NSS (SC mode) contains CP-writeable pages, to which virtual machine users have just read-only access. CP writes data into these pages, which may then be referenced by virtual machines. Serialization of data in the CP-writeable pages, if desired, must be arranged between the CP function providing the data and the virtual machines referencing it.
5. The use of the PARMREGS values is described with the IPL command.
6. Named saved systems (NSS) are loaded using the IPL command. Saved segments are loaded using DIAGNOSE X'64', the CMS SEGMENT assembler language macro, or the CMS SEGMENT command.
7. When the user references any pages either defined as or defaulted to "no data saved," the user gets a new page. A new page is assigned as a full frame of zeros just as any page that does not have a copy backed on auxiliary storage.
8. Storage keys are not saved for pages either defined as or defaulted to "no data saved." When these pages are first referenced, they have keys of zero. The virtual machine may set the keys for writable storage appropriately.
9. Each segment has an associated address range of 1Mb, regardless of the number of pages specified.

Security Notes

1. When RSTD is specified, only those users whose directory entries contain a NAMESAVE statement specifying this "name" are allowed to IPL the NSS. This allows the installation to ensure data integrity by restricting access to NSSs that contain shared writable storage (defined with the SW and SN page descriptor codes), or to a CP NSS (defined with the "SC" page descriptor code), which may contain sensitive data.
2. Virtual machines which share read/write pages, defined with the SW and SN page descriptor codes, are responsible for the integrity of the shared data. The operating system that uses these shared writable pages should ensure that these virtual machines are properly protected from one another.

Example

```
DEFSYS CMS 0-4 EW E-21 EW 800-86F SR MINSIZE=256K PARMREGS=0-15
DEFSYS WSS 0-8 EW 300-5FF SW 600-FFF SN MINSIZE=4M RSTD
DEFSYS GCS 0-8 EW 300-5FF SW 600-FFF SN MINSIZE=4M RSTD VMGROUP
```

Responses

None.

Migration Notes

VM/SP HPO: In VM/SP HPO, NSSs are defined at system generation time by the NAMESYS entries in DMKSNT. VM/XA SP allows this function to be done dynamically at any time during system operation so that an existing NSS can be changed without regenerating the system.

VM/XA SF: VM/XA SP has a CP writable NSS. This has restrictions against multiple copies. NSSs not defined with an SC page range are unaffected, except that you can't define both a CP and non-CP NSS with the same name.

DETACH

Use the DETACH command to:

- Remove real or logical devices from your virtual machine configuration or from the CP system.
- Remove virtual devices, virtual processors, real Expanded Storage, or a virtual Vector Facility from your virtual machine configuration.

The format box below shows all the operands available with the DETACH command. Complete descriptions of the operands appear on the following pages, in the same sequence as the operands appear here.

		Class
DETach	CPU $\left\{ \begin{array}{l} \text{cpuaddr1} \\ \text{cpuaddr1 cpuaddr...} \\ \text{cpuaddr1-cpuaddrn} \\ \text{ALL} \end{array} \right\}$	G
	$\left\{ \begin{array}{l} \text{rdev} \\ \text{rdev ...} \\ \text{rdev-rdev} \end{array} \right\} \quad [\text{FRom}] \quad \left\{ \begin{array}{l} \text{userid} \\ \text{SYSTEM} \\ * \end{array} \right\}$	B
	$\left\{ \begin{array}{l} \text{ldev} \\ \text{ldev ...} \\ \text{rdev ldev ...} \\ \text{ldev-ldev} \end{array} \right\} \quad [\text{FRom}] \quad \left\{ \begin{array}{l} \text{userid} \\ * \end{array} \right\}$	
	{Virtual} $\left\{ \begin{array}{l} \text{vdev} \\ \text{vdev ...} \\ \text{vdev-vdev} \end{array} \right\}$	G
	VECTor $\left\{ \begin{array}{l} \{ \text{nn} \} \left[\{ \text{nn1-nn2} \} \dots \right] \\ \{ \text{nn1-nn2} \} \left[\{ \text{nn1-nn2} \} \dots \right] \\ \text{ALL} \end{array} \right\}$	G
	$\left\{ \begin{array}{l} \text{XSTore} \{ [\text{FRom}] \text{userid} \} \\ \text{XStorage} \end{array} \right\}$	B
	$\left\{ \begin{array}{l} \text{XSTore} \\ \text{XStorage} \end{array} \right\}$	G

General Usage Notes for DETACH

You cannot be certain that the device or Expanded Storage capability has been detached until you receive the appropriate response from the system. If after a reasonable amount of time you haven't received the expected response, issue the QUERY command to determine the status of the device. Detachment may be pending or may have been unsuccessful. Note that if you issued the command from the CMS environment and received a return code of 0, you cannot assume that the device has been detached. A return code of 0 indicates only that your request has been accepted.

DETACH

Migration Notes

VM/SP HPO: The LEAVE and UNLOAD options are not supported by VM/XA SP.

DETACH CPU

Privilege Class: G

Use DETACH CPU to remove one or more virtual processors from your virtual machine configuration.

DETach	CPU { cpuaddr cpuaddr1 cpuaddr... cpuaddr1-cpuaddrn ALL }
--------	--

where:

CPU cpuaddr

is the processor address of the processor that you want to detach from your virtual machine configuration.

CPU cpuaddr1 cpuaddr...

is a list of addresses and/or ranges delimited by spaces.

CPU cpuaddr1-cpuaddrn

specifies a nonwrapping pair of addresses separated by a hyphen ("-") and not separated by any blanks.

CPU ALL

specifies all virtual processors currently defined.

Usage Notes

1. You can detach any virtual processor from your virtual machine configuration except the base processor. You cannot detach the base processor from your virtual machine system.
2. Whenever you detach a virtual processor, your virtual machine is reset as if a CP SYSTEM CLEAR command had been executed. All virtual storage is cleared to binary zeros. Therefore, you must repeat your IPL procedure to proceed.
3. When a virtual processor is detached, the associated virtual Vector Facility, if any, is also detached.
4. The DETACH CPU command terminates all IUCV connections and releases the IUCV buffer.

Responses

CPU nn DETACHED

indicates that the command has been successfully executed.

This response is followed by:

STORAGE CLEARED - SYSTEM RESET

indicating that all of your virtual machine's storage has been cleared to binary zeros and your processor(s) and I/O configuration have been reset.

DETACH (real device)

Privilege Class: B

Use DETACH (real device) to remove real local or remote devices from a user's virtual machine or from the CP system. When detaching a remote device, specify its logical device address.

DETach	$\left\{ \begin{array}{l} \text{rdev} \\ \text{rdev} \dots \\ \text{rdev-rdev} \end{array} \right\}$	[FRom]	$\left\{ \begin{array}{l} \text{userid} \\ \text{SYSTEM} \\ * \\ _ \end{array} \right\}$
	$\left\{ \begin{array}{l} \text{ldev} \\ \text{ldev} \dots \\ \text{ldev-ldev} \\ \text{rdev ldev} \dots \end{array} \right\}$	[FRom]	$\left\{ \begin{array}{l} \text{userid} \\ * \\ _ \end{array} \right\}$

where:

rdev
rdev ...
rdev-rdev

is a real device number, a list of real device numbers, or a range of real device numbers to be detached from the indicated user or from the system. On a single command line, you may combine device numbers, lists of device numbers, and ranges of device numbers. Specify a real device number as four hexadecimal digits.

ldev
ldev ...
ldev-ldev

is a logical device number, a list of logical device numbers, or a range of logical device numbers to be detached from the indicated user or from the system. On a single command line, you may combine device numbers, lists of device numbers, and ranges of device numbers. Specify a logical device number as an "L" followed by four hexadecimal digits.

rdev ldev

specifies a list of logical and real device numbers.

[FROM] $\left\{ \begin{array}{l} \text{userid} \\ \text{SYSTEM} \\ * \end{array} \right\}$

is the user identification of the virtual machine to which the device is currently attached. The specified user must be logged on, not in the process of logging on. If you are issuing the DETACH (real device) command to detach a device from your own virtual machine, you may specify userid as an asterisk (*). Specify SYSTEM to indicate that the device is a DASD device used by CP for allocation and control.

Usage Notes

1. CP does not REWIND/UNLOAD a tape when the tape is detached.
2. When a device is detached, all data blocks associated with the device are removed from the minidisk cache.

3. If you are detaching a multiple-exposure device, the device number that you specify must be the base exposure. When you detach a dedicated multiple-exposure device from a user, all exposures are processed.
4. If you specify multiple device numbers or a range of device numbers on the DETACH (real device) command line, you will receive an affirmative response for each device that is successfully detached and an error message for each device that cannot be detached.
5. If you attempt to detach an active shared device from the system, you will receive an error message indicating that the device is being shared by multiple users. In order to detach the device, each of the affected users must detach the device from their virtual systems.
6. You cannot detach volumes in the SYSCPVOL list that are in use by the system, or devices containing minidisks to which users are linked.
7. Tape devices are automatically rewound and unloaded when detached.
8. A previously attached device can be detached from a user even if the device is still in use.
9. If a device was attached using the CP ATTACH command and later detached, the system operator receives a message saying that the device is free.
10. The DETACH (real device) command does not execute synchronously when you are detaching real devices from another userid. The user of this command may receive control before all processing has been completed on behalf of the other user. Any later attempt to use the device before it becomes detached may yield unexpected results.

For example, suppose the following CMS command sequence exists in a CMS EXEC procedure:

```
CP DETACH 250 USERA  
CP ATTACH 250 SYSTEM
```

CP will begin to ATTACH device 250 to the system before the DETACH command completes. Since the device is still attached to USERA, you will receive an error message for the ATTACH command.

Thus, if you intend to issue commands using the same real device owned by another user, do not string the commands together (using the logical line end character) and do not issue the commands one right after the other in a CMS EXEC.

Responses

The user who currently owns the device (if different from the primary system operator), the user who issued the DETACH (real device) command (if different from the owner of the device), and the primary system operator (if different from the user who issued the command) each receive messages indicating that the DETACH (real device) was successful.

Response 1

```
type vdev DETACHED BY originator
```

is the response sent to the device owner (if the device owner is not the primary system operator). Note that the response refers only to the virtual device number (vdev) because this is the number by which the owner knows the device.

DETACH (real device)

Response 2

type {rdev} DETACHED {userid vdev}
 {ldev} {SYSTEM }

is the response issued to the originator of the DETACH (real device) command if the originator of the command is not the owner of the device.

Response 3

type {rdev} DETACHED {userid vdev} BY originator
 {ldev} {SYSTEM }

is the response sent to the primary system operator if another user (the originator) issued the DETACH (real device) command.

Response 4

DASD vdev vdev vdev vdev DETACHED BY originator

The user receives this response when a 3880 paging subsystem is detached by a class B user. The virtual base address and virtual exposures of the detached device are shown.

Response 5

DASD {rdev rdev rdev rdev} DETACHED {userid} vdev vdev vdev vdev
 {ldev ldev ldev ldev} {SYSTEM}

The originator receives this response when a 3880 paging subsystem is detached if the originator is not the owner of the device. The virtual base and virtual exposures of the detached device are shown.

Response 6

DASD {rdev rdev rdev rdev} DETACHED {userid} vdev vdev vdev vdev BY originator
 {ldev ldev ldev ldev} {SYSTEM}

The primary system operator receives this response when a 3880 paging subsystem is detached if another user issued the DETACH (real device) command. The virtual base and virtual exposures of the detached device are shown.

In these responses, the term "type" is one of the following:

Type	Meaning
DASD	Direct access storage device
TAPE	Magnetic tape
RDR	Card reader
PRT	Printer
PUN	Card punch
GRAF	Display device
CTCA	Channel-to-channel adapter
CTLR	3704 or 3705 Communications Controller
MSC	Mass Storage Controller
DEV	Any other device

Migration Notes

VM/SP HPO

1. VM/XA SP supports neither the CHANNEL operand nor two options for tape devices: UNLOAD and LEAVE.
2. In VM/XA SP, if a person issues a DETACH command on behalf of another user, the detachment does not take place immediately (it is asynchronous). The detachment occurs just prior to the other user being dispatched.
3. VM/XA SP provides a response for each detached device.
4. The VM/XA SP response includes the user's virtual device address.
5. VM/XA SP limits the range of devices detached to 256.
6. VM/XA SP accepts FR as the minimum truncation for FROM.

VM/XA SF: VM/XA SP uses GRAF rather than DISP in its response.

DETACH VIRTUAL (devices)

Privilege Class: G

Use DETACH virtual (devices) to remove virtual devices from your virtual machine configuration.

DETach	[Virtual] { vdev vdev ... vdev-vdev }
--------	--

where:

VIRTUAL

indicates you are detaching virtual devices

vdev

vdev ...

vdev-vdev

is a virtual device number, a list of virtual device numbers, or a range of virtual device numbers to be detached from your virtual machine. On a single command line, you may combine device numbers, lists of device numbers, and ranges of device numbers. If you are detaching a list of virtual devices, insert at least one blank between the device numbers you specify. If you are detaching a range of devices, insert a hyphen (-) between the device numbers you specify. The maximum number of devices allowed in a range is 256.

Usage Notes

1. You can detach a virtual device even if it is currently in use.
2. When you detach a virtual device, it becomes inaccessible to your virtual machine. When you detach a real device, it is released and becomes available for attaching to your virtual machine, another user, or the CP system.
3. Tape devices are automatically rewound and unloaded when detached.
4. If you detach a dedicated device, the system operator will receive a message indicating that the device is free.
5. If you are detaching a dedicated multi-exposure device, specify the base exposure as the virtual device number.

If you are detaching a multi-exposure device and have specified a range of virtual device numbers, all of the exposures will be detached when the base exposure is encountered. The range that you specified may not include all of the exposures; this is not an error situation. All of the exposures will be detached. The DETACH VIRTUAL command uses only the base exposure device number when you specify multiple virtual device numbers on the command line. Any device number other than the base exposure will be reported as an error.

6. Command processing does not terminate if an error (other than a syntax error) is encountered when processing a range of device numbers or multiple device numbers. Instead, an error message is issued for each device that cannot be detached. Informational messages are issued to identify those devices that were successfully detached. These informational messages may appear several times in one command execution when you have specified on the command line a range of device numbers or multiple device numbers.

Response 1

type vdev DETACHED

is the response you receive for each device that was successfully detached (if you are not the primary system operator).

Response 2

type rdev DETACHED userid vdev

is the response sent to the system operator each time a real device is detached from the specified user. rdev is the real device number associated with the device.

Response 3

CTCA vdev DROPPED FROM userid vdev

is the response if the device detached was a virtual CTCA connected using the COUPLE command.

where:

CTCA vdev

is the virtual device number of the CTCA that you detached.

userid vdev

is the user identification and virtual device number of the CTCA that your CTCA had been coupled to.

This response is always followed by the response:

CTCA vdev DETACHED

to indicate that the command has been successfully executed. The response:

CTCA vdev DROPPED BY userid vdev

is sent to the user whose CTCA connection was dropped.

Response 4

vdev pagedev1 pagedev2 pagedev3 DETACHED

is the response the user receives when a 3350 DASD attached to a 3380 model 11 or 21 storage control unit is detached after the user has initiated the detach operation.

where:

vdev

is the base virtual device number of the multiple-exposure device that was detached.

pagedev1**pagedev2****pagedev3**

are the virtual paging device numbers of the multiple-exposure device that was detached.

DETACH VIRTUAL (devices)

Response 5

vdev pagedev1 pagedev2 pagedev3 DETACHED BY OPERATOR

is the response the user receives when a 3350 DASD attached to a 3380 model 11 or 21 storage control unit is detached by the operator.

where:

vdev

is the base virtual device number of the multiple-exposure device that was detached.

pagedev1

pagedev2

pagedev3

are the virtual paging device numbers of the multiple-exposure device that was detached.

operator

is the userid of the operator who detached your virtual device.

Response 6

DASD { rdev rdev rdev rdev } DETACHED { userid } rdev rdev rdev rdev
 { ldev ldev ldev ldev } { SYSTEM }

The primary system operator receives this response when a real 3880 paging subsystem is detached from a specified user. The virtual base address and virtual exposures of the detached device are shown.

In the above responses, 'type' is one of the following:

Type	Meaning
DASD	Direct access storage device
TAPE	Magnetic tape
RDR	Card reader
PRT	Printer
PUN	Card punch
GRAF	Display device
CONS	Console
CTCA	Channel-to-channel adapter
CTLR	3704 or 3705 Communications Controller
MSC	Mass Storage Controller
DEV	Any other device

Migration Notes

VM/SP HPO:

1. VM/XA SP returns a response line for each detached device.
2. The VM/XA SP response from detaching a virtual CTCA device uses CTCA, or DROPPED instead of CTC, or DROP as in VM/SP HPO.
3. VM/XA SP allows DETACH during I/O tracing.
4. The VM/XA SP response has five digits for the number of users.

VM/XA SF

1. VM/XA SP uses GRAF rather than DISP in its response.
2. The VM/XA SP response from detaching a 3880 paging subsystem will not contain the letters D-A-S-D, as will the response in VM/XA SF.

DETACH VECTOR

Privilege Class: G

Use the DETACH VECTOR command to remove a virtual Vector Facility from a virtual configuration.

DETach	VECtor $\left\{ \begin{array}{l} \{nn\} \\ \{nn1-nn2\} \\ \left[\begin{array}{l} \{nn\} \\ \{nn1-nn2\} \end{array} \dots \right] \\ \text{ALL} \end{array} \right\}$
--------	---

where:

VECTOR

detaches the virtual Vector Facilities from the virtual processors with the addresses specified.

nn

is a single virtual processor address in hexadecimal.

nn1-nn2

is a nonwrapping pair of addresses in hexadecimal separated by a hyphen and not separated by any blanks.

ALL

specifies all virtual vector facilities currently defined.

Usage Notes

1. The virtual Vector Facility is also removed if its associated virtual processor is detached.
2. When a virtual VF is detached from a dedicated virtual processor of the preferred (V=R) guest and automatic dedication is set on, VM/XA SP attempts to reassign that virtual processor to a real processor without a real VF. If this attempt fails (because there is no nondedicated real processor without VF), then the virtual processor remains dedicated to a processor with VF, and that real VF becomes idle. It cannot be shared by other users until the processor has been undedicated.
3. If the virtual CPU belongs to the V=R guest who has the automatic dedication option turned off, or the virtual CPU belongs to a V=V or V=F user, the real processor is not undedicated by the DETACH VECTOR command.

Examples

To detach the virtual VF from CPU 1, enter:

```
DETACH VECTOR 1
```

To detach the virtual VFs from CPUs 0, 2, and 3, enter:

```
DETACH VECTOR 0 2-3
```

Responses

VECTOR FACILITY nn DETACHED

where:

nn

is the virtual processor address from which the virtual VF has been detached.

DETACH XSTORE

Privilege Class: B, G

Use **DETACH XSTORE** to remove a real Expanded Storage partition from a user's virtual machine.

		Class
DETach	{ XSTore } { [FRom] userid }	B
	{ XSTorage }	
	{ XSTore }	G
	{ XSTorage }	

where:

XSTORE
XSTORAGE

indicates that you are detaching Expanded Storage.

FROM userid

is the user identification of the virtual machine to which Expanded Storage is currently attached. The specified user must be logged on, not in the process of logging on. **SYSTEM** is not a valid **userid**.

Usage Notes

1. The detached virtual machine partition becomes immediately available for **CP** usage.
2. Expanded Storage can be detached from a user even if it is still in use. However, unexpected results can occur.
3. Whenever the Expanded Storage is detached, the system operator receives a message indicating that the storage is available.
4. The **DETACH XSTORE** command does not execute synchronously when you are detaching Expanded Storage from another **userid**. The issuer of this command may regain control before all processing has been completed for the other user.

For example, suppose the following **CMS** command sequence exists in a **CMS** exec procedure:

```
CP DETACH XSTORE USERA
CP ATTACH XSTORE USERB
```

CP may begin to attach Expanded Storage to **USERB** before the **DETACH** command completes. Since Expanded Storage is still attached to **USERA**, a delay occurs until the detachment is complete.

5. If a class **B** user issues **DETACH XSTORE** with no operands, then the command is in the class **G** format. Therefore, the command detaches any Expanded Storage from the user issuing the command.
6. If a **RETAIN XSTORE** command was issued but left pending **RETAIN**, the Expanded Storage detached will be used in an attempt to satisfy the amount remaining from the **RETAIN** request.

Responses

Receiving messages indicating that the DETACH XSTORE was successful are: the user who currently owns the Expanded Storage partition, the user who issued the DETACH XSTORE command (if different from the owner of the Expanded Storage partition), and the primary system operator (if different from the user who issued the command).

Response 1

XSTORE DETACHED BY originator

is the response sent to the owner of the Expanded Storage partition.

Response 2

XSTORE DETACHED FROM userid

is the response issued to the originator of the DETACH XSTORE command if the originator is not the owner of the Expanded Storage partition.

Response 3

XSTORE DETACHED FROM userid BY originator

is the response sent to the primary system operator if another user (the originator) issued the DETACH XSTORE command.

DIAL

Privilege Class: Any

Use the DIAL command to connect a display logically for use by a multiple-access operating system running in a virtual machine.

Dial	userid [vdev]
------	---------------

where:

userid

is the identification of the virtual machine in which the multiple-access operating system is running. SYSTEM and * are not valid userids for this command.

vdev

is the device number of the local virtual display to which the connection is to be made.

Usage Notes

1. VM/XA SP accepts a DIAL command only at logon, and only as a substitute for a LOGON command. The display you use must be supported by both VM/XA SP and the multiple-access operating system.
2. Once the connection is made, your display operates entirely under the control of the multiple-access operating system. The DIAL command matches your display to an equivalent type defined in the multiple-access system. If no matching display type exists, the connection cannot be made and an error message is issued.
3. The operator of a virtual system can return the display to CP control by issuing the CP RESET command. A virtual system user can return the display by causing an unsolicited device end to be generated for the dialed display. To generate an unsolicited device end, you can use the TEST/NORMAL or POWER OFF/POWER ON functions.
4. A SNA/CCS terminal user cannot dial into the VTAM service machine controlling the terminal.
5. The write-for-position operation is not supported for SNA/CCS terminals that are controlled by a level of VTAM service machine that is prior to ACF/VTAM Version 3, Release 1.1, for VM/SP (VM/VTAM). For this operation, a write CCW is chained to a read CCW; the write operation sets the 3270 buffer address from which the read operation is to begin. If a user attempts a write-for-position operation that is not supported by the controlling VTAM service machine, then the virtual machine receives an operation exception on the read operation.
6. The DIAL command is not supported for SNA/CCS terminals controlled by a VCNA level of VTAM service machine. ACF/VTAM Version 3 for VM/SP (VM/VTAM) VSMs and later support the DIAL command for SNA/CCS terminals.

Response 1

DIALED TO userid vdev

is the response sent to the user indicating that a logical connection has been made.

where:

userid

is the user identification of the virtual machine in which the multiple-access operating system that you want to use is running.

vdev

is the virtual device number of the display to which the dialed connection has been made.

Response 2

$$\left. \begin{array}{l} \text{type } \left\{ \begin{array}{l} \text{rdev} \\ \text{ldev} \end{array} \right\} \\ \text{luname} \end{array} \right\} \text{ DIALED TO userid DIALED} = \text{nnnnn}$$

is the response sent to the primary system operator. It indicates a successful connection to the virtual machine (userid) and the total number of VM/XA SP displays (nnnnn) currently connected to other virtual machines.

type

is the user's display device type.

rdev**ldev**

is the real or logical device number of your console.

luname

identifies a SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

Response 3

DROP FROM userid vdev

is the message sent to the user when the display is reset.

Response 4

$$\left. \begin{array}{l} \text{type } \left\{ \begin{array}{l} \text{rdev} \\ \text{ldev} \end{array} \right\} \\ \text{luname} \end{array} \right\} \text{ DROP FROM userid DIALED} = \text{nnnnn}$$

is the message sent to the primary system operator when the display is reset.

Migration Notes

VM/SP HPO: The VM/XA SP response has five digits for the number of users.

VM/XA SF: The VM/XA SP responses are different and *type rdev* is omitted from the beginning of the responses.

DISABLE**Privilege Class:** A, B, F

Use the **DISABLE** command to prevent display stations and SNA/CCS terminals from accessing the VM/XA SP system.

DISAb1e	<pre> rdev rdev ... rdev-rdev ... ldev ldev ... ldev-ldev ... rdev ldev ... SNA [userid] ALL </pre>
---------	--

where:

rdev**rdev ...****rdev-rdev ...**

is a real device number, a list of real device numbers, or a range of real device numbers of display stations that you want to disable. Specify each real device number as one to four hexadecimal digits. If you specify more than one device number, each must be separated from the others by at least one blank.

ldev**ldev ...****ldev-ldev ...**

is a logical device number, a list of logical device numbers, or a range of logical device numbers of 328x display printers that you want to disable. Only 328x display printers are valid for this operand. Specify a logical device number as an "L" followed by four hexadecimal digits. If you specify more than one device number, each must be separated from the others by at least one blank.

rdev ldev ...

is a list of real and logical device numbers of display stations that you want to disable. Each must be separated from the others by at least one blank.

SNA userid

disables communication with the SNA/CCS terminals owned by the VTAM service machine whose userid is specified. If no userid is specified, communication with all SNA/CCS terminals for all VTAM service machines is disabled.

ALL

disables all real 3270 displays, including the operator's display, all real 328X printers, and all SNA/CCS terminals.

Note: The logical 328X printers are not disabled.

Usage Notes

1. On a single command line you may combine device numbers, lists of device numbers, and ranges of device numbers. The beginning and ending device numbers of a range must either be both real display device numbers or logical display printer numbers.
2. If the displays you specify on the **DISABLE** command are not active, they are disabled immediately. If the displays are active, they are disabled when the user issues a **CP LOGOFF** or **CP DISCONNECT** command. A display is active if a user is logged on to it, or if a user is dialed from it.
3. If you issue a **CP ENABLE** command to a display before it is physically disabled, you reinstate the display's enabled status and negate previously issued **DISABLE** commands directed to the display.

Responses

COMMAND COMPLETE

indicates the successful execution of the command.

Migration Notes

VM/SP HPO: If **VM/XA SP** encounters an invalid device number in a list, it issues an error message and continues processing the rest of the devices.

DISCONNECT

Privilege Class: Any

Use the DISCONNECT command to disconnect your display station from the VM/XA SP system without stopping operations in your virtual machine.

DISConnect	[HOLD]
------------	--------

where:

HOLD

causes the TTY display terminal telecommunication connection to the 7171 ASCII DACU to remain after your session is disconnected.

Usage Notes

1. Once you have issued the DISCONNECT command, you remain disconnected until you issue the CP LOGON command. If your virtual machine goes into a disabled wait state or if a read is posted from the virtual machine or CP, and no stacked console input exists (that is, physical interaction from the console is necessary), you have 15 minutes in which to log on or your virtual machine is automatically logged off. This is an important consideration if you are using an operating system that issues frequent "reads" to your console.
2. If your connection is broken because of console errors, CP places your virtual machine in disconnect mode for up to 15 minutes. Your virtual machine will not continue to run unless you have previously issued CP SET RUN ON. You can log on within the 15-minute interval if you want to continue operations on your virtual machine; if you do not log on or your secondary user does not log on within the 15 minute interval your virtual machine is automatically logged off.
3. If you do not have a functional secondary user and your virtual machine is in disconnect mode, all "writes" or output messages sent to your virtual console are ignored unless you have issued the CP command SPOOL CONSOLE START. If you have a secondary user who is logged on and not disconnected, that user receives your messages and can enter commands and responses for you.
4. When your virtual machine is reconnected using the normal logon procedure, it is placed in CP console function mode. To resume execution on your virtual machine, enter the CP BEGIN command.
5. If you specify the HOLD option on a SNA/CCS terminal, the command is invalid and an error message is issued. If you specify the HOLD option on a TTY display terminal connected to a 7171 ASCII DACU, it is accepted and processed. The HOLD option is accepted but ignored on all other terminals.
6. If your installation has coded the E3270HLD feature in the system real I/O configuration file for your TTY display terminal, the telecommunication connection will always remain, regardless of the HOLD option.

Response 1

DISCONNECT AT hh:mm:ss zone weekday mm/dd/yy

is the response to the user who issued the command.

where:

hh:mm:ss

is the current time

zone

is the time zone

weekday

is the current day of the week

mm/dd/yy

is the current date in month/day-of-month/year format.

Response 2

$\left. \begin{array}{l} \text{type } \{ \text{rdev} \} \\ \{ \text{ldev} \} \\ \text{luname} \end{array} \right\} \text{DISCONNECT userid USERS = nnnnn}$

is the response sent to the primary system operator

where:

type

is the user's display type.

rdev

ldev

is the real or logical device number associated with the display.

luname

identifies a SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

userid

is the identification of the user who has disconnected from the system.

USERS = nnnnn

is the total number of users remaining on the system.

Migration Notes

VM/SP HPO: VM/XA SP allows you to spell out the entire command DISCONNECT.

DISPLAY

Use the **DISPLAY** command described in this section to display the following real and virtual machine components:

- Contents of first-level storage (the real storage of the 370-XA processor environment)
- Contents of second-level storage (the storage that appears real to the operating system running in your virtual machine)
- Contents of third-level storage (the storage that appears virtual to the operating system running in your virtual machine)
- Old and new PSWs and interrupt information for the following interrupts:

SVC
I/O
PROGRAM
EXTERNAL
MACHINE CHECK

- Storage keys
- Contents of registers:
 - Prefix
 - General
 - Floating-point
 - Control
- Subchannel information blocks (SCHIBS)
- The current program status word (CURRENT PSW)
- The contents of registers, activity count, and section size for the Vector Facility:

Vector registers (VR)
Vector register pair(s) (VP)
Vector status register (VSR)
Vector mask register (VMR)
Vector activity count (VAC)
Vector section size (VSS).

Complete Format for DISPLAY

The format box below shows all of the operands available with the **DISPLAY** command. Complete descriptions of the operands appear on the following pages, in the same sequence as the operands appear here.

		Class
Display	$\left[\begin{array}{c} R \\ L \\ V \\ W \end{array} \right] \left[\begin{array}{c} T \\ K \\ S \\ I \\ N \\ U \end{array} \right] \left\{ \begin{array}{l} [\text{hexloc1}] \left[\begin{array}{l} \{-\} [\text{hexloc2}] \\ \{:\} \text{END} \end{array} \right] \\ \underline{0} \\ \{.\} [\text{bytecount}] \\ \text{END} \end{array} \right\}$	G
	$H \left[\begin{array}{c} T \\ K \\ S \\ I \\ N \\ U \end{array} \right] \left\{ \begin{array}{l} [\text{hexloc1}] \left[\begin{array}{l} \{-\} [\text{hexloc2}] \\ \{:\} \text{END} \end{array} \right] \\ \underline{0} \\ \{.\} [\text{bytecount}] \\ \text{END} \end{array} \right\}$	C/E
	PREFIX	G
	$\text{Psw} \left[\begin{array}{c} \text{EXT} \\ \text{SVC} \\ \text{PROG} \\ \text{PRG} \\ \text{MCH} \\ \text{I/O} \\ \text{ALL} \end{array} \right]$	G
	$\left\{ \begin{array}{c} G \\ Y \\ X \end{array} \right\} \left[\begin{array}{c} \text{reg1} \\ \text{reg1} \end{array} \right] \left\{ \begin{array}{l} \{-\} [\text{reg2}] \\ \{:\} \text{END} \\ \{.\} [\text{regcount}] \\ \text{END} \end{array} \right\}$	G
	$\text{SCHib} \left\{ \begin{array}{l} [\text{vsch1}] \left[\begin{array}{l} \{-\} [\text{vsch2}] \\ \{:\} \text{END} \end{array} \right] \\ \{.\} [\text{count}] \\ \text{END} \end{array} \right\}$ <p style="text-align: center;">ALL</p>	G
	$\left\{ \begin{array}{c} \{VR\} \\ \{VP\} \end{array} \right\} \left[\begin{array}{c} \text{reg1} \left[\begin{array}{l} \{-\} [\text{reg2}] \\ \{:\} \text{END} \end{array} \right] \\ \{.\} [\text{regcount}] \\ \text{END:} \end{array} \right] \left[\begin{array}{c} ,\text{elt1} \left[\begin{array}{l} \{-\} [\text{,elt2}] \\ \{:\} \text{END} \end{array} \right] \\ \{.\} [\text{eltcount}] \\ \text{END:} \end{array} \right] \end{array} \right]$	G
	VSR	
	VMR	
	VAC	
	VSS	

General Usage Notes for DISPLAY

1. Using a single command, you can display the contents of multiple storage locations, registers, and control words.

There are two rules that you must follow in order for the command to execute properly:

- a. Do not put blanks between operands that relate to a single DISPLAY request, except for PSW and SCHIB which require at least one blank between the first and second operands.

For example, if you want to display first-level storage location 40, enter:

```
display h40
```

- b. Separate each set of display requests from others by at least one blank.

For example, if you want to display first-level storage location 40 and the contents of register 5, enter:

```
display h40 g5
```

2. The defaults for the following DISPLAY operands are:

Command Operand	Default Before Rounding
G	0-end
G-	0-end
G-5	0-G5
G5-	G5-end
G5	G5
T	error
T-	T0-end
T-1000	T0-1000
T1000-	T1000-end
T1000	T1000
PSW	current

When displaying a range of addresses or subchannels, the command operand must be followed by an address, a range, or a punctuation mark ('-', ':', '.'), which will form a range. When displaying registers, the command operand can stand alone.

3. If you enter a command line that contains an invalid operand, the DISPLAY command will terminate when it encounters that operand. Any previous valid operands are processed before termination occurs.
4. The DISPLAY command allows you to display information in the following formats:
 - Hexadecimal string/Hexadecimal word segments
 - EBCDIC translation for uppercase letters and numbers 0-9
 - Hexadecimal word segments with EBCDIC translation
 - Instruction
 - Storage key

Note: Hexadecimal to EBCDIC translation is done for the numerals 0 - 9 and uppercase letters A - Z only. If you enter multiple operands on a single command line to display storage or a register and do not specify the format that you want the DISPLAY command processor to use, the last format type that you explicitly entered on the command line will define the type of display that you receive. That is, whenever you explicitly enter a display type, you implicitly define the default for subsequent operands of the current display function. If you do not enter any display format type on the command line, the system defaults will be employed.

For example, the command:

```
display 10 20 T40 80 G12 5 L60-100
```

displays the following, respectively:

```
hexadecimal location 10 without EBCDIC translation
hexadecimal location 20 without EBCDIC translation
hexadecimal location 40 with EBCDIC translation
hexadecimal location 80 with EBCDIC translation
general register 12
general register 5
hexadecimal locations 60 through 100 with EBCDIC translation
```

5. If you want to terminate the DISPLAY function while data is being displayed, press the PA1 key. When the display terminates, you can enter another command.
6. Use the DISPLAY command to display the contents of various storage locations, registers, and control words. You can examine this information during program execution, and may be able to determine the cause of program errors.

To do this, you would typically define a TRACE trap, using the CP TRACE command, that will stop program execution at a specified point. Once program execution suspends, your virtual machine will enter the CP environment, and you can then issue the DISPLAY command. Alternatively, you can execute the DISPLAY command directly from TRACE using the TRACE command CMD option. For more information, see the description of the CP TRACE command.
7. If the DISPLAY command processor is suppressing identical lines, it issues a response every time a segment boundary is crossed rather than waiting for a nonidentical line. This provides you with an opportunity to terminate DISPLAY processing by pressing the PA1 key.

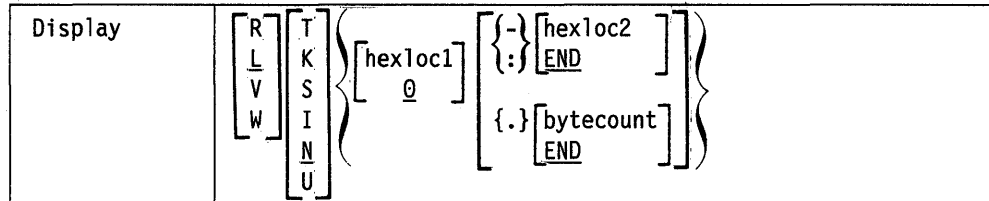
Migration Notes

VM/SP HPO: In VM/XA SP, you can store into the CAW or CSW but you cannot display their contents.

DISPLAY GUEST STORAGE

Privilege Class: G

Use DISPLAY GUEST STORAGE to display the contents of storage at specified locations in your virtual machine.



Note: Do not put blanks between operands that relate to a single DISPLAY request, except for PSW and SCHIB which require at least one blank between the first and second operands. For more information, see “General Usage Notes for DISPLAY” on page 128.

where:

R

displays the contents of second-level storage (storage that appears real to your virtual machine).

L

displays the contents of second-level storage (storage that appears real to your virtual machine) if the guest virtual machine is operating in DAT OFF mode; displays the contents of third-level storage (storage that appears virtual to your virtual machine) if the guest virtual machine is operating in DAT ON mode.

V

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using control register 1.

W

displays the contents of third-level storage (storage that appears virtual to your virtual machine) using control register 7.

T

displays the contents of storage in hexadecimal with EBCDIC translation. Storage is displayed in word segments. An address range will be displayed starting at the X'10' boundary in which the specified address resides. Storage is displayed in X'10' increments.

The storage key is displayed on the first line, as well as at every page boundary.

K

displays the storage keys in hexadecimal for all page frames or partial page frames included in the specified address range.

S

displays the contents of storage in hexadecimal translation. Storage is displayed as a string starting at the location specified. The specified address range is not boundary aligned.

The storage key is displayed on the first line, as well as at every page boundary.

I

displays the contents of storage in instruction format. The specified address is rounded down to the nearest half word boundary.

The storage key is displayed on the first line, as well as at every page boundary.

N

displays the contents of storage in hexadecimal without EBCDIC translation. This is the default. Storage is displayed in word segments. An address range will be displayed starting at the full word in which the specified address resides. Storage is displayed in full word increments.

The storage key is displayed on the first line, as well as at every page boundary.

U

displays the contents of storage in EBCDIC translation. Storage is displayed as an EBCDIC string starting at the address specified. The specified address range is not boundary aligned.

The storage key is displayed on the first line, as well as at every page boundary.

If you do not specify any of the options T, K, S, I, N, or U, the contents on storage are displayed in hexadecimal word segments.

hexloc1

is the starting storage location that is to be displayed. The address hexloc1 may be one-to-eight hexadecimal digits; leading zeros are optional. The default is zero.

Storage locations and the minimum amount displayed are rounded according to the following conventions:

Rounding Conventions (in hex)

Storage Locations	Start	Minimum Length
T	10	10
K	7FF	7FF
S	1	1
I	2	2-6 (depending on instruction)
N	4	4
U	1	1

**{-} hexloc2
{:} END**

is the last of a range of hexadecimal storage locations whose contents are to be displayed. If specified, hexloc2 must be equal to or greater than hexloc1 and must be within the virtual machine storage size. The address, hexloc2, may be from one-to-eight hexadecimal digits; leading zeros are optional. The symbol “:” means the same as a “-” when you specify a range of addresses.

Specify END if you want to display the remaining contents of storage. END is the default. and you specify “-”, “:”, or “.”.

**{.} bytecount
END**

is a hexadecimal integer designating the number of bytes of storage (starting with the byte at hexloc1) to be displayed.

The bytecount that you specify must have a value of at least one and may be from one to eight hexadecimal digits. Leading zeros are optional.

DISPLAY GUEST STORAGE

Specify END if you want to display the remaining contents of storage. END is the default if you specify : “-”, “:”, or “.”.

Usage Notes

When running with a saved segment, you can display storage locations outside the range of your virtual machine size if they are within the saved segment. If locations exist between the upper limit of your virtual machine and the address at which the saved segment was saved, any attempt to display those locations (or associated keys) results in the “nonaddressable storage” response being displayed.

Refer to “General Usage Notes for DISPLAY” on page 128.

Responses

One or more of the following responses is displayed, depending upon the operands you specified:

Response 1

```
Xhexloc word1 word2 word3 word4 [Key] [*EBCDIC TRANSLATION*] [Lhexloc]
```

This is the response you receive if your display request was L, R, V, or W, with the T or N option. X indicates the type of display request (L, R, V, or W). The EBCDIC translation is given only for the T option. The storage key is displayed on the first line of a request, and at every page boundary.

For a Vhexloc or Whexloc request, the Lhexloc trailer shows the second-level address that corresponds to the given third-level address. The Lhexloc trailer is shown also when DAT-ON mode causes Lhexloc to resolve to a third-level address. The storage key is displayed on the first line of a request, and at every page boundary.

Response 2

```
Xhexloc *EBCDIC TRANSLATION* [key] [Lhexloc]
```

This is the response you receive when your display request is L, R, V, or W, with the U option. Xhexloc is the hexadecimal location of the first data byte to be displayed and X can be L, R, V, or W, depending on your display request. A minimum of 1 character and a maximum of 32 characters will be displayed on each line. The contents of storage are displayed only in an EBCDIC translation. Periods in your display represent nonprintable characters. The storage key is displayed on the first line of a request and at every page boundary.

For a Vhexloc or Whexloc request, the Lhexloc trailer shows the second-level address that corresponds to the given third-level address. The Lhexloc trailer is shown also when DAT-ON mode causes Lhexloc to resolve to a third-level address. The storage key is displayed on the first line of a request, and at every page boundary.

Response 3

```
Xhexloc mnemonic X-X mnemonic X-X mnemonic X-X [key] [Lhexloc]
```

This is the response you receive when your display request is L, R, V, or W, with the I option. Xhexloc is the hexadecimal storage location of the first instruction to be displayed and X can be L, R, V, or W, depending on the request. Mnemonic is up to a five character representation of the instruction code. X-X is the actual

hexadecimal contents of storage and is either two, four, or six bytes as determined by the instruction length code. Unrecognizable instruction codes will be indicated by a series of asterisks. A minimum of one instruction and a maximum of three instructions will be displayed on each line. The storage key is displayed on the first line of a request and at every page boundary.

For a Vhexloc or Whexloc request, the Lhexloc trailer shows the second-level address that corresponds to the given third-level address. The Lhexloc trailer is shown also when DAT-ON mode causes Lhexloc to resolve to a third-level address. The storage key is displayed on the first line of a request, and at every page boundary.

Response 4

Xhexloc stringhexadecimaldata [key][Lhexloc]

This is the response you receive if your display request was L, R, V, or W with the S option. X indicates the type of display request (L, R, V, or W). The storage key is displayed on the first line of a request, and at every page boundary.

For a Vhexloc or Whexloc request, the Lhexloc trailer shows the second-level address that corresponds to the given third-level address. The Lhexloc trailer is shown also when DAT-ON mode causes Lhexloc to resolve to a third-level address. The storage key is displayed on the first line of a request, and at every page boundary.

Response 5

Xhexloc1 TO hexloc2 KEY = kk

This is the response when your display request is L, R, V, or W, with the K option. hexloc1 and hexloc2 are storage locations, and kk is the associated storage key. The X in Xhexloc can be L, R, V, or W, depending upon the display request. A response will be sent for every contiguous address range that has unique storage keys and whenever a segment boundary is crossed.

Response 6

If your display request includes a request to display nonaddressable storage, you receive the following response:

Xhexloc1 TO hexloc2 NON-ADDRESSABLE STORAGE

hexloc1 and hexloc2 are storage locations. The X in Xhexloc1 can be L, R, V, or W, depending on the request.

Response 7

If your display request includes a request to display one or more lines identical to the previous line, you receive the following response:

Xhexloc1 TO hexloc2 SUPPRESSED LINE(S) SAME AS ABOVE ...

hexloc1 and hexloc2 are storage locations. The X in Xhexloc1 can be L, R, V, or W, depending on the request.

Migration Notes

VM/SP HPO

1. In VM/XA SP you can display larger storage sizes than in VM/SP HPO.
2. In VM/XA SP you can display more levels of storage than in VM/SP HPO.

VM/XA SF

1. In VM/XA SP, the response SUPPRESSED LINE(S) SAME AS ABOVE appears whether or not the amount of storage to be displayed exceeds X'100' bytes.
2. VM/XA SP displays storage keys at the first line of a request and at every page boundary. It shows storage keys even if you request only one address.
3. In VM/XA SP, the default option is N.
4. In VM/XA SP, the N option rounds storage locations to fullword boundaries, and displays storage in fullword increments.

DISPLAY HOST STORAGE

Privilege Class: C and E

Use DISPLAY HOST STORAGE to display the contents of first-level storage.

Display	H	$\left[\begin{array}{c} T \\ K \\ S \\ I \\ N \\ U \end{array} \right] \left\{ \begin{array}{l} \left[\text{hexloc1} \right] \left\{ \begin{array}{l} - \\ : \end{array} \right\} \left[\begin{array}{l} \text{hexloc2} \\ \text{END} \end{array} \right] \\ \left[\text{0} \right] \left\{ \begin{array}{l} . \end{array} \right\} \left[\begin{array}{l} \text{bytecount} \\ \text{END} \end{array} \right] \end{array} \right\}$
---------	---	--

Note: Except for DISPLAY PSW and DISPLAY SCHIB, do not put blanks between operands that relate to a single DISPLAY request. DISPLAY PSW and DISPLAY SCHIB require at least one blank between the first and second operands. For more information, see “General Usage Notes for DISPLAY” on page 128.

where:

H

displays the contents of first-level storage (storage that appears real to CP).

T

displays the contents of storage in hexadecimal with EBCDIC translation. Storage is displayed in word segments. An address range will be displayed starting at the X'10' boundary in which the specified address resides. Storage is displayed in X'10' increments.

The storage key is displayed on the first line, and at every page boundary.

K

displays in hexadecimal the storage keys for all page frames or partial page frames included in the specified address range.

S

displays the contents of storage in hexadecimal translation. Storage is displayed as a string starting at the location specified. The specified address range is not boundary aligned.

The storage key is displayed on the first line, and at every page boundary.

I

displays the contents of storage in instruction format. The specified address is rounded down to the nearest half word boundary.

The storage key is displayed on the first line, and at every page boundary.

N

displays the contents of storage in hexadecimal without EBCDIC translation. This is the default. Storage is displayed in word segments. An address range will be displayed starting at the full word in which the specified address resides. Storage is displayed in full word increments.

The storage key is displayed on the first line, and at every page boundary.

DISPLAY HOST STORAGE

U

displays the contents of storage in EBCDIC translation. Storage is displayed as an EBCDIC string starting at the address specified. The specified address range is not boundary aligned.

The storage key is displayed on the first line, and at every page boundary.

If you do not specify any of the options T, K, S, I, N, or U, the contents of storage are displayed in hexadecimal word segments.

hexloc1

is the starting storage location that is to be displayed. The address hexloc1 may be one to eight hexadecimal digits; leading zeros are optional. The default is zero.

{-} [**hexloc2**]
{:} [**END**]

is the last of a range of hexadecimal storage locations whose contents are to be displayed. If specified, hexloc2 must be equal to or greater than hexloc1, and must be within the bounds of addressable storage. The address hexloc2 may be from one to eight hexadecimal digits; leading zeros are optional. The symbol ":" means the same as "-" when specifying a range of addresses.

Specify END if you want to display the remaining contents of storage. END is the default if you specify "-", ":", or ".".

{.} [**bytecount**]
[**END**]

is a hexadecimal integer designating the number of bytes of storage (starting with the byte at hexloc1) to be displayed. The sum of hexloc1 and bytecount must be less than the bounds of addressable storage. This value is the lesser of the storage size specified at system generation and the real storage size.

The specified bytecount must have a value of at least one and may be from one to eight hexadecimal digits. Leading zeros are optional.

Specify END if you want to display the remaining contents of storage. END is the default if you specify "-", ":", or ".".

Usage Notes

Only CP-owned pages may be displayed.

While displaying data in CP storage, the running of a virtual machine or someone else using the STORE HOST command could cause data to be changed. This could create an inconsistency in the data that is viewed.

Responses

One or more of the following responses is displayed, depending upon the operands specified.

Response 1

```
Hhexloc word1 word2 word3 word4 [Key] [*EBCDIC TRANSLATION*]
```

This is the response for the storage location requests when H is your display request with T or N option. The EBCDIC translation is only displayed if you have also entered the T option. The storage key is displayed on the first line, and at every page boundary.

Response 2

Hhexloc *EBCDIC TRANSLATION* [key]

If your display request is H with the U option, you receive this response. Hhexloc is the hexadecimal location of the first data byte to be displayed. A minimum of 1 character and a maximum of 32 characters will be displayed on each line. The contents of storage will be displayed only in EBCDIC translation. Periods in the display data represent nonprintable characters. The storage key is displayed on the first line, and at every page boundary.

Response 3

Hhexloc mnemonic X-X mnemonic X-X mnemonic X-X [key]

If your display request is H with the I option, you receive this response. Hhexloc is the hexadecimal storage location of the first instruction to be displayed. mnemonic is up to a 5-character representation of the instruction code. X-X is the actual hexadecimal contents of storage and is either 2, 4, or 6 bytes as determined by the instruction length code. Unrecognizable instruction codes will be indicated by a series of asterisks. A minimum of one instruction and a maximum of three instructions will be displayed on each line. The storage key is displayed on the first line and at every page boundary.

Response 4

Hhexloc stringhexadecimaldata [key]

This is the response you receive if your display request was H with the S option. The storage key is displayed on the first line of a range, and at every page boundary.

Response 5

Hhexloc1 TO hexloc2 KEY = kk

If your display request is H with the K option, you receive this response. hexloc1 and hexloc2 are storage locations and kk is the associated storage key for Hhexloc1. You will receive this response for every contiguous address range that has unique storage keys, and whenever a segment boundary is crossed.

Response 6

If the storage location you indicated in the command line is not a CP-owned page, you receive the following response:

Hhexloc1 TO hexloc2 SUPPRESSED NON-CONTROL-PROGRAM PAGE

hexloc1 and hexloc2 are storage locations.

Response 7

If your display request includes a request to display one or more lines identical to the previous line, you receive the following response:

Xhexloc1 TO hexloc2 SUPPRESSED LINE(S) SAME AS ABOVE ...

hexloc1 and hexloc2 are storage locations. The X in Xhexloc can be L, R, V, or W, depending on the request.

DISPLAY HOST STORAGE

Migration Notes

VM/XA SF

1. In VM/XA SP, the response **SUPPRESSED LINE(S) SAME AS ABOVE** appears whether or not the amount of storage to be displayed exceeds X'100' bytes.
2. VM/XA SP displays storage keys at the first line of a request and at every page boundary. It shows storage keys even if you request only one address.
3. In VM/XA SP, the default format is N.
4. In VM/XA SP, the N option rounds storage locations to fullword boundaries, and displays storage in fullword increments.

DISPLAY PREFIX

Privilege Class: G

Use DISPLAY PREFIX to display the contents of the prefix register, which is the absolute location of the guest page 0.

Display	PREFIX
---------	--------

where:

PREFIX

displays the contents of the prefix register.

Responses

PREFIX = xxxxxxxx

displays the contents of the prefix register in hexadecimal.

DISPLAY PROGRAM STATUS WORDS

Privilege Class: G

Use DISPLAY PROGRAM STATUS WORDS to display information about your virtual machine's program status words (PSWs) as two hexadecimal words.

Display	Psw	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> EXT SVC PROG PRG MCH I/O ALL </div>
---------	-----	---

where:

PSW

requests the display of a current virtual machine PSW. When entered without additional operands, the value of the current PSW will be displayed.

EXT

SVC

PROG

PRG

MCH

I/O

displays the current values and appropriate interrupt information for both the new and old PSWs specified, that is, EXTERNAL, SVC, PROGRAM CHECK, MACHINE CHECK, and I/O.

ALL

displays the current values and appropriate interrupt information for the current PSW and all old and new interrupt PSWs (except RESTART).

Responses

One or more of the following responses is displayed, depending upon the operands specified.

Response 1

PSW = xxxxxxxx xxxxxxxx

displays the contents of the PSW in hexadecimal.

Response 2

```

EXT nnnn 18 OLD xxxxxxxx xxxxxxxx 58 NEW xxxxxxxx xxxxxxxx
SVC nnnn 20 OLD xxxxxxxx xxxxxxxx 60 NEW xxxxxxxx xxxxxxxx
PRG nnnn 28 OLD xxxxxxxx xxxxxxxx 68 NEW xxxxxxxx xxxxxxxx
MCH nnnn 30 OLD xxxxxxxx xxxxxxxx 70 NEW xxxxxxxx xxxxxxxx
I/O nnnn 38 OLD xxxxxxxx xxxxxxxx 78 NEW xxxxxxxx xxxxxxxx
    
```

displays the contents of the old and new PSWs and the interruption codes in hexadecimal. Each nnnn represents the EC-mode interruption code. The numbers 18, 20, 28... are storage locations. The X's represent the contents of the PSWs.

DISPLAY REGISTERS

Privilege Class: G

Use DISPLAY REGISTERS to display the contents of general purpose registers, floating point registers, or control registers.

Display	$\left\{ \begin{array}{c} G \\ Y \\ X \end{array} \right\} \left[\text{reg1} \right] \left[\begin{array}{l} \{-\} \text{reg2} \\ \{: \} \text{END} \\ \{.\} \text{regcount} \\ \text{END} \end{array} \right]$
---------	--

Note: Do not put blanks between operands that relate to a single DISPLAY request (except for PSW and SCHIB which require a blank). For more information, see “General Usage Notes for DISPLAY” on page 128.

where:

Greg1

is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, general register whose contents are to be displayed. If you specify G without a register number, the contents of all the general registers are displayed.

Yreg1

is an integer (0, 2, 4, or 6) representing the first, or only, floating-point register whose contents are to be displayed. If you specify Y without a register number, the contents of all of the floating-point registers are displayed.

Xreg1

is a decimal number from 0 to 15 or a hexadecimal number from 0 to F representing the first, or only, control register whose contents are to be displayed. If you specify X without a register number, the contents of all the control registers are displayed.

$\{-\} \text{reg2}$
 $\{: \} \text{END}$

is a number representing the last register whose contents are to be displayed. You must specify the hyphen “-” when you are requesting more than one register by register number. You must specify the range of registers in ascending order. Therefore, the value you specify for reg2 must be equal to or greater than the value you specify for reg1.

The symbol “:” means the same as “-” when specifying a range of addresses.

If you specify G or X, reg2 may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F. If you specify Y, reg2 may be 0, 2, 4, or 6. The contents of registers reg1 through reg2 are displayed.

Specify END to display the contents of all remaining registers. END is the default. Remaining registers are those in the range:

- From reg1 through register 15, if you are displaying general registers
- From reg1 through register 6, if you are displaying floating point registers
- From reg1 through register 15, if you are displaying control registers

DISPLAY REGISTERS

{.} **regcount**

END

is a decimal number from 1 to 16 or a hexadecimal number from 1 to F specifying the number of registers (starting with reg1) whose contents are to be displayed. If you specify G or X, regcount can be a decimal number from 1 to 16 or a hexadecimal number from 1 to F.

If display type Y is specified, regcount must be 1, 2, 3, or 4. The sum of reg1 and regcount must be a number that does not exceed the maximum register number for the type of registers being displayed.

Specify END to display the contents of all remaining registers. END is the default. Remaining registers are those in the range:

- From reg1 through register 15, if you are displaying general registers
- From reg1 through register 7, if you are displaying floating point registers
- From reg1 through register 15, if you are displaying control registers

Responses

One or more of the following responses is displayed, depending upon the operands specified.

Response 1

Displaying General Registers:

GPR nn = genreg1 genreg2 genreg3 genreg4

is the response you receive when you display general registers. nn indicates the first (or only) register whose contents are to be displayed; these contents are displayed in genreg1.

The contents of each register is displayed in hexadecimal. Up to four registers per line are displayed for each range you specify. If required, multiple lines are displayed, up to a maximum of four lines.

Response 2

Displaying Floating Point Registers:

FPR nn = xxxxxxxxxxxxxxxx ±.ddddddddddddddd E dd

This is the response you receive when you display floating-point registers. nn is the even-number floating-point register whose contents are displayed on the line. The contents of the floating-point registers are displayed in both the internal hexadecimal format and the E format; the first part of the response is the contents of the register in hexadecimal and the second part of the response is the contents in exponential decimal format. Only one register is displayed per line. If you specified a range of registers, multiple response lines are displayed.

Response 3

Displaying Control Registers:

ECR nn = ctlreg1 ctlreg2 ctlreg3 ctlreg4

This is the response you receive when you display control registers; nn is the first (or only) control register whose contents you want to display.

The contents of the requested control registers are displayed in hexadecimal. Up to four registers per line are displayed. If required, multiple response lines are displayed.

Migration Notes

VM/XA SF: The VM/XA SP response uses GPR, FPR and ECR instead of G, Y, and X as in VM/XA SF.

DISPLAY SCHIB

Privilege Class: G

Use DISPLAY SCHIB to display information from your subchannel information blocks (SCHIBs).

Note: This command is only valid if your virtual machine is operating in 370-XA mode.

Display	SCHib { [vsch1] [{ - } [vsch2] [END]] } { . } [count] [END] } ALL
---------	---

where:

SCHIB

displays information from your subchannel information blocks.

vsch1

is the starting (or only) virtual subchannel whose SCHIB is to be displayed.

{ - } vsch2
{ : } END

specifies that all SCHIBs in the range *vsch1* to *vsch2* are to be displayed, or that all SCHIBs from *vsch1* to the highest number subchannel are to be displayed. END is the default if you specify dash (-), a colon (:), or a period (.).

The symbol “:” means the same as “-” when specifying a range.

{ . } count
END

is the number of consecutive SCHIBs starting at *vsch1* that are to be displayed. Specify END to display all SCHIBs from *vsch1* to the highest number subchannel. END is the default if you specify dash (-), a colon (:), or a period(.).

ALL

specifies that SCHIBs for all virtual devices are to be displayed.

Usage Notes

1. If you specify a list of SCHIBS, and one is not found, the DISPLAY command will terminate. For example, if you enter:

```
display SCHIB 0022 0055 0011
```

 and subchannel 0055 is not found, the command will terminate. If you specify a range of SCHIBs, the DISPLAY command will terminate if the first SCHIB in the range is not found. Otherwise, all those SCHIBs that were found will be displayed.
2. If your virtual machine is running in 370-XA mode, you can use the CP QUERY VIRTUAL command to determine the subchannel number associated with a virtual device.

- The response to this command is a header line followed by a line of information for each SCHIB that you specify. (The header line identifies, column by column, the SCHIB information below it. See "Responses" below.) After every ten lines of SCHIB information, the header line reappears.

Responses

```
SCHIB DEV INT-PARM ISC FLG LP PNO LPU PI MBI PO PA CHPID0-3 CHPID4-7
nnnn nnnn nnnnnnnn n nn nn nn nn nn nnnn nn nn nnnnnnnn nnnnnnnn
```

If an interrupt is pending for the virtual subchannel, the following line describes the pending interrupt:

```
KEY ECC FPI ZN SCTL CCW-ADDR DS SS CNT EXT-LOG
n n n n nnnn nnnnnnnn nn nn nnnn nnnnnnnn
```

where:

INT-PARM	is the interrupt parameter.
ISC	is the interrupt subclass.
FLG	is the flag field.
LP	is the logical path mask.
PNO	is the path not operational mask.
LPU	is the last path used mask.
PI	is the path installed mask.
MBI	is the measurement block index.
PO	is the path operational mask.
PA	is the path available mask.
CHPID	are the channel path identifiers.

For the second line of the response, provided when an interrupt is pending on the subchannel:

KEY	is the SCSW protect key.
ECC	are the logout condition code bits.
FPI	are the format, prefetch, and initial response bits.
ZN	the initial status condition.
SCTL	are the function and activity control bytes.
CCW-ADDR	is the SCSW CCW address.
DS	are the SCSW device status bits.
SS	are the subchannel status bits.
CNT	is the SCSW residual CCW data count.
EXT-LOG	is the extended status/time word.

DISPLAY VECTOR

Privilege Class: G

This command displays to the terminal all the contents of the following:

- Vector registers (VR)
- Vector register pair or pairs (VP)
- Vector status register (VSR)
- Vector mask register (VMR)
- Vector activity count (VAC)
- Vector section size (VSS).

Display	$\left\{ \begin{array}{l} \text{VR} \\ \text{VP} \end{array} \right\} \left[\text{reg1} \left[\begin{array}{l} \{-\} \\ \{:\} \end{array} \right] \left[\begin{array}{l} \text{reg2} \\ \text{END} \end{array} \right] \right] \left[\begin{array}{l} \{.\} \\ \text{regcount} \\ \text{END} \end{array} \right] \right] \left[, \text{elt1} \left[\begin{array}{l} \{-\} \\ \{:\} \end{array} \right] \left[\begin{array}{l} \text{elt2} \\ \text{END} \end{array} \right] \right] \left[\begin{array}{l} \{.\} \\ \text{eltcount} \\ \text{END} \end{array} \right] \right] \right]$
	<p>VSR VMR VAC VSS</p>

where:

VR
displays one or more vector registers in hexadecimal.

VP
displays one or more even-odd vector register pairs as 64-bit elements in hexadecimal, along with the decimal equivalent in exponential format.

reg1
If you wish to display vector registers (VR), reg1 is a decimal number from 0 to 15 or a hex number from X'0' to X'F' representing the first, or only, vector register whose contents are to be displayed. If VR is specified without a register number, it will default to register 0.

If you wish to display a vector register pair (VP), reg1 is an even decimal number from 0 to 14 or an even hexadecimal number from X'0' to X'E' of the first or only vector register pair. If reg1 is not specified with VP, it will default to vector register pair 0.

$\left\{ \begin{array}{l} \{-\} \\ \{:\} \end{array} \right\} \text{reg2}$

If you wish to display a range of vector registers, reg2 is a decimal number from 0 to 15 or a hex number from X'0' to X'F' representing the last vector register whose contents are to be displayed. If you specify VP, reg2 must designate an even-numbered register. The dash (-) or colon (:) following reg1 specifies a range of register values for display. Specify the range of registers in ascending order. Reg2 must be equal to or greater than reg1. The contents of registers reg1 to reg2 will then be displayed.

{-}[END]
{:}

When used in the register specification, this calls for a display of all register contents from the starting registers specified by reg1 to register 15, or register pair 14-15.

{.}regcount

is a decimal number from 1 to 16 or a hex number from 1 to F specifying the number of vector registers or pairs whose contents are to be displayed. If VP is specified, the valid range is 1 to 8. If you display vector registers, the sum of reg1 and regcount must not be larger than 16.

,elt1

The comma indicates that a register element specification follows. If the comma and element specification is not given, CP displays the contents of elements 0-3 for VR, or element 0 only for VP. Etl1 is a hexadecimal number from zero to one less than the section size representing the first, or only, element to be displayed.

{-}elt2
{:}

is a hexadecimal number from 0 to one less than the section size representing the last element whose contents are to be displayed. The dash (-) and colon (:) before indicates that you wish to display a range of elements. The range of elements must be specified in ascending order. The operand elt2 must be equal to or greater than elt1. The contents of elements elt1 to elt2 are displayed.

{.}eltcount

is a hexadecimal number from 1 to the section size specifying the number of elements whose contents are to be displayed starting with elt1. The range indicated by elt1 and eltcount must not exceed the section size.

{-}[END]
{:}

When used in the element specification, this calls for a display of the contents of all elements from the starting element specified by elt1 to the last element, which is one less than the section size.

VSR

displays the current vector status register.

VMR

displays the current vector mask register.

VAC

displays the current vector activity count.

VSS

displays the vector section size.

Usage Notes

No blanks are allowed between operands that relate to a single display request.

DISPLAY VECTOR

Response 1

VRnn,elt = xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

The vector register elements are always displayed in groups of four with the contents shown in hexadecimal. If required, multiple lines are displayed for each register.

where:

nn
indicates the register whose contents are displayed.

elt
indicates the first element number whose contents are displayed.

Response 2

VPnn,elt = xxxxxxxxxxxxxxxx s.dddddddddddddd Esdd

The contents of the register pair is displayed in both the hexadecimal format (the first part of the response) and in exponential decimal format (the second part of the response). One element of a register pair is displayed per line. If required, multiple lines are displayed for each register pair.

where:

nn
indicates the register whose contents are to be displayed.

elt
indicates the first element number whose contents are to be displayed.

s
is the positive or negative indicator.

Response 3

VSR = xxxxxxxx xxxxxxxx

The vector status register contents are displayed in hexadecimal.

Response 4

VMR = xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

The vector mask register contents are displayed in hexadecimal.

Response 5

VAC = xxxxxxxx xxxxxxxx

The vector activity count contents are displayed in hexadecimal.

Response 6

VECTOR SECTION SIZE: nnn

The vector section size is displayed in hexadecimal.

Response 7

VX_n = ZERO (IN-USE BIT IS OFF)

This response is given when the “in-use” bit of the referenced vector pair in the vector status register is off for a display vector register request. When the in-use bit is off, the register contents are defined at zero.

X

is the indicator for “R” (register) or “P” (pair).

n

is the hexadecimal register number.

Migration Notes

VM/SP HPO: VM/XA SP evaluates the command for proper syntax prior to checking whether the Vector Facility is available.

DRAIN**Privilege Class: D**

Use the DRAIN command to stop spooling operations on the specified real unit record devices after the file currently being processed has completed.

DRain	<div style="border-left: 1px solid black; border-right: 1px solid black; border-bottom: 1px solid black; padding: 5px;"> Reader Printer PUnch rdev rdev ... ALL </div>
-------	--

where:

READER**RDR****PRINTER****PRT****PUNCH****PCH**

drains all readers, printers, or punches.

rdev rdev ...

is a real device number or list of real device numbers of the real spooled devices that you want drained.

ALL

drains all spooled unit record devices in the system.

Usage Notes

1. Use the DRAIN command to bring the spooling system or a specified device to a controlled halt. You would normally drain all unit record devices before a system shutdown, and drain a printer before you change the contents of the UCS printer buffer.
2. A device is drained immediately if it is not active when the DRAIN command is issued; otherwise, it is drained when processing of the current file has completed. You can restart a drained device by issuing the CP START command. If you issue a START command to a device before draining has been completed, the device does not enter drained status and it continues processing.
3. If you specify more than one real device number, execution of the DRAIN command terminates upon encountering an invalid real device number. None of the remaining devices are drained.

Responses

Inactive Printer Responses

The following response occurs for each printer that attains the drained status:

```
PRT rdev DRAINED userid CLASS c ...

PRT rdev FORM form { AUTO } { SEP } { NO3800 }
                   { MANUAL } { NOSEP } { BEG3800 }
                   { SETUP } { ANY3800 }
```

For 3800 printers only (third line):

```
PRT rdev FLASH name IMAGE imagelib CHARS name FCB { name } { HOLD }
                                                    { 6 } { PURGE }
                                                    { 8 }
                                                    { 10 }
                                                    { 12 }
```

For impact printers only (third line):

```
PRT rdev { NOFOLD } IMAGE imagelib CHARS chars [FCB name] [INDEX nn]
          { FOLD }
```

Inactive Punch Responses

The following response occurs for each punch that attains the drained status:

```
{PUN} rdev DRAINED userid CLASS c ...

{PUN} rdev FORM form { AUTO } { SEP }
                    { MANUAL } { NOSEP }
                    { SETUP }
```

Inactive Reader Response

```
{RDR} rdev DRAINED SYSTEM
```

Active Printer Responses

The following response occurs for each printer that attains the drained status:

```
PRT rdev DRAINING userid CLASS c ...

PRT rdev FORM form { AUTO } { SEP } { NO3800 }
                   { MANUAL } { NOSEP } { BEG3800 }
                   { SETUP } { ANY3800 }
```

DRAIN

For 3800 printers only (3rd and 4th lines):

```
PRT rdev FLASH name IMAGE imagelib CHARS name FCB { name } { HOLD }
                                                { 6 } { PURGE }
                                                { 8 }
                                                { 10 }
                                                { 12 }
```

```
PRT rdev {PRINTING} userid FILE spoolid RECS { nnnn } COPY [*]nnn SEQ sss
                                                { nnnK }
                                                { nnnM }
```

For impact printers only (3rd and 4th lines):

```
PRT rdev {NOFOLD} IMAGE imagelib CHARS chars [FCB name] [INDEX nn]
        { FOLD }
```

```
PRT rdev {PRINTING} userid FILE spoolid RECS { nnnn } COPY [*]nnn SEQ sss
                                                { nnnK }
                                                { nnnM }
```

is the message you receive when the system begins processing a spool file that has the MSG attribute.

Active Punch Responses

The following response occurs for each punch that attains the drained status:

```
{PUN} rdev DRAINING userid CLASS c ...
```

```
{PUN} rdev FORM form { AUTO } { SEP }
                      { MANUAL } { NOSEP }
                      { SETUP }
```

```
{PUN} rdev {PUNCHING} userid FILE spoolid RECS { nnnn } COPY [*]nnn SEQ sss
                                                { nnnK }
                                                { nnnM }
```

Active Reader Response

```
{RDR} rdev DRAINING SYSTEM
```

where:

PRT
PUN
RDR

is the device that you drained or are draining.

rdev

is the real device number of the drained or draining device.

CLASS c...

are those output classes that the drained or draining device was servicing. One to eight classes are displayed for each printer or punch device that was drained or is draining.

FORM form

changes the form name of a spool file. The form name is one-to-eight characters.

AUTO

indicates that the operator wants CP to schedule forms output. The currently active form is processed first. When all files with the current form have been printed or punched, CP automatically selects a new form for the device. The operator will receive a message to change forms, and the output device will be idle. The operator can then put in the new form and press the START button (or issue the START command) to print files with the new form.

MANUAL

indicates that the operator will schedule output forms. Only spool files with the form currently active on the device are processed. When the last spool file with the current form is printed or punched, the output device stops. The operator can then schedule a new form by issuing a START command with the FORM option.

SETUP

indicates that the operator wants CP to print a page so that forms alignment can be verified. (This option is only valid for impact printers.) The operator will get a message indicating that he or she should set up the forms. When the form has been set up, the operator should press the STOP button, then the START button. An alignment page will then be printed. It is the first page of the file, with letters replaced with X's and the numbers replaced by 9's. Each time the operator presses the STOP or START buttons after a full alignment page has been printed, another alignment page will be printed. When the forms are aligned, you should issue the "START" command for the device. The printer will then return to its previous mode (AUTO or MANUAL) and the file will print in its entirety.

If neither MANUAL, AUTO, nor SETUP is specified and the printer is currently in SETUP mode, it will return to the previous mode, AUTO or MANUAL. If no START command has been issued with the MANUAL, AUTO, or SETUP options since a system cold start, the default is MANUAL.

SEP**NOSEP**

indicates the current separator status.

NO3800**BEG3800****ANY3800**

specifies the restrictions on the 3800 load CCWs in effect once the device is STARTed again.

rdev

is the real 3800 printer or list of 3800 printer numbers that you want drained.

FLASH name

signifies that a form overlay contained in the 3800 printer is to be superimposed (or flashed) onto the specified number of copies of the output. The name, to be one-to-four characters in length, identifies the forms overlay to be used in the projection.

IMAGE imagelib

specifies the name of the system image library file which will contain the load images (character sets), FCBs, graphic character modifications, or copy modifications that will be used for printing files and separators. "imagelib" may be from one-to-eight characters in length, and must be an existing image library file.

FCB name

is the name of the forms control buffer (FCB) module to be used by the printer in the vertical formatting of a page.

Note: This does not appear for 1403 type printers.

CHARS name

specifies the name of the default character set used when printing spool files which do not specify a character set.

HOLD

prevents the printing of a file by placing it in a user HOLD status.

PURGE

closes and purges the spool file on the specified 3800 printer regardless of the CONT, KEEP, and HOLD setting for the device.

NOFOLD**FOLD**

indicates whether lowercase characters are being translated (folded) into uppercase for printing.

CHARS chars

indicates the character set being used by the printer.

INDEX nn

indicates the character position on the page at which to begin printing.

Note: This only appears for 3211 type printers.

PRT**PUN**

indicate the type of device on which the file is being processed.

rdev

is the real device number.

userid

is the name of the spool file owner.

FILE spoolid

is the spool file spoolid number.

RECS $\left\{ \begin{array}{l} \text{nnnn} \\ \text{nnnK} \\ \text{nnnM} \end{array} \right\}$

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

specifies the number of file copies you want printed or punched when the file is produced.

When the optional asterisk (*) is specified, the 3800 printer will print each page from the spooled output files nnn times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (*) is ignored and duplication occurs normally.

SEQ sss

is the sequence number for the file on the printer or punch.

Migration Notes*VM/SP HPO*

1. The VM/XA SP response provides more detail (two lines) than VM/SP HPO concerning the printer being drained.
2. VM/XA SP does not support logical printers.

VM/XA SF: A third line appears for impact printers other than the 1403 without the UCS feature.

DUMP

Use the DUMP command described in this section to dump the following real and virtual machine components to your virtual printer:

- Contents of first-level storage
- Contents of second-level storage
- Contents of third-level storage
- Old and new PSWs and interrupt information for the following interrupts:

- SVC
 - I/O
 - PROGRAM
 - EXTERNAL
 - MACHINE CHECK

- Storage keys
- Contents of registers:
 - Prefix
 - General
 - Floating-point
 - Control
- Subchannel information blocks (SCHIBs)
- The current program status word (CURRENT PSW)
- The following Vector Facility information:

- Vector registers (VR)
 - Vector register pair(s) (VP)
 - Vector status register (VSR)
 - Vector mask register (VMR)
 - Vector activity count (VAC)
 - Vector section size (VSS).

Complete Format for DUMP

The format box below shows all of the operands available with the DUMP command. Complete descriptions of the operands appear on the following pages, in the same sequence as the operands appear here.

DUMP

General Usage Notes for DUMP

1. The DUMP command prints the contents of various storage locations, registers, and control registers. You may be able to detect the cause of program errors by examining this information.
2. Using a single DUMP command, you can dump multiple storage locations, registers, and control words.

There are two rules that you must follow in order for the command to execute properly:

- a. Do not put blanks between operands that relate to a single DUMP request, except for PSW and SCHIB which require at least one blank between the first and second operands:

For example, if you want to dump hexadecimal location 40, enter:

```
dump h40
```

- b. Separate each set of dump requests from others by at least one blank.

For example, if you want to dump the hexadecimal location 40 and the contents of register 5, enter:

```
dump h40 g5
```

3. You can assign a dump identification to any dump that you create by means of the DUMP command. Note, however, that even though you can use a single DUMP command to make multiple DUMP requests, you can assign only one dump identification per command.

The dump identifier is an asterisk (*) followed by a 1- to 97-character identifier that is to be assigned to the dump. If specified, the dump identifier must be the last operand you enter on the command line. Blanks are permitted in the dump identification.

For example:

```
dump g *general registers
```

creates a dump of your general registers, and assigns the dump the identifier "general registers."

4. As each request is processed, a message is sent to your display device. This will permit you to terminate DUMP processing by pressing the PA1 key. When the command terminates, you may enter another command.

5. The defaults for the following DUMP operands are:

Command Operand	Default Before Rounding
G	0-end
G-	0-end
G-5	0-G5
G5-	G5-end
G5	G5
T	error
T-	T0-end
T-1000	T0-1000
T1000-	T1000-end
T1000	T1000
PSW	current

6. When dumping a range of addresses or subchannels, the command operand must be followed by an address, a range, or a punctuation mark (“-”, “:”, “.”), which will form a range. When dumping registers, the command operand can stand alone.
7. If you enter a command line that contains an invalid operand, the DUMP command will terminate when it encounters that operand. Any previous valid operands will be processed before termination occurs.
8. The DUMP command allows you to dump information in the following formats:
 - Hexadecimal string/Hexadecimal word segments
 - EBCDIC translation for uppercase letters and numbers 0-9
 - Hexadecimal word segments with EBCDIC translation
 - Instruction
 - Storage key.

Note: Hexadecimal to EBCDIC translation is done for the numerals 0 to 9 and uppercase characters A to Z only.

If you enter multiple operands on a single command line for a storage or register dump, and do not specify the format that you want the DUMP command processor to use, the last format type that you explicitly entered on the command line will define the type of dump that you receive. That is, whenever you explicitly enter a dump type, you implicitly define the default for subsequent operands of the current dump function. If you do not enter any dump format type on the command line, the system defaults will be employed.

For example:

```
dump 10 20 T40 80 G12 5 L60-100
```

dumps the following, respectively:

- Hexadecimal location 10 without EBCDIC translation
- Hexadecimal location 20 without EBCDIC translation
- Hexadecimal location 40 with EBCDIC translation
- Hexadecimal location 80 with EBCDIC translation
- The contents of general register 12
- The contents of general register 5
- Hexadecimal locations 60 through 100 with EBCDIC translation

DUMP

The DUMP command suppresses identical lines and imbeds a response whenever a page boundary is crossed.

Migration Notes

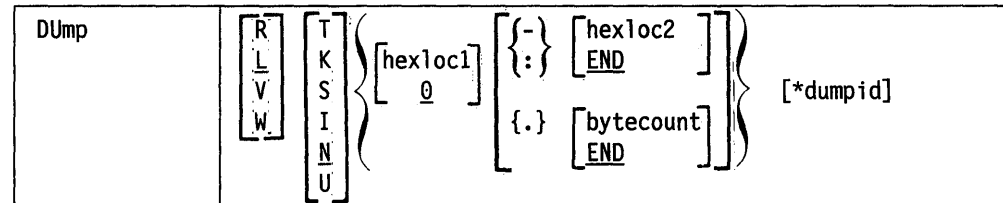
VM/SP HPO

1. In VM/XA SP, you can store into CAW or CSW but you cannot dump their contents.

DUMP GUEST STORAGE

Privilege Class: G

Use DUMP GUEST STORAGE to dump the contents of specified locations in virtual machine storage.



Note: Do not put blanks between operands that relate to a single DUMP request (except for PSW and SCHIB). Note that you can assign only one dump identifier (*dumpid) per command. For more information, see “General Usage Notes for DUMP” on page 158.

where:

R

dumps the contents of second-level storage.

L

dumps the contents of second-level storage if the guest virtual machine is operating in DAT OFF mode; dumps the contents of third-level storage if the guest virtual machine is operating in DAT ON mode.

V

dumps the contents of third-level storage using control register 1.

W

dumps the contents of third-level storage using control register 7.

T

dumps the contents of storage in hexadecimal with EBCDIC translation. Storage is dumped in word segments. An address range will be dumped starting at the X'10' boundary in which the specified address resides. Storage is dumped in X'10' increments.

The storage key is dumped on the first line and at every page boundary.

K

dumps the storage keys in hexadecimal for all page frames or partial page frames included in the specified address range.

S

dumps the contents of storage in hexadecimal translation. Storage is dumped as a string starting at the location specified. The specified address range is not boundary aligned.

The storage key is dumped on the first line and at every page boundary.

I

dumps the contents of storage in instruction format. The specified address is rounded down to the nearest half word boundary.

The storage key is dumped on the first line and at every page boundary.

N

dumps the contents of storage in hexadecimal without EBCDIC translation. This is the default. Storage is dumped in word segments. An address range will be dumped starting at the full word in which the specified address resides. Storage is dumped in full word increments.

The storage key is dumped on the first line and at every page boundary.

U

dumps the contents of storage in EBCDIC translation. Storage is dumped as an EBCDIC string starting at the address specified. The specified address range is not boundary aligned.

The storage key is dumped on the first line and at every page boundary.

If you do not specify any of the options T, K, S, I, N, or U, the contents of storage are dumped in hexadecimal word segments.

hexloc1

is the starting storage location that is to be dumped to your virtual printer. The address hexloc1 may be one-to-eight hexadecimal digits; leading zeros are optional. The default is 0. The starting address will be translated to a hex 10 boundary.

Storage locations and the minimum amount dumped are rounded according to the following conventions:

Rounding Conventions (in hex)

Storage Locations	Start	Minimum Length
T	10	10
K	7FF	7FF
S	1	1
I	2	2-6 (depending on instruction)
N	4	4
U	1	1

**{-} hexloc2
{:} END**

is the last of a range of hexadecimal storage locations whose contents are to be dumped to your virtual printer. If specified, hexloc2 must be equal to or greater than hexloc1 and within the virtual storage size. The address, hexloc2, may be from one-to-eight hexadecimal digits; leading zeros are optional.

Specify END if you want to dump the remaining contents of storage. END is the default if you specify “-”, “:”, or “.”.

The symbol “:” means the same as “-” when specifying a range of addresses.

**{.} bytecount
END**

is a hexadecimal number designating the number of bytes of storage (starting with the byte at hexloc1) to be dumped to your virtual printer.

The specified bytecount must have a value of at least one and may be from one to eight hexadecimal digits; leading zeros are optional.

Specify END if you want to dump the remaining contents of storage. END is the default if you specify “-”, “:”, or “.”.

***dumpid**

is an asterisk followed by a 1- to 97-character identifier that is to be assigned to the dump. If specified, the dumpid must be the last operand you enter on the command line.

Usage Notes

1. The DUMP GUEST STORAGE command dumps the program status words (PSWs) and the contents of registers (general, floating-point, control, and prefix) along with the contents of the specified storage. If your virtual machine is operating in 370-XA mode, the subchannel information blocks (SCHIBs) are dumped. If you define a Vector Facility, then the vector registers (VR), vector register pairs (VP), vector status register (VSR), vector mask register (VMR), vector activity count (VAC), and vector section size (VSS) are dumped.
2. You can dump storage locations outside the range of your virtual machine size if they are within a saved segment. Any attempt to dump locations (or associated keys) between the upper limit of your virtual machine and the address at which the saved segment is saved results in the "nonaddressable storage" response appearing in the printed output.

Responses

One or more of the following responses appears in your spooled output, depending upon the operands that you specified:

Response 1

Xhexloc word1 word2 word8 [Key] [*EBCDIC TRANSLATION*] [Lhexloc]

This is the response you receive if your request was L, R, V, or W with the T or N option. X indicates the type of request (L, R, V, or W). The EBCDIC translation is given only for the T option.

The storage key will be dumped on the first line, and at every page boundary.

For a Vhexloc or Whexloc request, the Lhexloc trailer shows the guest real address corresponding to the given guest virtual address. The Lhexloc trailer is also shown when DAT-ON mode causes Lhexloc to resolve to a third-level address. The storage key is dumped on the first line of a request and at every page boundary.

Response 2

Xhexloc *EBCDIC TRANSLATION* [key] [Lhexloc]

This is the response you receive when your request is L, R, V, or W, with the U option. Xhexloc is the hexadecimal location of the first data byte to be dumped; X can be L, R, V, or W, depending on your request. A minimum of 1 character and a maximum of 64 characters appear on each line. The contents of storage are dumped in EBCDIC translation only. Periods represent nonprintable characters. The storage key is dumped on the first line of a request and at every page boundary.

For a Vhexloc or Whexloc request, the Lhexloc trailer shows the guest real address corresponding to the given guest virtual address. The Lhexloc trailer is also shown when DAT-ON mode causes Lhexloc to resolve to a third-level address. The storage key is dumped on the first line of a request and at every page boundary.

DUMP GUEST STORAGE

Response 3

```
Xhexloc mnemonic X-X mnemonic X-X mnemonic X-X [key] [Lhexloc]
```

This is the response you receive when your request is L, R, V, or W, with the I option. Xhexloc is the hexadecimal storage location of the first instruction to be dumped and X can be L, R, V, or W, depending on the request. Mnemonic is up to a 5-character representation of the instruction code. X-X is the actual hexadecimal contents of storage and is either 2, 4, or 6 bytes as determined by the instruction length code. Unrecognizable instruction codes are indicated by a series of asterisks. The minimum of one instruction and a maximum of five instructions are dumped per line. The storage key is dumped on the first line of a request and at every page boundary.

For a Vhexloc or Whexloc request, the Lhexloc trailer shows the guest real address corresponding to the given guest virtual address. The Lhexloc trailer is also shown when DAT-ON mode causes Lhexloc to resolve to a third-level address. The storage key is dumped on the first line of a request and at every page boundary.

Response 4

```
Xhexloc stringhexadecimaldata [key][Lhexloc]
```

This is the response you receive if your dump request was L, R, V, or W with the S option. X indicates the type of dump request (L, R, V, or W). The storage key is dumped on the first line of a request, and at every page boundary.

For a Vhexloc or Whexloc request, the Lhexloc trailer shows the guest real address corresponding to the given guest virtual address. The Lhexloc trailer is also shown when DAT-ON mode causes Lhexloc to resolve to a third-level address. The storage key is dumped on the first line of a request and at every page boundary.

Response 5

```
Xhexloc1 TO hexloc2 KEY = kk
```

This is the response when you dump storage keys; hexloc1 and hexloc2 are storage locations and kk is the associated storage key for Xhexloc1. X can be L, R, V, or W, depending upon your dump request. A response is dumped for every contiguous address range that has unique storage keys, and whenever a segment boundary is crossed.

Response 6

If your request includes a request to dump nonaddressable storage, you receive the following response:

```
Xhexloc1 TO hexloc2 NON-ADDRESSABLE STORAGE
```

hexloc1 and hexloc2 are storage locations. The X in Xhexloc can be L, R, V, or W, depending on the request.

Response 7

If your request includes a request to dump one or more lines identical to the previous line, you receive the following response:

```
Xhexloc1 TO hexloc2 SUPPRESSED LINE(S) SAME AS ABOVE ...
```

hexloc1 and hexloc2 are storage locations. The X in Xhexloc can be L, R, V, or W, depending on the request.

Response 8

Responses 8 and 9 appear on your display screen.

DUMPING LOC Xhexloc

This response is sent to your console to indicate that a segment boundary has been reached. Each segment boundary location has an address that ends in X'0000'.

In this response, hexloc is the storage boundary that has been reached, and X can be L, R, V, or W, depending upon your dump request.

Response 9

COMMAND COMPLETE

indicates that the command has completed.

Migration Notes*VM/SP HPO*

1. In VM/XA SP, you can dump larger storage sizes than in VM/SP HPO.
2. In VM/XA SP, you can dump more levels of storage than in VM/SP HPO.
3. VM/XA SP also dumps subchannel information blocks (SCHIBs) when in 370-XA mode, and all vector information when a Vector Facility is defined.

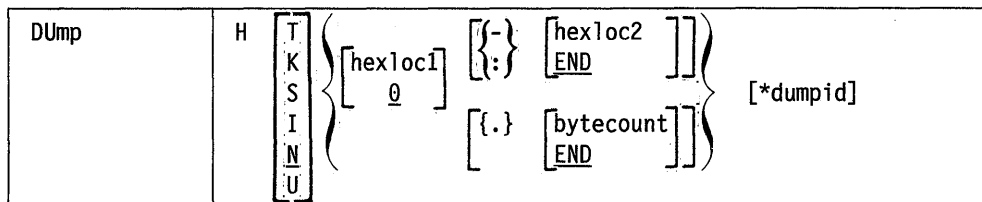
VM/XA SF

1. In VM/XA SP, the response SUPPRESSED LINE(S) SAME AS ABOVE appears whether or not the amount of storage to be dumped exceeds X'100' bytes.
2. In VM/XA SP, storage keys are dumped at the first line of a request and at every page boundary. It shows storage keys even if you request only one address.
3. In VM/XA SP, the default option is N.
4. In VM/XA SP, the N option rounds storage locations to fullword boundaries, and dumps storage in fullword increments.

DUMP HOST STORAGE

Privilege Class: C,E

Use DUMP HOST STORAGE to dump the contents of first-level storage in specified locations.



Note: Do not put blanks between operands that relate to a single DUMP request, except for PSW and SCHIB which require at least one blank between the first and second operands. Note that you can assign only one dump identifier (**dumpid*) per command. For more information, see “General Usage Notes for DUMP” on page 158.

where:

H

dumps the contents of first-level storage.

T

dumps the contents of storage in hexadecimal with EBCDIC translation. Storage is dumped in word segments. An address range will be dumped starting at the X'10' boundary in which the specified address resides. Storage is dumped in X'10' increments.

The storage key is dumped on the first line and at every page boundary.

K

dumps the storage keys in hexadecimal for all page frames or partial page frames included in the specified address range.

S

dumps the contents of storage in hexadecimal translation. Storage is dumped as a string starting at the location specified. The specified address range is not boundary aligned.

The storage key is dumped on the first line and at every page boundary.

I

dumps the contents of storage in instruction format. The specified address is rounded down to the nearest half word boundary.

The storage key is dumped on the first line and at every page boundary.

N

dumps the contents of storage in hexadecimal without EBCDIC translation. This is the default. Storage is dumped in word segments. An address range will be dumped starting at the full word in which the specified address resides. Storage is dumped in full word increments.

The storage key is dumped on the first line, as well as at every page boundary.

U

umps the contents of storage in EBCDIC translation. Storage is dumped as an EBCDIC string starting at the address specified. The specified address range is not boundary aligned.

The storage key is dumped on the first line and at every page boundary.

If you do not specify any of the options T, K, S, I, N, or U, the contents of storage are dumped in hexadecimal word segments.

hexloc1

is the starting storage location that is to be dumped to your virtual printer. The address hexloc1 may be one to eight hexadecimal digits; leading zeros are optional. The default is 0.

**{-} hexloc2
{:} END**

is the last of the range of hexadecimal storage locations whose contents are to be dumped to your virtual printer. If specified, hexloc2 must be equal to or greater than hexloc1 and within the bounds of addressable storage (the lesser of the storage size specified at system generation and the real storage size). The address, hexloc2, may be from one to eight hexadecimal digits; leading zeros are optional.

Specify END if you want to dump the remaining contents of storage. END is the default.

The symbol “:” means the same as “-” when specifying a range of addresses.

**{.} bytecount
END**

is a hexadecimal integer designating the number of bytes of storage (starting with the byte at hexloc1) to be dumped to your virtual printer. The sum of hexloc1 and bytecount must be an address that does not exceed the bounds of addressable storage (the lesser of the storage size specified at system generation and the real storage size). The specified bytecount must have a value of at least one and may be from one to eight hexadecimal digits. Leading zeros are optional.

Specify END if you want to dump the remaining contents of storage. END is the default.

***dumpid**

is an asterisk followed by a 1- to 97-character identifier that is to be assigned to the dump. If specified, the dumpid must be the last operand you enter on the command line.

Usage Notes

You can dump only CP owned pages.

While dumping data in CP storage, the running of a virtual machine or someone using the STORE HOST command could cause data to be changed. This could create an inconsistency in the data that is viewed.

DUMP HOST STORAGE

Responses

One or more of the following responses appears in your spooled output, depending upon the operands specified.

Response 1

```
Hhexloc word1 word2 ..... word8 [Key] [*EBCDIC TRANSLATION*]
```

This is the response for the storage location requests when H is your dump request with the T or N option. The EBCDIC translation is only dumped if you have also entered the T option.

The storage key is dumped on the first line, and at every page boundary.

Response 2

```
Hhexloc *EBCDIC TRANSLATION* [key]
```

If your request is H with the U option, you receive this response. Hhexloc is the hexadecimal location of the first data byte to be dumped. A minimum of 1 character and a maximum of 64 characters are dumped per line. The contents of storage will be dumped only in EBCDIC translation. Periods represent nonprintable characters. The storage key is dumped on the first line and at every page boundary.

Response 3

```
Hhexloc mnemonic X-X mnemonic X-X mnemonic X-X [key]
```

If your request is H with the I option, you receive this response. Hhexloc is the hexadecimal storage location of the first instruction to be dumped. mnemonic is up to a 5-character representation of the instruction code. X-X is the actual hexadecimal contents of storage and is either 2, 4, or 6 bytes as determined by the instruction length code. Unrecognizable instruction codes are indicated by a series of asterisks. A minimum of one instruction and a maximum of five instructions are dumped per line. The storage key is dumped on the first line and at every page boundary.

Response 4

```
Hhexloc stringhexadecimaldata [key]
```

This is the response you receive if your dump was H with the S option. The storage key is dumped on the first line and at every page boundary.

Response 5

```
Hhexloc1 TO hexloc2 KEY = kk
```

If your request is H with the K option you receive this response. hexloc1 and hexloc2 are storage locations and kk is the associated storage key. You will receive this response for every contiguous address range that has unique storage keys, and whenever a segment boundary is crossed.

Response 6

If the storage location you indicated in the command line is not a CP owned page, you receive the following response:

```
Hhexloc1 TO hexloc2 SUPPRESSED NON-CONTROL-PROGRAM PAGE
```

hexloc1 and hexloc2 are storage locations.

Response 7

If your request includes a request to dump one or more lines identical to the previous line, you receive the following response:

```
Xhexloc1 TO hexloc2 SUPPRESSED LINE(S) SAME AS ABOVE ...
```

hexloc1 and hexloc2 are storage locations. The X in Xhexloc can be L, R, V, or W, depending on the request.

Response 8

Responses 8 and 9 appear on your display screen.

```
DUMPING LOC Hhexloc
```

This response is sent to your console to indicate that a segment boundary has been reached. Each segment boundary has an address that ends in X'0000'.

Response 9

```
COMMAND COMPLETE
```

indicates that the command has successfully executed.

Migration Notes

VM/XA SF

1. In VM/XA SP, the response SUPPRESSED LINE(S) SAME AS ABOVE appears whether the amount of storage to be dumped exceeds X'100' bytes.
2. VM/XA SP dumps storage keys at the first line of a range and at every page boundary. It shows storage keys even if you request only one address.
3. In VM/XA SP, the default format is N.
4. In VM/XA SP, the N option rounds storage locations to fullword boundaries and dumps storage in fullword increments.

DUMP PREFIX

Privilege Class: G

Use DUMP PREFIX to dump the contents of the prefix register to your virtual printer.

Dump	PREFIX [*dumpid]
------	------------------

where:

PREFIX

 dumps the contents of the prefix register.

***dumpid**

 is an asterisk followed by a 1- to 97-character identifier that is to be assigned to the dump. If specified, the dumpid must be the last operand you enter on the command line.

Usage Notes

The contents of the prefix register is the absolute location of the guest page 0.

Response 1

PREFIX = xxxxxxxx

is the contents of the prefix register in hexadecimal.

The following response will appear on your display screen.

Response 2

COMMAND COMPLETE

indicates that the command has successfully executed.

DUMP PROGRAM STATUS WORDS

Privilege Class: G

Use DUMP PROGRAM STATUS WORDS to dump information about your virtual machine's program status words (PSWs) as two hexadecimal words.

DUmp	Psw	EXT SVC PROG PRG MCH I/O ALL	[*dumpid]
------	-----	--	-----------

where:

PSW

requests the dump of a current virtual machine PSW. When entered without operands, the value of the current PSW will be dumped.

EXT

SVC

PROG

PRG

MCH

I/O

umps the current values and appropriate interrupt information for both the new and old PSWs specified, that is, EXTERNAL, SVC, PROGRAM CHECK, MACHINE CHECK, and I/O.

ALL

umps the current values and appropriate interrupt information for the current PSW and all old and new interrupt PSWs (except RESTART).

*dumpid

is an asterisk followed by a 1- to 97-character identifier that is to be assigned to the dump. If specified, the dumpid must be the last operand you enter on the command line.

Responses

One or more of the following responses appears in your spooled output, depending upon the operands specified.

Response 1

Dumping the Current PSW

PSW = xxxxxxxx xxxxxxxx

The contents of the PSW are dumped in hexadecimal.

DUMP PROGRAM STATUS WORDS

Response 2

```
EXT nnnn 18 OLD XXXXXXXX XXXXXXXX 58 NEW XXXXXXXX XXXXXXXX
SVC nnnn 20 OLD XXXXXXXX XXXXXXXX 60 NEW XXXXXXXX XXXXXXXX
PRG nnnn 28 OLD XXXXXXXX XXXXXXXX 68 NEW XXXXXXXX XXXXXXXX
MCH nnnn 30 OLD XXXXXXXX XXXXXXXX 70 NEW XXXXXXXX XXXXXXXX
I/O nnnn 38 OLD XXXXXXXX XXXXXXXX 78 NEW XXXXXXXX XXXXXXXX
```

The contents of the old and new PSWs and the interruption codes are dumped in hexadecimal. The nnnn represents the interruption code. The numbers 18, 20, 28... are storage locations. The X's represent the contents of the PSWs.

The following response will appear on your display screen.

Response 3

```
COMMAND COMPLETE
```

indicates that the command has successfully executed.

DUMP REGISTERS

Privilege Class: G

Use DUMP REGISTERS to dump the contents of general purpose registers, floating point registers, or control registers.

DUmp	$\left. \begin{array}{c} \{G\} \\ \{Y\} \\ \{X\} \end{array} \right\} \text{reg1}$	$\left[\begin{array}{l} \{-\} \text{reg2} \\ \{:\} \text{END} \\ \{.\} \text{regcount} \\ \{.\} \text{END} \end{array} \right]$	[*dumpid]
------	--	--	-----------

Note: Do not put blanks between operands that relate to a single DUMP request, except for PSW and SCHIB which require at least one blank between the first and second operands. Note that you can assign only one dump identifier (**dumpid*) per command. For more information, see “General Usage Notes for DUMP” on page 158.

where:

Greg1

is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, general register whose contents are to be dumped to your virtual printer. If you specify G without a register number, the contents of all the general registers are dumped.

Yreg1

is an integer (0, 2, 4, or 6) representing the first, or only, floating-point register whose contents are to be dumped to your virtual printer. If you specify Y without a register number, the contents of all of the floating-point registers are dumped.

Xreg1

is a decimal number from 0 to 15 or a hexadecimal number from 0 to F representing the first, or only, control register whose contents are to be dumped to your virtual printer. If you specify X without a register number, the contents of all of the control registers are dumped.

$\left. \begin{array}{l} \{-\} \text{reg2} \\ \{:\} \text{END} \end{array} \right\}$

is a number representing the last register whose contents are to be dumped. You must specify the hyphen “-” or “:” when you are requesting more than one register by register number. You must specify the range of registers in ascending order. Therefore, the value you specify for reg2 must be equal to or greater than the value you specify for reg1.

If you specify G or X, reg2 may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F. If you specify Y, reg2 may be 0, 2, 4, or 6. The contents of registers reg1 through reg2 are dumped to your virtual printer.

The symbol “:” means the same as “-” when specifying a range of addresses.

DUMP REGISTERS

Specify END to dump the contents of all remaining registers. END is the default. Remaining registers are those in the range:

- From reg1 through register 15, if you are dumping general registers
- From reg1 through register 6, if you are dumping floating point registers
- From reg1 through register 15, if you are dumping control registers.

{.} **regcount**

END

is a decimal number from 1 to 16 or a hexadecimal number from 1 to F specifying the number of registers (starting with reg1) whose contents are to be dumped. Specify END to dump the contents of all remaining registers. END is the default. Remaining registers are as follows:

- General registers 1-15
- Floating point registers 1-7
- Control registers 1-15.

If the dump type G or X is specified, regcount can be a decimal number from 1 to 16 or a hexadecimal number from 1 to F.

If dump type Y is specified, regcount must be 1, 2, 3, or 4. The sum of reg1 and regcount must be a number that does not exceed the maximum register number for the type of registers being dumped.

***dumpid**

is an asterisk followed by a 1- to 97-character identifier that is to be assigned to the dump. If specified, the dumpid must be the last operand you enter on the command line.

Responses

One or more of the following responses appears in your spooled output, depending upon the operands that you specified:

Response 1 -- Dumping General Registers

GPR nn = genreg1 genreg2 genreg3 genreg4

This is the response you receive when you dump general registers. nn indicates the first (or only) register whose contents are to be dumped; the contents of nn are dumped in genreg1. The contents of the registers are dumped in hexadecimal. Up to four registers per line are dumped for a range of registers. Multiple lines are dumped if required, with a maximum of four lines needed to dump all 16 general registers.

Response 2 -- Dumping Floating-Point Registers

FPR nn = xxxxxxxxxxxxxxxx ±.dddddddddddddd E dd

This is the response you receive when you dump floating-point registers. nn is the even-number floating-point register whose contents are dumped on this line. The contents of the requested floating-point registers are dumped in both the internal hexadecimal format and the E format. One register is dumped per line. Multiple lines are dumped for a range of registers. The first part of the response is the value in hexadecimal and the second part of the response is in exponential decimal format.

Response 3 -- Dumping Control Registers

```
ECR nn = ctlreg1 ctlreg2 ctlreg3 ctlreg4
```

This is the response you receive when you dump control registers. nn is the first or only control register whose contents you want to dump. The contents of the following consecutive registers are ctlreg2, ctlreg3, and so on. The contents of the requested control registers are dumped in hexadecimal. Up to four registers per line are dumped. Multiple lines are dumped if required, up to a maximum of four lines needed to dump all 16 control registers.

Response 4

```
COMMAND COMPLETE
```

indicates that the command has successfully executed.

Migration Notes

VM/XA SF: The VM/XA SP dump output file uses GPR, FPR, and ECR instead of G, Y, and X as in VM/XA SF.

DUMP SCHIB

Privilege Class: G

Use DUMP SCHIB to dump information from your subchannel information blocks (SCHIBs). This command is only valid if your virtual machine is operating in 370-XA mode.

DUp	SCHib	[vsch1]	{ - }	[vsch2]	} [*dumpid]
			{ : }	[END]	
			{ . }	[count]	
		} ALL			

Note: You can assign only one dump identifier (**dumpid*) per command. For more information, see “General Usage Notes for DUMP” on page 158.

where:

SCHIB

 dumps information from your subchannel information blocks.

vsch1

 is the starting (or only) virtual subchannel whose SCHIB is to be dumped.

{ - } vsch2
{ : } END

 specifies that all SCHIBs in the range vsch1 to vsch2 are to be dumped. Specify END to dump all SCHIBs from vsch1 to the highest number subchannel. END is the default.

 The symbol “:” means the same as “-” when specifying a range.

{ . } count
 END

 specifies the number of consecutive SCHIBs starting at vsch1 that are to be dumped. Specify END to dump all SCHIBs from vsch1 to the highest number subchannel. END is the default.

ALL

 specifies that SCHIBs for all virtual devices are to be dumped.

***dumpid**

 is an asterisk followed by a 1- to 97-character identifier that is to be assigned to the dump. If specified, the dumpid must be the last operand you enter on the command line.

Usage Notes

1. If you specify a list of SCHIBs and one is not found, the DUMP command will terminate. For example, if you enter:

```
dump SCHIB 0022 0055 0011
```

and subchannel 0055 is not found, the command will terminate. If you specify a range of SCHIBs, the DUMP command will terminate if the first SCHIB in the range is not found. Otherwise, all those SCHIBs that were found will be dumped.

2. If your virtual machine is running in 370-XA mode, you can use the CP QUERY VIRTUAL command to determine the subchannel number associated with a virtual device.
3. The response to this command is a header line followed by a line of information for each SCHIB that you specify. (The header line identifies, column by column, the SCHIB information that follows it. See "Responses" below.) After every ten lines of SCHIB information, the header line reappears.

Response 1

The following response appears in your spooled output.

```
SCHIB DEV INT-PARM ISC FLG LP PNO LPU PI MBI PO PA CHPID0-3 CHPID4-7
nnnn nnnn nnnnnnnn n nn nn nn nn nn nnnn nn nn nnnnnnnn nnnnnnnn
```

If an interrupt is pending for the virtual subchannel, the following additional line will be output to describe the pending interrupt:

```
KEY ECC FPI ZN SCTL CCW-ADDR DS SS CNT EXT-LOG
n n n n nnnn nnnnnnnn nn nn nnnn nnnnnnnn
```

where:

INT-PARM	is the interrupt parameter.
ISC	is the interrupt subclass.
FLG	is the flag field.
LP	is the logical path mask.
PNO	is the path not operational mask.
LPU	is the last path used mask.
PI	is the path installed mask.
MBI	is the measurement block index.
PO	is the path operational mask.
PA	is the path available mask.
CHPID	are the channel path identifiers.

For the second line of the response, provided when an interrupt is pending on the subchannel:

KEY	is the SCSW protect key.
ECC	are the logout condition code bits.
FPI	are the format, prefetch, and initial response bits.
ZN	the initial status condition.
SCTL	are the function and activity control bytes.
CCW-ADDR	is the SCSW CCW address.
DS	are the SCSW device status bits.
SS	are the subchannel status bits.
CNT	is the SCSW residual CCW data count.
EXT-LOG	is the extended status/time word.

Response 2

The following response appears in your display screen.

```
COMMAND COMPLETE
```

indicates that the command has been successfully executed.

DUMP VECTOR

Privilege Class: G

Dump vector registers dumps the contents of vector registers, vector register pair or pairs, vector status register, vector mask register, vector activity count, and vector section size to the virtual printer.

Dump	$\left\{ \begin{array}{l} \text{VR} \\ \text{VP} \end{array} \right\} \left[\text{reg1} \left[\left\{ \begin{array}{l} - \\ : \end{array} \right\} \left[\begin{array}{l} \text{reg2} \\ \text{END} \end{array} \right] \right] \right] \left[\text{,elt1} \left[\left\{ \begin{array}{l} - \\ : \end{array} \right\} \left[\begin{array}{l} \text{elt2} \\ \text{END} \end{array} \right] \right] \right] \left[\text{*dumpid} \right]$ $\left[\left\{ \begin{array}{l} . \\ . \end{array} \right\} \left[\begin{array}{l} \text{regcount} \\ \text{END} \end{array} \right] \right] \left[\left\{ \begin{array}{l} . \\ . \end{array} \right\} \left[\begin{array}{l} \text{eltcount} \\ \text{END} \end{array} \right] \right]$
	VSR [*dumpid]
	VMR [*dumpid]
	VAC [*dumpid]
	VSS [*dumpid]

where:

VR

dumps one or more vector registers in hexadecimal.

VP

dumps one or more even-odd vector register pairs as 64-bit elements in hexadecimal, along with the decimal equivalent in exponential format.

reg1

If you wish to dump vector registers (VR), reg1 is a decimal number from 0 to 15 or a hex number from X'0' to X'F' representing the first, or only, vector register whose contents are to be dumped. If VR is specified without a register number, it will default to register 0.

If you wish to dump a vector register pair (VP), reg1 is an even decimal number from 0 to 14 or an even hexadecimal number from X'0' to X'E' of the first or only vector register pair. If reg1 is not specified with VP, it will default to vector register pair 0.

$\left\{ \begin{array}{l} - \\ : \end{array} \right\} \text{reg2}$

If you wish to dump a range of vector registers, reg2 is a decimal number from 0 to 15 or a hex number from X'0' to X'F' representing the last vector register whose contents are to be dumped. If you specify VP, reg2 must designate an even-numbered register. The dash (-) or colon (:) following reg1 specifies a range of register values for dumping. Specify the range of registers in ascending order. Reg2 must be equal to or greater than reg1. The contents of registers reg1 to reg2 will then be dumped.

$\left\{ \begin{array}{l} - \\ : \\ . \end{array} \right\} \text{[END]}$

When used in the register specification, this calls for a dump of all register contents from the starting registers specified by reg1 to register 15, or register pair 14-15.

{.}regcount

is a decimal number from 1 to 16 or a hex number from 1 to F specifying the number of vector registers or pairs whose contents are to be dumped. If VP is specified, the valid range is 1 to 8. If you dump vector registers, the sum of reg1 and regcount must not be larger than 16.

,elt1

The comma indicates that a register element specification follows. If the comma and element specification is not given, CP dumps the contents of elements 0-3 for VR, or element 0 only for VP. Etl1 is a hexadecimal number from 0 to one less than the section size representing the first, or only, element to be dumped.

**{-}elt2
{:}**

is a hexadecimal number from 0 to one less than the section size representing the last element whose contents are to be dumped. The dash (-) and colon (:) before indicates that you wish to dump a range of elements. The range of elements must be specified in ascending order. The operand elt2 must be equal to or greater than elt1. The contents of elements elt1 to elt2 are dumped.

{.}eltcount

is a hexadecimal number from 1 to the section size specifying the number of elements whose contents are to be dumped starting with elt1. The range indicated by elt1 and eltcount must not exceed the section size.

**{-}[END]
{:}**

When used in the element specification, this calls for a dump of the contents of all elements from the starting element specified by elt1 to the last element, which is one less than the section size.

VSR

dumps the current vector status register.

VMR

dumps the current vector mask register.

VAC

dumps the current vector activity count.

VSS

dumps the vector section size.

***dumpid**

is an asterisk followed by a 1- to 98-character identifier that is to be assigned to the dump. If specified, the dumpid must be the last operand you enter on the command line.

Usage Notes

1. No blanks are allowed between operands that relate to a single dump request.
2. If either "VR" or "VP" alone is specified, the default is to dump all elements of all registers.
3. For existing dump functions, a dumpid can optionally be specified.

DUMP VECTOR

Response 1

$VR_{nn,elt} = \text{xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx}$

The register elements are always dumped in groups of eight per line with the contents shown in hexadecimal. If required, multiple lines are dumped for each register.

where:

nn

indicates the register whose contents are to be dumped.

elt

indicates the first element number whose contents are dumped.

Response 2

$VP_{nn,elt} = \text{xxxxxxxxxxxxxxxxxxxx s.ddddddddddddddd Esdd ...}$

The contents of the register pair is dumped in both the hexadecimal format (the first part of the response) and in exponential decimal format (the second part of the response). Two elements of a register pair are dumped. The contents of two vector register pairs are dumped per line. If required, multiple lines are dumped for each register.

where:

nn

indicates the first register pair whose contents are dumped.

elt

indicates the first element number whose contents are to be dumped.

s

is the positive or negative indicator.

Response 3

$VSR = \text{xxxxxxxx xxxxxxxx}$

The vector status register contents are dumped in hexadecimal.

Response 4

$VMR = \text{xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx}$

The vector mask register contents are dumped in hexadecimal.

Response 5

$VAC = \text{xxxxxxxx xxxxxxxx}$

The vector activity count contents are dumped in hexadecimal.

Response 6

$VECTOR\ SECTION\ SIZE: nnn$

The vector section size is dumped in hexadecimal.

Response 7

VX_n = ZERO (IN-USE BIT IS OFF)

This response is given when the “in-use” bit of the referenced vector pair in the vector status register is off for a dump vector register request. When the in-use bit is off, the register contents are defined at zero.

X

is the indicator for “R” (register) or “P” (pair).

n

is the hexadecimal register number.

Migration Notes

VM/SP HPO: VM/XA SP evaluates the command for proper syntax prior to checking whether the Vector Facility is available.

ECHO

Privilege Class: G

Use the ECHO command to place your display device in the echo environment. In the echo environment, any input line you enter will be transmitted back to your display a specified number of times.

ECho	[nn] [1]
------	-----------------

where:

nn

is the number of times that the line is repeated. The default is 1. An invalid entry (that is, one that is less than 1 or greater than 99 or contains nonnumeric characters) is treated as 1.

Usage Notes

1. After the line has been repeated the specified number of times, CP issues another read to the display so you can enter another data line.
2. You can terminate the display before it is completed by pressing the PA1 key when the screen is in MORE or HOLDING status. Pressing PA1 at any other time will not terminate the display.
3. No line editing is done; the output line is the same as the input line and may contain any of the logical line editing symbols.

Response 1

ECHO ENTERED; TO TERMINATE TEST, TYPE END

is displayed after you have issued the ECHO command to indicate that your virtual machine has entered the echo environment.

Response 2

ENTER LINE

is displayed to prompt you to enter an input line. If you enter the reply END, you will return to the CP command environment.

ENABLE

Privilege Class: A, B, F

Use the ENABLE command to enable disabled or nonenabled display devices and SNA/CCS terminals. This allows these devices to access the VM/XA SP system.

ENable	{ rdev1 [rdev2 ...] rdev1-rdev2 SNA [userid] ALL }
--------	---

rdev1 [rdev2 ...]

is the device number or list of device numbers of display stations that you want to enable. If you specify more than one device number, separate them by one or more blanks.

rdev1-rdev2

is a range of real device numbers of display devices to be enabled.

SNA userid

enables communication with the SNA/CCS terminals owned by the VTAM service machine whose userid is specified. If no userid is specified, communication is enabled with all SNA/CCS terminals for all VTAM service machines.

ALL

enables all previously disabled or nonenabled locally attached 3270 display devices, including 328X printers and all SNA/CCS terminals.

Note: The logical 328X printers, however, are not enabled.

Usage Notes

1. On a single command line you may combine device numbers, lists of device numbers, and ranges of device numbers.
2. Do not attempt to enable display devices that are not supported. Use nonsupported console devices as dedicated devices attached to virtual machines.
3. If a display station has already been enabled, it is not affected by the ENABLE command.

Responses

COMMAND COMPLETE

indicates that the command has been completed. No error indication is provided if no displays are available to be enabled.

Migration Notes

VM/SP HPO: If VM/XA SP encounters an invalid device number in a list, it issues an error message and continues processing the rest of the devices.

EXTERNAL**Privilege Class: G**

Use the EXTERNAL command to make a simulated external interruption pending to your virtual machine. The virtual machine has control immediately.

EXternal	<p><u>KEY</u></p> <p>INTerval</p> <p>{ MALFunction EMERGENCY } cpuaddr</p> <p>{ CALL }</p> <p>[code [CR0 bitno] [CMD command]]</p>
----------	--

where:

KEY

is the operator external interruption key.

INTERVAL

is the interval timer at location 80. You can specify INTERVAL only if your virtual machine is in System/370 mode. When you specify INTERVAL, an interval timer interrupt is made pending.

MALFUNCTION

is a malfunction alert signal from the specified virtual processor address.

EMERGENCY

is an emergency signal from the specified virtual processor address.

CALL

is an external call SIGP signal from the specified virtual processor address; and is accepted only when no external call is already pending.

cpuaddr

is a virtual processor address from 0 to 3F (hexadecimal).

code

is an external interruption code.

CR0

indicates that a mask bit in CR0 is to be applied to this interruption.

bitno

is the bit number in CR0 that masks this interruption. It is a decimal number from 16 to 31.

CMD

indicates that a CP command is to be issued when the interruption is reflected. You must enter the command at the end of the command line; the remainder of the line will be interpreted as the CP command.

command

is the console command function to be issued when the interruption is reflected.

Usage Notes

1. The CP BEGIN command is not required after you issue the EXTERNAL command.
2. Since there is no virtual processor address argument for code:
 - a. 'EXTERNAL 1200' cannot be an alternative syntax for 'EXTERNAL MALFUNCTION cpua'
 - b. 'EXTERNAL 1201' cannot be an alternative syntax for 'EXTERNAL EMERGENCY cpua'
 - c. 'EXTERNAL 1202' cannot be an alternative syntax for 'EXTERNAL CALL cpua.' This is also true because of the architectural requirement that only one external call can be pending for a virtual processor at any one time.

A bit in CR0 can be specified for masking of this interruption. Since the interruption code does not imply a bit in CR0:

- a. 'EXTERNAL 40' is not an alternative syntax for 'EXTERNAL KEY.'
- b. 'EXTERNAL 80' is not an alternative syntax for 'EXTERNAL INTERVAL.' This is also true because of the architectural requirement that only one interval timer interruption can be pending for a virtual processor at any one time.

You can specify a CP command that you want to be executed once the interruption is reflected to your virtual machine. This ability is intended for use in system testing.

Responses

None.

Migration Notes

VM/SP HPO: As another option, VM/XA SP allows you to specify the type of external interrupt.

FLUSH

Privilege Class: D

Use the FLUSH command either to halt and immediately purge or, optionally, to hold the current output file on a specified real unit record device. If the device is not in intervention required status, has not been drained, or has not been taken offline, it will resume activity with the next available spool file.

Flush	rdev [ALL] [H01d]
-------	--------------------------

where:

rdev

is the real device number of the real printer or punch whose current file is to be terminated.

ALL

deletes all copies of the current output file. If more than one copy of the current output file is being printed or punched and you specify ALL, all copies are deleted; if you do not specify ALL, only the current copy will be deleted, and the next copy, if any, will be printed or punched.

HOLD

places the spool file in system HOLD status; the file will not be purged. You can reset the file status using the CP CHANGE command.

Usage Notes

1. You will normally use the FLUSH command when a mistake in output queuing has occurred, or when a high-priority job must be started before the current job has finished. For example, you would issue FLUSH if a file was directed to a printer equipped with the wrong printer train or if a file was misdirected to the real punch rather than to a user's virtual reader.
2. Special handling is required if a spool file is started on a real output device that cannot be made ready. If such a condition arises, the operator will receive an error message, and can then recover the file and restart it. To do so, he must enter the following command:
 FLUSH rdev HOLD
3. If the operator has specified HOLD, the file will be placed in system HOLD status; otherwise, if multiple copies of the file were to be processed, the file will be requeued and started on another device, if one is available.

Responses

{PRT} rdev { FILE FLUSHED } userid FILE spoolid RECS { nnnn } COPY [*]nnn SEQ sss
 {PUN} { FILE HELD }
 { COPY FLUSHED } { nnnK }
 { nnnM }

where:

PRT

PUN

is the type of device that you have flushed.

rdev

is the real device number of the printer or punch to be affected.

FILE FLUSHED

FILE HELD

COPY FLUSHED

indicates the disposition of the file.

userid

is the identification of the user who was to receive the output.

FILE spoolid

is the system-assigned spoolid number by which VM/XA SP refers to the file.

RECS $\left\{ \begin{array}{l} \text{nnnn} \\ \text{nnnK} \\ \text{nnnM} \end{array} \right\}$

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

specifies the number of file copies you want printed or punched when the file is produced.

When the optional asterisk (*) is specified, the 3800 printer will print each page from the spooled output files nnn times before going on to the next page.

If a file is not spooled to the 3800 printer, the function indicated by the asterisk (*) is ignored and duplication occurs normally.

SEQ sss

is the sequence number for the file on the printer or punch.

Migration Notes

VM/SP HPO

1. VM/XA SP provides information about the FILE being flushed, not information about the PRINTER being flushed, as in VM/SP HPO.
2. VM/XA SP does not support logical printers.

FORCE

Privilege Class: A

Use the FORCE command to force any user on the system to logoff.

FORCE	userid
-------	--------

where:

userid

is the user identification of the user who is to be logged off.

Usage Notes

The FORCE command cannot log off a hung user or a user who may have outstanding I/O, which prevents completion of either the LOGOFF or FORCE commands by CP. The system operator can determine which I/O device is involved and make the device ready so that the user can continue operations or be forced from the system. The QUERY command can help determine the status of all or selected devices.

Response 1

```
LOGOFF AT hh:mm:ss zone weekday mm/dd/yy {BY SYSTEM}
                                     {userid }
```

accompanies the normal accounting message produced at logoff.

where:

hh:mm:ss

is the current time in hours:minutes:seconds.

zone

is the time zone.

weekday

is the current day of the week.

mm/dd/yy

is the current date in month/day-of-month/year format.

BY {SYSTEM} {userid }

indicates who forced the logoff of your virtual machine. If you were logged off by another user, and not the system, that user's userid appears in the logoff message.

Response 2

```
{ type {rdev} } LOGOFF AS userid USERS = nnnnn FORCED
 { luname }
```

is the message received by the primary system operator.

where:

type

is the display device type.

rdev

ldev

is the real or logical device number.

luname

identifies a SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

AS userid

is the identification of the user who was forced off.

USERS = nnnnn

is the number of users remaining on the system.

Response 3

If a user has issued the CP DISCONNECT command, and the system operator issues the CP FORCE command to remove that user from the VM/XA SP system, the primary system operator receives the following response:

```
USER DSC LOGOFF AS userid USERS = nnnnn FORCED
```

where:

USER DSC

indicates that the user's virtual machine was disconnected when the FORCE command was issued.

LOGOFF AS userid

is the identification of the user who was forced off.

USERS = nnnnn

is the number of users who remain on the VM/XA SP system.

Migration Notes

VM/SP HPO: In VM/XA SP, users being forced receive a response indicating whether the system or another userid is forcing them.

VM/XA SF: In VM/XA SP, the response uses GRAF instead of DISP as in VM/XA SF.

FORWARD
Privilege Class: D

Use the FORWARD command to reposition current spool output on a real printer or punch.

FORward	rdev	[nnn] [1]
---------	------	------------------

where:

rdev

is the real device number of the real printer or punch that you want forward spaced.

nnn

specifies the number of pages or cards to be forward spaced on the printer or punch. The limit is 999. If you do not specify a value, the default is 1.

Usage Notes

1. Pages are counted in FORWARD operations by looking for skip to channel 1 specifications (that is, page ejects). If a file contains no skip to channel 1 specifications, 66 output lines (including blank lines) will be treated as a "page." Counting begins at the end of the current buffer.

Responses

```
{PRT} rdev FORWARD userid FILE spoolid RECS { nnn } COPY [*]nnn SEQ sss
{PUN}                                     { nnnK }
                                           { nnnM }
```

where:

PRT**PUN**

indicates the type of device that has been forward spaced.

rdev

is the real device number.

userid

is the identification of the user whose file was active and forward spaced.

FILE spoolid

is the system-assigned spoolid number by which VM/XA SP refers to the file.

```
RECS { nnn }
      { nnnK }
      { nnnM }
```

is the number of logical records left in the file. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

is the remaining number of copies of the file to be produced, where 001 is the last copy. The asterisk (*) is for the 3800 printer and indicates that the printer will copy each page nnn times before going on to the next page.

SEQ sss

is the sequence number for the file on the printer or punch.

FREE

Use the FREE command described in this section to:

- Remove a set of spool files belonging to a specified user from system hold status
- Remove a specified user from LOGON HOLD status so the user can access the system.

Complete Format for FREE

The format box below shows all of the operands available with the FREE command. Complete descriptions of the operands appear on the following pages, in the same sequence as the operands appear here.

		Class
Free	userid LOGon	B
	userid [Printer PUnch <u>ALL</u>]	D

Usage Notes

None.

FREE LOGON

Privilege Class: B

Use FREE LOGON to remove a user from LOGON HOLD status so the user may access the system.

FRee	userid LOGon
------	--------------

where:

userid

is the user identification of the user you want to remove from LOGON HOLD status.

LOGON

allows the specified user to access the system.

Usage Notes

Use FREE LOGON in conjunction with the CP HOLD LOGON command.

Responses

None.

FREE PRINTER/PUNCH

Privilege Class: D

Use FREE PRINTER, PUNCH to remove a user's existing spool files from system HOLD status. Until you issue the HOLD command, any subsequent files that the user creates are not put in system HOLD status.

Free	userid	Printer PUnch ALL
------	--------	-------------------------

where:

userid

is the user whose spool files are to be released from a system hold status. The user need not be logged on when you issue the command.

PRINTER

PRT

frees the printer files for the specified user.

PUNCH

PCH

frees the punch files for the specified user.

ALL

frees all printer and punch files for the specified user.

Usage Notes

1. A spool file can be in either of two types of HOLD status, or in both types:

- User HOLD
- System HOLD.

User HOLD is a facility for the file owner to manipulate and control through the SPOOL, CLOSE, and CHANGE commands. System HOLD is a separate function. Generally, it pertains to an error situation and is wholly managed by the class D operator through the CHANGE SYS/NOSYS and HOLD/FREE commands.

If a spool file is in user HOLD status, the file can be freed for processing by (a) the user who created the file or (b) a user with class D CP command privileges.

If a spool file is in system HOLD status, only an authorized user can free the file for processing by using the FREE or CHANGE NOSYS commands. The authorized user of the FREE command can free previously held files of any user from system HOLD status.

If a spool file is in both user HOLD status and system HOLD status, both HOLDs must be removed before the file can be processed.

2. If the file is in HOLD status, it will not be selected for processing on a real printer or punch. Use HOLD to temporarily detain a file or a group of files so you can correct or modify a spooling condition.

Responses

None.

HALT

Privilege Class: A

Use the HALT command to terminate execution of an active channel program on a specified real device.

HALT	rdev
------	------

where:

rdev

is the device number of the real device that you want to stop.

Usage Notes

1. **Warning:** Use this command only in extreme cases and only after careful consideration. Indiscriminate use of this command may cause unwanted results.
2. After issuing the HALT command, you should issue the QUERY (real devices) command to determine the status of the device. If the device is in an intervention required state, you should make the device ready.

Responses

HALT INITIATED TO type rdev

indicates that VM/XA SP has accepted the command and initiated action to halt the device.

HALT NOT INITIATED BECAUSE type rdev IS NOT ACTIVE

indicates that a HALT was issued to an idle device and that VM/XA SP has not accepted the command.

where:

type

is one of the following:

Type	Meaning
DASD	Direct access storage device
TAPE	Magnetic tape
RDR	Card reader
PRT	Printer
PUN	Card punch
GRAF	Display device
CONS	Console
CTCA	Channel-to-channel adapter
CTLR	3704 or 3705 Communications Controller
MSC	Mass Storage Controller
DEV	Any other device.

rdev

is the real device number of the device.

Migration Notes

VM/SP HPO: In VM/XA SP the response indicates whether the system initiated the halt.

HOLD

HOLD

Use the HOLD command, described here and on the following pages, to:

- Place user spool files into system HOLD status
- Prevent specific users from logging on to the system.

Complete Format for HOLD

The format box below shows all of the operands available with the HOLD command. Complete descriptions of the operands appear on the following pages, in the same sequence as the operands appear here:

		Class
H01d	userid LOGon	B
	userid [Printer PUnch ALL]	D

General Usage Notes for HOLD

The HOLD command works for a maximum of 32 userids at any one time.

HOLD LOGON

Privilege Class: B

Use HOLD LOGON to prevent a specified user from accessing the system. The format of HOLD LOGON is:

H0ld	userid LOGon
------	--------------

where:

userid

is the user identification of the user you want to prevent from accessing the system. The user does not have to be logged on when you issue the command.

LOGON

prevents the specified user from logging on to the system.

Usage Notes

1. Use the CP FREE command to release a user from LOGON HOLD status.
2. The HOLD LOGON command is not allowed for the system operator userid.

Responses

None.

HOLD PRINTER/PUNCH

Privilege Class: D

Use HOLD PRINTER/PUNCH to place a user's existing spool files into system HOLD status. Until you issue the FREE command, any subsequent files that the user creates will also be put into HOLD status. The format of HOLD PRINTER/PUNCH is:

HOLD	userid	Printer PUnch ALL
------	--------	-------------------------

where:

userid

is the identification of the user whose spool files are to be placed in system HOLD status. The user need not be logged on when you issue the command.

PRINTER

PRT

holds the printer files for the specified user.

PUNCH

PCH

holds the punch files for the specified user.

ALL

holds all the printer and punch files for the specified user.

Usage Notes

1. You can place a spool file in either of two types of HOLD status, or in both types:

- User HOLD status
- System HOLD status.

A spool file is in user HOLD status if the user who created the file assigned it HOLD status using the CP SPOOL, CLOSE, or CHANGE commands. A file is in system hold status if the class D operator has placed it there through the CHANGE SYS command/option or through the HOLD command. Also, the system can place a file in system HOLD if an error occurs.

If a file is in system HOLD status, only a class D user can free the file for processing. The user of the HOLD command has the ability to place in system HOLD status the files of any user on the system.

If a spool file is in both user HOLD status and system HOLD status, both HOLDs must be removed before the file can be processed.

2. HOLD userid status is maintained whether the user is logged on or across a warm start.
3. When a user is put in HOLD status, all current files are changed to system HOLD, as well as any additional files created while the user remains in HOLD.
4. If the file is in HOLD status, it will not be selected for processing by a real printer or punch. Use HOLD to temporarily detain a file or a group of files so you can correct or modify a spooling condition.

5. A maximum of 32 users can be held at any one time.
6. Use the CP FREE command to release a user's files from system hold status.

Responses

None.

INDICATE

Use the INDICATE command described in this section to display at your console the use of and contention for system resources.

If you are a class G user, use the INDICATE command to:

1. Display recent contention for system resources. This can be helpful to predict system throughput and response time characteristics which your virtual machine may experience now and in the near future.
2. Display environment characteristics of your virtual machine. This includes machine type (370 or XA), the origin of the system IPLed (loaded) in your virtual machine, and the presence or quantity of system resources available to your virtual machine.
3. Display measurements of resources used by your virtual machine. Such measurements are accumulators; that is, always increasing after virtual machine logon.

If you are a class E (and, in some cases, class B and C) user, the INDICATE command provides all the class G functions and additionally:

1. Detailed information on use of and contention for system resources. Userids of virtual machines currently using certain resources can be displayed.
2. The status of currently active virtual machines as determined by the system scheduler and dispatcher.
3. Environment characteristics of, and measurements of resources used by, any virtual machine logged on.

Complete Format for INDICATE

The format box below shows all of the operands available with the INDICATE command. Complete descriptions of the operands appear on the following pages, in the same sequence as the operands appear here.

		Class
INDicate	[I/O]	B/E
	[<u>LOAD</u>]	E/G
	[PAGING [<u>WAIT</u>]] [ALL]	E
	[Queues]	E
	[USER [*]] [<u>userid</u>] [SYSTEM]	E/G
	[VECtor]	C/E

General Usage Notes for INDICATE

The INDICATE command:

1. Provides the system operator with a “snapshot” of system activities
2. Provides a means of determining the execution characteristics of a program with respect to the resources it uses.

Migration Notes

VM/SP HPO

1. In VM/XA SP, the responses are different than in VM/SP HPO because of differences in software architecture.
2. The VM/XA SP response has five digits for the number of users.

VM/XA SF: In VM/XA SP the responses are different than in VM/XA SF because of changes in the scheduler.

INDICATE I/O**Privilege Class:** B, E

Use **INDICATE I/O** to list all virtual machines currently in an I/O wait state and to obtain for each listed virtual machine the device number of the real device to which the most recent virtual I/O operation was mapped.

INDicate	I/O
----------	-----

where:

I/O

displays a list of virtual machines that are in I/O wait.

Usage Note:

Because the response indicates only an instantaneous sample, issue the **INDICATE I/O** command several times before you assume a condition to be persistent.

Response 1

```
userid1 xxxx userid2 yyyy ... (up to five userids per line)
```

where:

userid1**userid2**

is the user identification of the virtual machine.

xxxx**yyyy**

is the real device number of the device to which the most recent I/O operation was mapped, or '----' (4 dashes) when the I/O is to a virtual device rather than to a real device.

Note: In the case where a virtual machine may have issued multiple I/O operations, the response indicates the real device number corresponding to the most recent operation issued.

Response 2

```
NO USERS IN I/O WAIT
```

is issued when appropriate.

INDICATE LOAD

Privilege Class: E, G

Use INDICATE LOAD to provide an indication of the operating load on the VM/XA SP system, by displaying:

- The percentage utilization for each processor in your system
- The usage of real storage
- Information concerning Expanded Storage
- The paging rate
- The number of users in the dispatch, eligible, and dormant lists
- The percentage of time that the system is executing instructions on the Vector Facility
- The number of recent Vector Facility users
- The number of real Vector Facilities that can be shared by users.

Note: LOAD is the default operand of the INDICATE command.

INDicate	[<u>LOAD</u>]
----------	-----------------

where:

LOAD

displays an indication of the operating load on the VM/XA SP system.

Usage Notes

1. Vector Facility information will only be a part of the response if vectors are available. VEC-nnn% will appear for each processor if there is at least one processor with a vector. VECTOR-(#users) will appear if there is at least one processor with a vector.

The Expanded Storage information will be displayed only if this storage is available.

2. If the wait-state interpretation capability is available on the real processor, it is used for dedicated virtual processors unless the guest is in System/370 mode with SET TIMER ON. When the wait-state interpretation capability is used, the guest virtual machine remains dispatched even when it enters a PSW enabled-wait state. Time spent dispatched in the wait state is included in the processor utilization statistics reported for the guest.

Response 1

Class G response will appear as:

```
AVGPROC-nnn% pp [AVGVEC-nnn% vv]
[XSTORE-nnnnnn/SEC MIGRATE-cccc/SEC]
[XSTORE MDC READS-nnnnnn/SEC HIT-RATIO-hhh% WRITES-mmmmmm/SEC]
STORAGE-nnn% PAGING-nnnn/SEC STEAL-nnn%
Q0-nnnnn Q1-nnnnn          Q2-nnnnn EXPAN-nnn Q3-nnnnn EXPAN-nnn
```

INDICATE LOAD

Response 2

Class E response will appear as:

```
AVGPROC-nnn% pp [AVGVEC-nnn% vv VECTOR-aaaaa]
[XSTORE-nnnnnn/SEC MIGRATE-cccc/SEC]
[XSTORE MDC READS-nnnnnn/SEC HIT-RATIO-hhh% WRITES-mmmmmm/SEC]
STORAGE-nnn% PAGING-nnnn/SEC STEAL-nnn%
Q0-nnnnn(zzzzz) DORMANT-nnnnn
Q1-nnnnn(zzzzz) E1-nnnnn(zzzzz)
Q2-nnnnn(zzzzz) EXPAN-nnn E2-nnnnn(zzzzz)
Q3-nnnnn(zzzzz) EXPAN-nnn E3-nnnnn(zzzzz)

PROC adr1-nnn% [VEC-nnn%] PROC adr2-nnn% [VEC-nnn%]
PROC adrm-nnn% [VEC-nnn%] PROC adrn-nnn% [VEC-nnn%]
```

where:

AVGPROC-nnn%

is the average processor utilization percentage for all processors combined, not including dedicated processors. nnn ranges from 0 to 100.

PP

is the number of processors but does not include dedicated processors.

AVGVEC-nnn%

is the average vector utilization percentage for all Vector Facilities, not including dedicated vectors. nnn ranges from 0 to 100.

vv

is the number of Vector Facilities available, not including dedicated vectors.

VEC-aaaaa

is the number of recent Vector Facility users. Recent users are those virtual CPUs (non-dedicated) that are in the dispatch list or in the eligible list and that, during their last time slice, have used a Vector Facility. This number only appears when Vector Facilities are installed in the configuration.

STORAGE-nnn%

indicates the usage of real storage. It is an approximate percentage. It shows the relationship of the number of pages needed for the working sets of virtual machines in the dispatch list to the number of pageable pages in the system. It is possible for STORAGE to exceed 100%.

PAGING-nnnn/SEC

is a value that indicates the current system paging activity in decimal and does not include spooling information.

STEAL-nnn%

indicates the percentage of pages stolen that were RESERVED pages or that were referenced pages owned by users who were in the dispatch list. Page read requests include all pages read in from DASD and XSTORE.

XSTORE-nnnnnn/SEC

indicates the current system expanded storage activity, for both paging to real storage from XSTORE and paging to XSTORE from real storage.

MIGRATE = cccc/SEC

indicates the number of pages per second migrating from Expanded Storage to DASD.

READS-nnnnnn/SEC

is the number of data pages moved from the minidisk cache to main storage.

WRITES-mmmmmm/SEC

is the number of data pages moved from main storage to the minidisk cache (all the updates and inserts in the cache).

HIT-RATIO-hhh%

is the percentage of successful lookups in the minidisk cache. This is the percentage of eligible read I/Os that were avoided because of minidisk cache.

Qn

indicates a smoothed average of the number of Qn users in the dispatch list. A Qn user is an En user who is now in the dispatch list.

Q0 indicates the users that were added to the dispatch list with no delay in the eligible list, regardless of the length of their current transaction.

Q1 indicates the users that have just begun a transaction, and therefore are assumed to be currently processing short transactions.

Q2 indicates the users which did not complete their current transactions during their first dispatch list stay and therefore are assumed to be running medium-length transactions.

Q3 indicates the users that did not complete during their second dispatch stay and therefore are assumed to be running long transactions.

EXPAN-nnn

indicates the total delay in response time that a virtual machine experiences due to contention for those resources scheduled by the VM/XA scheduler. Q2 gives the expansion factor for those completing in Q2 (medium-length), and Q3 for those completing in Q3 (long-running transactions).

En

indicates a smoothed average of the number of En users in the eligible list.

Although a user may briefly be considered an E0 user, no E0 counts appear. This is because, by definition, such a user spends virtually no time in the eligible list. After smoothing, this count would always be 0 if shown. If, by unusual timing circumstances, a user is found in an E0 category, the user is counted with the Q0 user counts.

E1 indicates the users that have just begun a transaction, and therefore are assumed to be currently processing short transactions.

E2 indicates the users that did not complete their current transactions during their first dispatch list stay and therefore are assumed to be running medium-length transactions.

E3 indicates the users that did not complete during their second dispatch list stay and therefore are assumed to be running long transactions.

ZZZZ

is the number of users within the Qn or En counts that are designated as "loading users." A loading user is a user assumed to be loading a working set into memory. A user is labeled "loading" or "not loading" based primarily on the number of page reads done during the user's last dispatch time slice. A high count of page reads is assumed to designate a loading user. A user also may be labeled "loading" if the system has moved any of the user's referenced pages out to DASD while the user was not in the dispatch list. The "loading" designation

INDICATE LOAD

is given in anticipation of the user's need to read pages back into storage when next dispatched.

DORMANT-nnnnn

is a smoothed average of the number of dormant list users.

PROC adrn-nnn%

represents the percentage of time that the system is running (for example, not waiting) on the processor indicated. adrn represents the processor ID in hexadecimal. nnn ranges from 0 to 100.

VEC-nnn%

represents the percentage of time that the system is executing instructions on the Vector Facility associated with the PROC at address adrn. nnn ranges from 0 to 100.

INDICATE PAGING

Privilege Class: E

Use INDICATE PAGING to display a list of virtual machines that are in page wait, or to display page residency data for all users of the system, including the system nucleus and pageable routines.

INDicate	PAGing	<u>WAIT</u> ALL
----------	--------	--------------------

where:

PAGING

displays information about the users in page wait, or page residency data for all users of the system.

WAIT

displays the user identifications of the virtual machines that are currently in page wait, and the number of page frames allocated in preferred and nonpreferred auxiliary storage. This is the default.

Use PAGING WAIT when the INDICATE QUEUES command has shown that a significant proportion of the active users of the system are persistently in page wait.

ALL

displays the page residency data of all users of the system.

Usage Notes

1. The INDICATE PAGING command is useful in all environments, but particularly in those with Expanded Storage available for CPU use and for those with 2305 DASDS as “preferred” auxiliary storage. If XSTORE is available for CPU use, it is the number of pages this user has in XSTORE blocks. Otherwise, it is the number of pages on preferred DASD devices.
2. A colon (:) separates preferred DASD from nonpreferred DASD paging activity in the response.
3. A single response line may show paging activity for up to three userids. Both preferred DASD/non-preferred DASD, and Expanded Storage/DASD responses may appear in the display.

Response 1

```
userid1 xxxxxx:yyyyy userid2 xxxxxx:yyyyy userid3 xxxxxx:yyyyy
userid4 ...(up to three userids per line)
```

where:

```
userid1
userid2
userid3
userid4
```

is the user identification of a virtual machine.

INDICATE PAGING

xxxxxx

is the decimal number of pages allocated on preferred auxiliary storage for this user. If Expanded Storage is available for CP use, it is the number of pages this user has in XSTORE blocks. Otherwise, it is the number of pages on preferred DASD devices.

yyyyyy

is the decimal number of pages allocated on non-preferred auxiliary storage for this user. If Expanded Storage is available for CP use, it is the number of pages this user has on DASD devices. Otherwise, it is the number of pages on non-preferred DASD devices.

Note: You can receive responses for one or more virtual machines. For example, suppose you issue the INDICATE PAGING WAIT command, and you receive the following response:

```
userid01 000021:000110 userid02 000023:000127
```

This response indicates that two users are in page wait. If the two users shown in the response were to execute programs of similar characteristics, then userid01 would be expected to experience more page wait than userid02.

Response 2

```
NO USERS IN PAGE WAIT
```

is issued when appropriate.

INDICATE QUEUES

Privilege Class: E

Use INDICATE QUEUES to display, in priority order, the current members of the dispatch and eligible lists. Each member is a virtual CPU of an active user. Status critical to the system scheduler and dispatcher is also displayed for each virtual machine.

INDicate	Queues
----------	--------

where:

QUEUES

displays information about active users in the system.

Usage Notes

When INDICATE QUEUES consistently shows En users, an eligible list is forming because one or more system resources appears to be constrained. Under such conditions, forcing active users to wait in the eligible list increases virtual machine throughput as a whole, and also reduces system overhead needed to maintain a large, dynamic dispatch list. The appearance of a constrained resource is affected by:

1. The amount of resources available on the system.
2. The total demand for the resources.
3. The fraction of resource which is available to certain user classes. For example, a preferred virtual machine does not contend for storage, but all other virtual machines must vie for the leftover storage. Also, the SET SRM and SET RESERVED commands can affect the amount of resources available to certain user classes.

The system resources tracked are: storage, paging capacity, and CPU.

Responses

```
userid ttnn aa bbb ssssss/tttttt userid2 ...
(up to 2 userids per line)
```

where:

userid1

userid2

is the user identification of a virtual machine.

tt

is a code for the VMDTYPE, if other than the user. It can be:

- MP for virtual MP. Each virtual CPU defined for a user will show up separately with the same user id and a type of MP. Each is a separately dispatchable unit.
- Blank if the VMDBK is any other type.

nn

is the virtual CPU number of this VMDBK for virtual MP guests, or blank if not for virtual MP guests.

INDICATE QUEUES

aa

is the current list status of the user:

Q0 - In the dispatch list and exempt from eligible list delays.

Q1 - In the dispatch list and entered as an E1 user.

Q2 - In the dispatch list and entered as an E2 user.

Q3 - In the dispatch list and entered as an E3 user.

E1 - In the E1 eligible list.

E2 - In the E2 eligible list.

E3 - In the E3 eligible list.

Note: Users who have been given QUICKDSP will be Q0 whenever they are in the dispatch list. Other users may show up briefly as Q0 if they really belong in the eligible list, but must make brief appearances in the dispatch list (for example, to finish with and release a lock).

bbb

is one of the following status indicators:

Rnn

indicates that the user is the current RUNUSER on the specified real processor, where nn is the processor ID (in hexadecimal).

PG

indicates that the user is in page wait; the user's virtual machine is not executing because CP is trying to bring in a page from a paging device.

IO

indicates that the user is in I/O wait; that is, the user's virtual machine is not executing because access to the device is not available at the moment.

EX

indicates that the user is in instruction simulation wait; that is, the user's virtual machine is waiting for the completion of instruction simulation.

PS

indicates that the user is in PSW wait; that is, the user's virtual machine is in an enabled wait state.

Note: In cases where a virtual machine may be in more than one of the above states, only one state is displayed; the state displayed is the first one encountered in the order of priority indicated above.

sssss

is a decimal number indicating the number of pages resident in real storage.

ttttt

is a decimal number indicating the number of pages estimated by the dispatcher as the working set of this user.

Note: The response you receive will list users in the following order:

1. All dispatch list users ("Qn") in dispatch priority order.
2. All eligible list users ("En") in eligible list priority order.

These values are important in system performance analysis.

INDICATE USER

Privilege Class: E, G

Use INDICATE USER to find out about the resources used and occupied by a virtual machine.

Class G users can issue the command to find out about the resources used and occupied by their own virtual machines, while the class E user can issue the command to find out about the resources used and occupied by any virtual machine in the system.

INDicate	USER	[* _ userid SYSTEM
----------	------	---------------------------------

where:

USER

indicates that you are requesting information about the resources used and occupied by a virtual machine, the system, or a names saved system (NSS).

*

userid

SYSTEM

specifies the virtual machine for which you are requesting information. If you are requesting information about your own virtual machine, you can specify userid as an asterisk (*).

SYSTEM provides resource information about the system.

Only class E users may specify a userid other than their own.

Usage Notes

1. Only one virtual processor can have an IPLed device and/or a named saved system displayed at any given time. The rest will be NONE.
2. Use INDICATE USER to display the total amount of certain resources used by your virtual machine during the current session.
3. Use INDICATE USER before and after the execution of a program to determine the execution characteristics of that program in terms of the resources used.
4. Vector time is a subset of processor time; that is, using a second of vector processor time results in incrementing both the processor time and the Vector Facility time counts. This applies to both virtual (guest) and total time. The reason for this is that using the VF implies using the processor; all VF usage arises from executing instructions on that processor.
5. VVECTIME and TVECTIME only appear for those virtual machines with a Vector Facility defined.
6. If the wait-state interpretation capability is available on the real processor, it is used for dedicated virtual processors unless the guest is in System/370 mode with SET TIMER ON. When the wait-state interpretation capability is used, the guest virtual machine remains dispatched even when it enters a PSW enabled-wait state. Time spent dispatched in the wait state is included in the virtual and total processor time reported for the guest.

Response 1

```

USERID=userid MACH={370 } STOR={nnnnnnnnK } VIRT={R } XSTORE={nnnnnnnnM }
                {XA } {nnnnM } {F } {NONE }
                {V }

IPLSYS={DEV vdev } DEVNUM=nnnn
        {DEV NONE }
        {sysname }

PAGES: RES=nnnnnn WS=nnnnnn LOCK=nnnnnn RESVD=nnnnnn
NPREF=nnnnnn PREF=nnnnnn READS=nnnnnn WRITES=nnnnnn
CPU nn: CTIME=hh:mm VTIME=mmm:ss TTIME=mmm:ss IO=nnnnnn[+]
        RDR=nnnnnn PRT=nnnnnn PCH=nnnnnn
        [VVECTIME=mmm:ss TVECTIME=mmm:ss]
    
```

The response gives all the data from the user's VMDBK that is relevant to the user's virtual machine's paging activity, resource occupancy, processor usage, and accumulated I/O activity counts since logon. Time and count values are ever-increasing accumulators.

where:

userid

is the user identification of the virtual machine.

MACH = { 370 }
 { XA }

is user's virtual machine type: System/370, or 370-XA.

STOR = { nnnnnnnnK }
 { nnnnM }

is the virtual machine storage size. This value is displayed in megabytes (M) if the storage size is a multiple of megabytes; otherwise it is printed in kilobytes (K).

VIRT = (R)
 (F)
 (V)

indicates if the virtual machine is operating in the V=R, V=F or V=V area.

XSTORE = { nnnnnnnnM }
 { NONE }

If Expanded Storage is attached to this virtual machine, nnnnnnnnM is its size in megabytes. NONE indicates that Expanded Storage is not attached to this virtual machine.

If INDICATE USER is issued for SYSTEM, the XSTORE value displayed is the amount of Expanded Storage available for system use.

IPLSYS = { DEV vdev }
 { DEV NONE }
 { sysname }

vdev is the device number of the last IPLed device. If no device has been IPLed, NONE is displayed.

sysname is the name of the last named saved system to be IPLed. It was previously saved via the SAVESYS command.

DEVNUM = nnnn

indicates the number of devices attached to this user.

RES = nnnnnn

is the number of the user's virtual storage pages resident in real storage at the time the command is issued.

WS = nnnnnn

is the most recent system estimate of the user's working set size.

LOCK = nnnnnn

is the current number of the user's virtual storage pages that are locked.

RESVD = nnnnnn

is the current number of the user's virtual storage pages that are reserved.

NPREF = nnnnnn

is the current number of virtual pages allocated on system non-preferred paging space for this user.

PREF = nnnnnn

is the current number of virtual pages allocated on system preferred paging space for this user.

READS = nnnnnn

is the total number of page reads for this user since logon.

WRITES = nnnnnn

is the total number of page writes for this user since logon.

CPU nn

indicates which virtual processor the following information refers to.

CTIME = hh:mm

is total connect time for the user, in hours:minutes.

VTIME = mmm:ss

is total virtual time for the user, in minutes:seconds.

TTIME = mmm:ss

is total virtual and simulation time for the user, in minutes:seconds.

IO = nnnnnn[+]

is the total number of nonspooled I/O requests issued by the user since logon. "[+]" indicates that this guest has operated with the start interpretive-execution assist and the number of I/O requests issued may actually be larger than shown.

RDR = nnnnnn

is the total number of virtual cards read since the user logged on.

PRT = nnnnnn

is the total number of virtual lines printed since the user logged on.

PCH = nnnnnn

is the total number of virtual cards punched since the user logged on.

VVECTIME = mmm:ss

is the accumulated time in minutes and seconds that the specified virtual machine has used a Vector Facility since the last LOGON command.

TVECTIME = mmm:ss

is the accumulated time in minutes and seconds that the specified virtual machine has used a Vector Facility, plus the time CP has used the Vector Facility on behalf of this virtual machine since the last LOGON command.

Note: In all the above responses, the value nnnnnn is in decimal. VVECTIME and TVECTIME will not be shown if the virtual processor does not have a Vector Facility defined.

If your virtual machine is a virtual multiprocessor (MP) configuration, the first response is for the base processor. Responses for the additional virtual processors will follow the response for the base. In this case, the following fields are meaningful only for the virtual MP configuration as a whole:

MACH
STOR
VIRT
DEVNUM
RES
LOCK
RESVD
NPREF
PREF
READS
WRITES
XSTORE
WS.

The value of any one of the above fields in any response is identical to the value in the base processor response.

CTIME represents the total connect time only in the base processor response. For your other virtual processors, CTIME indicates the total elapsed time since creation of the virtual processor. If you use the CP DEFINE command to define the virtual processor after you logged on, this value differs from the value returned in the base processor response.

The following fields are tracked separately for each virtual processor in your virtual machine configuration:

VTIME
TTIME
IO
RDR
PRT
PCH
VVECTIME
TVECTIME

IPLSYS reflects the device number of the last device or the last named saved system IPLed on that virtual processor.

Response 2

Note: The response line beginning with XSTORE appears only if Expanded Storage is installed on the host machine configuration. This response line displays up to 999 999 Expanded Storage blocks (nnnnnn). When you exceed 999 999, the Expanded Storage block are counted in 1000 unit increments (nnnnnK).

```

.
.
.
XSTORE=nnnnnn READS=nnnnnn WRITES=nnnnnn MIGRATES=nnnnnn
          nnnnnK      nnnnnK      nnnnnK      nnnnnK
.
.
.

```

where:

$$\text{XSTORE} = \left\{ \begin{array}{l} \text{nnnnnn} \\ \text{nnnnnK} \end{array} \right\}$$

is the number of blocks allocated in Expanded Storage for this user.

$$\text{READS} = \left\{ \begin{array}{l} \text{nnnnnn} \\ \text{nnnnnK} \end{array} \right\}$$

is the number of reads from Expanded Storage to main storage for for this user.

$$\text{WRITES} = \left\{ \begin{array}{l} \text{nnnnnn} \\ \text{nnnnnK} \end{array} \right\}$$

is the number of writes to Expanded Storage from main storage for this user.

$$\text{MIGRATES} = \left\{ \begin{array}{l} \text{nnnnnn} \\ \text{nnnnnK} \end{array} \right\}$$

is the number of blocks transferred from Expanded Storage to DASD for this user.

INDICATE VECTOR**Privilege Class:** C, E

Use the INDICATE VECTOR command to display the names of active and loaded Vector Facility users. Active users are those virtual machines that are on the dispatch list or on the eligible list and that during their last time slice have used a Vector Facility. Loaded users are those whose vector status is loaded into a real Vector Facility at command execution time; these users are also listed, regardless of which list they are on.

INDicate	VEctor
----------	--------

Usage Notes

None.

Response 1

Two lists of userids are displayed in response to the INDICATE VECTOR command. The ACTIVE VECTOR USERS list shows the userid and virtual CPU address of each VMDBK that used the Vector Facility in its last time slice. The LOADED VECTOR USERS list shows the userid and virtual CPU address for each VMDBK that has its virtual Vector Facility currently loaded in a real Vector Facility; this list also shows the address of the real Vector Facility on which the VMDBK is loaded. An example of this response would be:

ACTIVE VECTOR USERS

```
MILLER  01, JONES  00, JONES  01, BROWN  00, BROWN  03
COMPTST 00, MVS2  01, MVS1  02
```

LOADED VECTOR USERS

```
JONES   00 ON CPU 01
JONES   01 ON CPU 02
SMITH   01 ON CPU 03
```

In this example, userid JONES is a virtual multiprocessor configuration with two virtual CPUs loaded into real Vector Facilities. Userid SMITH's virtual CPU 01 has its virtual Vector Facility loaded in the real Vector Facility attached to real processor 03.

Response 2

NO ACTIVE VECTOR USERS

is displayed when there are no recent Vector Facility users and when no user's virtual Vector Facility is loaded into any real Vector Facility.

Response 3

NO LOADED VECTOR USERS

is displayed when no user's Vector Facility status is loaded into a real Vector Facility.

IPL

Privilege Class: G

Use the IPL command to simulate an initial program load function for your virtual machine.

Ipl	$\left\{ \begin{array}{l} \text{vdev } \left[\begin{array}{l} \text{cylno} \\ \text{blkno} \end{array} \right] \left[\begin{array}{l} \text{Clear} \\ \text{NOClear} \end{array} \right] \text{ [STOP] [ATTN]} \\ \text{systemname} \end{array} \right\} \left[\begin{array}{l} \text{LOADParm load-parameter} \\ \text{PARM [parameter-string]} \end{array} \right]$
-----	--

Note: The operands CLEAR, STOP, and ATTN cannot be used with IPL *systemname*.

where:

vdev

specifies the virtual device that you want to IPL.

cylno

blkno

is the cylinder containing the IPL data. If this operand is specified, CP loads the IPL data from the specified virtual cylinder instead of from virtual cylinder 0. Virtual cylinder 0 is the default when cylno is not specified. This operand is only for virtual direct access storage devices. Block numbers refer to blocks that contain IPL data for fixed-block architecture devices. If you do not specify a block number, block 0 is used.

CLEAR

sets the virtual storage space to binary zeros before the operating system is loaded.

NOCLEAR

allows the contents of your virtual storage space to remain unchanged prior to program load. NOCLEAR is the default.

STOP

halts the virtual machine during the IPL procedure just after the initial PSW is loaded. This permits you to simulate an IPL procedure in instruction step mode for your virtual machine.

When your virtual machine stops, you can issue CP commands. To restart the virtual machine, issue the CP BEGIN command.

ATTN

sends an attention interruption to your virtual machine during the IPL procedure. When you are running OS/VS1 in your virtual machine, the attention interruption automatically activates the OS/VS1 FASTNIP feature.

LOADPARM load-parameter

specifies a 1- to 8-character load parameter. If fewer than 8 characters are specified, the load parameter is left-justified and padded with blanks. It may be necessary to enclose the load parameter in single quotes (see "Usage Notes" below). If LOADPARM is not specified, a load parameter of 8 EBCDIC blanks is formed.

The load parameter is returned to your virtual machine in the data block returned from the READ SCPINFO command issued by the SERVC and

DIAGNOSE code X'80' instructions. It is preserved until the system resets the load parameter to 8 EBCDIC blanks.

systemname

is a named saved system that was previously saved with the SAVESYS command.

Note: You cannot load a named saved system into a V=F or V=R machine. An attempt to do so results in an error message.

PARM [parameter-string]

specifies a parameter string up to 64 bytes long to pass to your virtual machine in general purpose registers at completion of IPL. The parameter string begins with the first non-blank character following PARM. It consists of all remaining characters which follow on the command line, including trailing blanks. If you specify other optional parameters on the IPL command, you must enter PARM last.

If you are IPLing a device, the parameter string is inserted into the virtual machine registers, four bytes per register, starting with register zero. If the string is less than 64 bytes, a single byte of binary zeroes is inserted following the string. If you enter PARM followed only by blanks, the byte of zeroes is passed in the high order byte of register zero. If you omit PARM, the virtual machine registers are unchanged.

If you are IPLing an NSS which was defined with the "PARMREGS=m-n" option on the DEFSYS command, the parameter string is inserted into the virtual machine registers "m" through "n", which are first initialized to binary zeroes. For further information, refer to the DEFSYS command. If you enter a string too large to fit in the designated registers, an error message is issued and the command is not processed. If you specify PARM followed by all blanks or if you omit PARM, the virtual machine register(s) specified by the PARMREGS option on the DEFSYS command will contain all zeroes.

If you are IPLing an NSS which was defined with the PARMREGS=NONE option on the DEFSYS command, then specifying the PARM option results in an error message being issued and in the command not being processed.

If you are IPLing an NSS which was defined without the PARMREGS=m-n option on the DEFSYS command, the parameter string is inserted into the virtual machine registers 0-15. The registers are not first initialized to binary zeroes. If you specify PARM followed by all blanks or if you omit PARM, the virtual machine registers are unchanged.

Usage Notes

1. IPL simulates the LOAD operation on the real system console. The specified virtual device is accessed and the required input/output operations are performed to retrieve the IPL data. This does not pertain to named saved systems.
2. Optionally, you can stop the IPL procedure after loading the virtual PSW. Also, you can pass parameters to the virtual machine's general registers. When the simulated load function is complete, CP initiates execution of the virtual machine by loading the IPL PSW, which was stored during the simulation process.
3. The IPL command terminates all IUCV connections and releases the IUCV buffer.

4. Any logical lines following the IPL command are ignored because execution of the IPL command transfers control from the CP environment to the virtual machine environment.
5. The IPL sequence for the program that you want to IPL must use only format 0 CCWs.
6. After you issue the IPL command, the pseudo page fault facility is deactivated. To activate (or reactivate) the pseudo page fault facility, you must issue the SET PAGEX command with the ON operand. For more information, see usage note 4 on page 539 in the description of the SET PAGEX command.
7. Care must be used when passing parameters to a named saved system (systemname). Named saved systems may expect certain registers to be initialized when they are given control. Indiscriminate use of the PARM operand could overlay a previously initialized register, causing unpredictable results.
8. When a systemname is equal to a hexadecimal device number, the device is IPLed if it exists; otherwise, the named saved segment is IPLed.
9. When the system is IPLing an NSS, the CLEAR function is always performed.
10. If you do not want leading blanks, embedded blanks, or single quotes in the load parameter, you may specify the load parameter as a single blank-delimited token. Specifying the load parameter within single quotes gives you leading blanks, embedded blanks, or single quotes. Remember that any single quote that is part of the load parameter must be doubled; for example, use LOADPARM 'BETSY''S' to specify BETSY'S as the load parameter.
11. The load parameter may not contain a X'15' character.
12. You may specify both the LOADPARM and PARM options when IPLing by either vdev or systemname. Both options may be specified on the same command line. However, if you specify the PARM option, it must be the last option on the command.
13. If you IPL an NSS that was created with the VMGROUP option on the DEFSYS command, your virtual machine becomes a member of the virtual machine group that is known by the NSS name. Members of a virtual machine group are permitted to connect to the signal system service and to provide intragroup signaling, including awareness messages (signal-in and signal-out) about members joining the group or leaving it.

If your machine is a member of a virtual machine group, a user with an appropriate privilege class may authorize trace data recording into a system data file for the group by issuing a TRSAVE command and specifying "For sysname".
14. The IPL command causes a virtual machine to be reset. If the virtual machine was a member of a virtual machine group (from a prior IPL of a VMGROUP NSS), that group membership is canceled. The virtual machine's IUCV connection to the signal system service is severed and a signal-out message is broadcast by the signal system service to all other members of the group enabled for signal-out notification.
15. If your virtual machine uses the DIAGNOSE X'98' facilities for locking virtual machine pages in real storage, issuing IPL unlocks those pages in certain cases. If you have locked pages in private virtual machine storage, issuing "IPL sysname" or "IPL vdev CLEAR" unlocks those pages. IPL releases all NSSs and saved segments from your virtual machine. Locked shared pages are

IPL

unlocked only by DIAGNOSE X'98' or by the release of the shared segment by the last user.

Responses

Usually, after a successful IPL, any responses you receive are those from the operating system that was loaded and initialized.

Response 1

When tracing is active, you get the following response:

Trace active at IPL

Response 2

When the STOP option is specified, you get the following response after the IPL is complete:

IPL stop at xxxxxxxx

where:

xxxxxxx

is the address where execution of the loaded program starts.

After this response is issued, your virtual console is placed into CP READ so that you may enter CP commands (for example, TRACE). After entering all desired CP commands, enter the BEGIN command to start executing the loaded program.

Migration Notes

VM/SP HPO: VM/XA SP does not support the PMA operand, which is supported in VM/SP HPO.

VM/XA SF: VM/XA SP does not reset the virtual console CONMODE setting during IPL.

LINK**Privilege Class: G**

Use the LINK command to make a device that is associated with another virtual machine available to your virtual machine configuration.

LINK	[TO] {userid} vdev1 [AS] vdev2 [mode] [[PASS=] password]
	{ * }

where:

[TO] userid

is the userid in the system directory whose entry is to be searched for device vdev1. If you omit the keyword TO, you cannot specify userid as T or TO.

[TO] *

indicates your devices are to be searched for device vdev1.

vdev1

is the specified user's virtual device number, as defined in the user's entry in the system directory.

[AS] vdev2

is the virtual device number that is to be assigned to the device for your virtual machine configuration.

mode

is the access mode. Valid modes are:

Mode	Meaning
R	Read-only access. The link will not be established if any other user has a link to the device in write mode.
RR	Read-only access. The link will be established in all cases.
W	Write access. The link will not be established if any other user has a link to the disk.
WR	Write access. If another user has a link to the disk, read-only access will be established.
M	Multiple-write access. The link will not be established if any other user has a link to the disk in write mode.
MR	Multiple-write access. If another user has a link to the disk in write mode, read-only access will be established.
MW	Multiple-write access. This link will be established in all cases.

If you do not specify a mode, the default is R if you are linking to another user's device. If you are linking to one of your own devices, the default is the user access mode specified in your entry in the system directory.

Warning: Permanent data loss can result if more than one user is writing to the same virtual disk.

Unpredictable results can occur when one user has read-only access to a device that is being updated by another user who has read and write access to the device.

[PASS =] password

is a one- to eight-character string. The password must match the access mode password specified for device vdev1 as indicated in the system directory entry for the userid that you specify. If a password is not specified and it is required, the user will be prompted for one.

Usage Notes

1. When linking to another user's non-fullpack minidisk on a cached control unit, you will be granted the cache access that is found in that user's directory.
2. When linking to another user's fullpack minidisk on a cached control unit, you will get a subsystem control level of NOCTL. NOCTL means that no CCWs are accepted that can exercise control of subsystem resources or functions, regardless of whether they directly relate to the device. For example, subsystem storage may not be turned on or off and caching may not be turned on or off for the device.
3. CP verifies the LINK command operands you enter against the information contained in the other user's entry in the system directory. If the other user has temporarily redefined any of this information, you must still specify the information as it is indicated in the directory, or you will receive an error message.
4. Do not confuse a LINK access mode password with a user LOGON password. You use a user LOGON password to gain access to the VM/XA SP system. A LINK access mode password controls whether you can access another user's DASD device using the LINK command.
5. If the operand immediately following vdev2 is not a valid access mode, the default mode will be established, and the operand will be interpreted as the access mode password. If you are using the default for the mode, the access mode password that you enter can not be the same as any of the access modes (R, RR, W, WR, M, MR, or MW).
6. If you are linking to one of your own disks, or if the access mode password is ALL (indicating that any user can use the device), you do not have to enter the password. You may specify userid as an asterisk (*) to indicate that the device is in your entry in the system directory; if you do specify a password, the system does not check it.

Note: If you are linking "*" to an MDISK statement defined by a LINK statement in your directory and the Resource Access Control Facility (RACF) is installed, then RACF processing will be invoked to see if you are in the RACF access list for the indirect link.
7. If you are linking to a multiple-exposure device, both vdev1 and vdev2 must specify the base exposure. All exposures will be processed.
8. You cannot request virtual reserve/release processing using the LINK command. Use the system directory MDISK statement to request virtual reserve/release processing.
9. If your installation is using the password suppression facility (controlled by the SET PASSWORD command), you will receive an error message if you enter the access mode password on the LINK command line. Wait for the "ENTER PASSWORD" prompt before you type in the password. The password will be automatically masked, thereby improving system security.

10. The access allowed by the LINK command to the vdev1 device belonging to userid is summarized below.

If you want to link to a device in *READ* mode:

Access mode you entered on the command line	Is a READ password specified in the system directory	Are there any other links to the device	Type of link that you will establish
R	No	Not applicable	None
	Yes	No	READ
	Yes	Yes - READ	READ
	Yes	Yes - WRITE	None
RR	Yes	Yes - READ	READ
	Yes	Yes - WRITE	READ

If you want to link to a device in *WRITE* mode:

Access mode you entered on the command line	Is a WRITE password specified in the system directory	Are there any other links to the device	Type of link that you will establish
W	No	Not applicable	None
	Yes	No	WRITE
	Yes	Yes - READ	None
	Yes	Yes - WRITE	None
WR	Yes	No	WRITE
	Yes	Yes - READ	READ
	Yes	Yes - WRITE	READ
M	No	Not applicable	None
	Yes	No	WRITE
	Yes	Yes - READ	WRITE
	Yes	Yes - WRITE	None

Access mode you entered on the command line	Is a WRITE password specified in the system directory	Are there any other links to the device	Type of link that you will establish
MR	Yes	No	WRITE
	Yes	Yes - WRITE	READ
	Yes	YES - READ	WRITE
MW	Yes	Yes - READ	WRITE
	Yes	No	WRITE

LINK

Response 1

If your installation is using the password suppression facility, you will be prompted to enter the access mode password with one of the following messages:

ENTER READ PASSWORD:

ENTER WRITE PASSWORD:

ENTER MULT PASSWORD:

Enter the appropriate password to obtain access to the desired disk.

Note: There is a system threshold value that disables the LINK command when a user continues to enter trial passwords to obtain a link to a minidisk. The maximum is ten incorrect passwords before the command is disallowed for the remainder of the logon session.

Response 2

DASD vdev2 LINKED R/O

indicates that a read-only link has been established to the indicated disk with a mode of R or RR, and that no other users are linked to the same disk.

Response 3

DASD vdev2 LINKED R/W

response indicates that a read/write link has been established to the indicated disk with a mode of W, WR, M, MR, or MW, and that no other users are linked to the same disk.

Response 4

DASD vdev2 LINKED R/O; R/W BY $\left\{ \begin{array}{l} \text{nnnnn USERS} \\ \text{userid} \end{array} \right\} \left[; \text{R/O BY } \left\{ \begin{array}{l} \text{nnnnn USERS} \\ \text{userid} \end{array} \right\} \right]$

indicates that a read/only link has been established to the indicated disk with a mode of RR, and warns that the disk is linked read/write by one or more users and possibly linked read/only by one or more users. If only one user has the given type of link, the number of users (nnnnn USERS) is replaced by that user's userid.

Response 5

DASD vdev2 LINKED R/W; R/O BY $\left\{ \begin{array}{l} \text{nnnnn USERS} \\ \text{userid} \end{array} \right\}$

indicates that a read/write link has been established to the given disk with a mode of M, MR, or MW, and informs you that the disk is also linked read/only by one or more other users. No other read/write links to the disk exist. If only one user has access, the number of users (nnnnn USERS) is replaced by that user's userid.

Response 6

DASD vdev2 LINKED R/W; R/W BY $\left\{ \begin{array}{l} \text{nnnnn USERS} \\ \text{userid} \end{array} \right\} \left[; \text{R/O BY } \left\{ \begin{array}{l} \text{nnnnn USERS} \\ \text{userid} \end{array} \right\} \right]$

indicates that a read/write link has been established to the given disk with a mode of MW, and warns you that the disk is linked read/write by one or more users and possibly linked read/only by some users. If only one user has access, the number of users (nnnnn USERS) is replaced by that user's userid.

Migration Notes*VM/SP HPO*

1. In VM/XA SP, for multiple exposure devices you must specify the base exposure. LINK provides access to all exposures.
2. The VM/XA SP response has five digits for the number of users.

LOADBUF
Privilege Class: D

Use the LOADBUF command to load the universal character set (UCS) buffer with a specified print chain/train image for the IBM 1403, 3203 Model 5, 3211, or 3262 printer, or to load the forms control buffer (FCB) with a specified image for the IBM 3203 Model 5, 3211, 3262, 4245 or 4248 printer. (You cannot load an FCB for a 1403 printer, and you cannot load a UCS for a 4245 or 4248 printer.)

Note: To load the forms control buffer (FCB) with a specified image for a *virtual* spooled 3203, 3211, 3262, 4245, or 4248 printer, use the LOADVFCB command. Also note that the INDEX option is valid only for the 3211 printer.

LOADBUF	rdev	{ UCS name [Ver FCB name [Index [nn]]	[Fold NOFOLD] }
---------	------	---	----------------------

where:

rdev

is the real device number of the printer whose buffer is to be loaded.

UCS name**FCB name**

specifies that you are loading a print chain image (UCS) or a forms control buffer (FCB). If you are loading both a print chain image and a forms control buffer, you must issue two LOADBUF commands for the specified printer.

name is the one- to four-character name of the UCS or FCB image that you are loading. The IBM-supplied names are given in the tables that follow.

IBM supplies the following UCS buffer images for the 1403 and 3203 Model 5 printers:

Name	Meaning
AN	Normal alphanumeric notation character set arrangement
HN	Normal hexadecimal notation character set arrangement
PCAN	Preferred alphanumeric notation character set arrangement
PCHN	Preferred hexadecimal notation character set arrangement
QN	PL/I - 60 graphics
QNC	PL/I - 60 graphics
RN	FORTTRAN, COBOL commercial
YN	High speed alphanumeric
TN	Text printing 120 graphics
PN	PL/I - 60 graphics
SN	Text printing 84 graphics

IBM supplies the following UCS buffer images for the 3211 printer:

Name	Meaning
A11	Standard Commercial
H11	Standard Scientific
G11	ASCII
P11	PL/I
T11	Text Printing

Name	Meaning
P48	48-character belt
P52	52-character belt (Austria/Germany)
P63	63-character belt, optimized
P64	64-character belt
P96	96-character belt
P116	116-character belt (French-Canadian)
P128	128-character belt (Katakana)

IBM supplies the following FCB images for the 3203 model 5, 3211, 3262, 4245, and 4248 printers:

Name	Meaning
FCB1	Space 6 lines/inch Length of page 66 lines

Line Represented	Channel Skip Specification
1	1
3	2
5	3
7	4
9	5
11	6
13	7
15	8
19	10
21	11
23	12
64	9

Name	Meaning
FCB8	Space 8 lines/inch Length of page 68 lines

Line Represented	Channel Skip Specification
1	1
4	2
8	3
12	4
16	5
20	6
24	7
28	8
32	10
36	11
63	12
66	9

Name	Meaning
FCBS	Space 8 lines/inch Length of page 68 lines

Line Represented	Channel Skip Specification
1	1
54	2
55	3
56	4
57	5
58	6
59	7
60	8
61	10
62	11
63	12
64	9

For the exact contents of these buffer images, see *IBM 3211 Printer, 3216 Interchangeable Train Cartridge, and 3811 Printer Control Unit Component Description and Operator's Guide*.

For information on how to add more FCB and UCS images, see *VM/XA SP Planning and Administration*.

VER

prints the contents of the buffer on the specified printer. This allows you to verify visually that the correct buffer was loaded.

INDEX [nn]

starts printing the output in the designated (nn) print position. "nn" must be a number from 1 to 31; a leading zero need not be specified. If you do not specify INDEX, no indexing will be performed.

If you specify INDEX without a value, the index value specified in the FCB macro becomes the index value. If you specify a value, it will override the value that you coded in the FCB macro.

Note that the INDEX option is valid only for the 3211 printer; it is not valid for the 3203 Model 5 printer.

FOLD**NOFOLD**

If FOLD is specified, all characters will be printed in uppercase. If NOFOLD is specified, the characters will not be folded into uppercase. If neither FOLD nor NOFOLD is specified, NOFOLD is the default. FOLD is used when a file specifies lowercase but no lowercase UCS is available.

Note: If you specify the FOLD option, the control unit ignores bit positions 0 and 1 of the EBCDIC code when comparing the character codes from the UCS buffer and the print line buffer. Only uppercase characters will be printed from either uppercase or lowercase data codes.

Warning: Misuse of the FOLD option can cause multiple characters to be printed for the same EBCDIC bit configuration.

Usage Notes

1. You must drain the printer before you issue the LOADBUF command.
2. The operator who is authorized to load the UCS and/or FCB buffer should use this command:
 - Whenever the printer or control unit is powered on.
 - Whenever the print chain or train is to be changed.
 - If the buffer has been causing an excessive amount of parity errors.
 - If the printer was previously attached to a user's virtual machine.
3. Load the FCB buffer:
 - Whenever the FCB buffer is to be changed.
 - If the printer was previously attached to a user's virtual machine.
4. FOLD or NOFOLD is accepted with or without the UCS or FCB options. If specified without either option, the UCS or FCB will not be changed, but the characters will be folded into upper case (or no longer folded if NOFOLD is specified).
5. The LOADBUF command automatically sets the Block Data Check latch for the UCS and/or FCB buffer load. If you specify VER, the image of the UCS load will be printed on the printer you indicated. The printed output must correspond to the description of the buffer load in the publications:
 - *IBM 2821 Control Unit*, or
 - *IBM 3203 Printer Model 5 Component Description and Operator's Guide* or
 - *IBM 3262 Printer Model 1 and 11 Component Description*, or

LOADBUF

- *IBM 3211 Printer, 3216 Interchangeable Train Cartridge, and 3811 Printer Control Unit Component Description and Operator's Guide.*

Responses

None.

Migration Notes

VM/SP HPO: FOLD or NOFOLD is accepted with or without the UCS or FCB options. If specified without either option, the UCS or FCB will not be changed. But the characters will be folded into uppercase (or no longer folded if NOFOLD is specified). If a 4245 or 4248 is specified with the UCS option, an error message will result.

LOADVFCB

Privilege Class: G

Use the LOADVFCB command to specify the forms control buffer image for a virtual impact printer.

Note: To load the forms control buffer (FCB) with a specified image for a *real* IBM 3203 model 5 or 3211, 3262, 4245, or 4248 printer, use the LOADBUF command.

LOADVFCB	vdev {FCB name} [Index [nn] [IMAGE image1ib]]
----------	--

where:

vdev

is the virtual device number of your virtual spooled impact printer.

FCB name

specifies the forms control buffer image for your virtual printer. name is the 1- to 4-character name of the FCB image. You may specify either the name of an IBM-supplied FCB image or the name of another FCB image supplied by your system programmer. The three IBM-supplied FCB images are FCB1, FCB8, and FCBS. The default FCB image is FCB1. The formats of the IBM-supplied FCB images are as follows:

Name	Meaning
FCB1	Space 6 lines/inch Length of page 66 lines

Line Represented	Channel Skip Specification
1	1
3	2
5	3
7	4
9	5
11	6
13	7
15	8
19	10
21	11
23	12
64	9

LOADVFCB

Name	Meaning
FCB8	Space 8 lines/inch Length of page 68 lines

Line Represented	Channel Skip Specification
1	1
4	2
8	3
12	4
16	5
20	6
24	7
28	8
32	10
36	11
63	12
66	9

Name	Meaning
FCBS	Space 8 lines/inch Length of page 68 lines

Line Represented	Channel Skip Specification
1	1
54	2
55	3
56	4
57	5
58	6
59	7
60	8
61	10
62	11
63	12
64	9

For the exact contents of these FCB images, see *IBM 3211 Printer, 3216 Interchangeable Train Cartridge, and 3811 Printer Control Unit Component Description and Operator's Guide*.

INDEX [nn]

is the first print position. The value nn must be a number from 1 through 31; a leading zero need not be specified. If you enter the keyword INDEX without a value, the value defaults to the value specified in the FCB image.

Note that the INDEX option is valid only for a virtual spooled 3211 printer; it is not valid for other impact printers.

IMAGE imagelib

specifies the image library which contains the specified FCB. If you do not specify the IMAGE option, you will get the default image library for the device type. The default image library is IMAGxxxx, where xxxx is the device type; for example, the default image library is IMAG1403 for a 1403 type printer.

Usage Notes

1. The virtual FCB image should match the FCB image on the real printer. If it does not, results are unpredictable.

If the spool file is printed on a real 1403 printer, the virtual FCB image is ignored and channel skipping is controlled by the carriage tape.

2. If you are running a program in your virtual machine that requires printer overflow sensing information and you are printing on a real 1403, you need to take special steps in order to retrieve the sensing information. First, ensure that the VM/XA SP directory entry for your virtual machine indicates that the printer is a 3203, 3211, 3262, 4245, or a 4248, even though the program and the guest operating system in which the program runs define the printer as a 1403. Then, use the LOADVFCB command to specify a virtual FCB image that matches the 1403. VM/XA SP uses this virtual FCB image as a basis for retrieving sensing information.
3. If the virtual machine operating system issues a LOAD FCB channel command word to the virtual spooled printer, the LOAD FCB CCW image replaces the virtual FCB image provided on the LOADVFCB command line. Note that the CMS PRINT command issues a LOAD FCB CCW and thereby replaces the virtual FCB image.
4. The LOADVFCB command allows specification of the extended FCB image for a virtual 4248 printer. If an attempt is made to send the extended FCB image to a printer that does not support the extended FCB format, an error message results.

Responses

None.

LOCATE

Privilege Class: C, E

Use the LOCATE command to display the address of CP control blocks associated with a user, a user's virtual device, or a real system device. You can also use the LOCATE command to display the address of a module identified by the symbolic label and to indicate whether it is resident or pageable.

LOCate	$\left. \begin{array}{l} \text{userid [vdev]} \\ \text{rdev} \\ \text{symbol} \end{array} \right\}$
--------	---

where:

userid

is the user identification of the virtual machine whose CP control blocks you are attempting to locate. You can specify userid as an asterisk (*) if you want to display the address of control blocks associated with your own virtual machine.

vdev

is a virtual device number. The address of the virtual device block (VDEV) and, if the device is a spooled device, the address of the virtual device simulation block (VDSBK) is displayed.

rdev

is a real device number. The address of the real device block (RDEV) and its associated control blocks, if any, are displayed.

symbol

is the symbolic label for a module. The address for the symbolic label is displayed. The symbol must be an entry point in the system.

Usage Notes

The *rdev* option is not valid for SNA/CCS terminals, which are connected through a VTAM service machine and which have no real device address known to CP. However, a dynamic RDEV control block exists for each logged on or dialed SNA/CCS terminal. For SNA/CCS terminals, the user finds the RDEV control block by using the *userid vdev* format of LOCATE. This format returns the address of the VDEV control block, and the VDEV contains a pointer to the RDEV.

Responses

All of the following responses display the address, in hexadecimal, of the information associated with your LOCATE request.

Note: For all the responses, if "*****" (a display of eight asterisks) is shown, then the corresponding CP control block does not exist.

Response 1

VMDBK = xxxxxxxx

Response 2

If the device you specify in vdev is a spooled device, the following response is displayed:

```
VMDBK      VDEV      VDSBK      VPXBK      VSPBK
XXXXXXXXX  XXXXXXXXX  XXXXXXXXX  XXXXXXXXX  XXXXXXXXX
```

If the device you specify in vdev is not a spooled device, you receive the following response:

```
VMDBK      VDEV
XXXXXXXXX  XXXXXXXXX
```

Response 3

```
RDEV      CPVOL      ALOC
XXXXXXXXX  XXXXXXXXX  XXXXXXXXX
```

Response 4

```
RDEV      RSPBK      SPFBK
XXXXXXXXX  XXXXXXXXX  XXXXXXXXX
```

Response 5

```
RDEV
XXXXXXXXX
```

Response 6

If in a resident module:

```
symbol = XXXXXXXX RESIDENT
```

where 'XXXXXXXX' is an address in first-level storage.

If in a pageable module:

```
symbol = XXXXXXXX PAGEABLE
```

where 'XXXXXXXX' is a CP virtual address.

Migration Notes

VM/SP HPO:

1. In VM/XA SP you may specify a CP symbol to be located.
2. The VM/XA SP response provides different control blocks than in VM/SP HPO.

LOCK

Privilege Class: A

Use the LOCK command to permanently lock in real storage selected pages of the pageable CP nucleus or selected pages of a user's virtual storage, thus excluding them from future paging activity.

LOCK	$\left\{ \begin{array}{l} \text{userid page1 pagen} \\ \text{SYSTEM page1 pagen} \\ \text{symbol} \end{array} \right\} \text{ [MAP]}$
------	---

where:

userid

is the user identification of a logged on user. If you are locking your own virtual pages, you can specify userid as an asterisk (*).

SYSTEM

locks one or more of the pageable CP pages. Any pageable pages are locked until released by the UNLOCK command, by a system clear, or a restart.

page1

is the hexadecimal value of the first user page to be brought into storage and locked.

pagen

is the hexadecimal value of the last user or SYSTEM page to be brought into storage and locked. If only one page is to be brought into storage, pagen must be the same as page1.

symbol

causes the page containing this symbol to be locked. This symbol must be one of the CP system symbols; and may be in either CP resident nucleus or the CP pageable area.

MAP

prints a map of the virtual storage pages locked and the real page that is assigned for each page locked. The MAP option is the default if the "symbol" form is used. The response to the MAP operand is:

VIRTPAGE	REALPAGE
virt-firstpage	real-firstpage
.	.
.	.
.	.
virt-lastpage	real-lastpage

Usage Notes

1. If the reserved page frames option is already in use by another virtual machine, you can use the LOCK command to enhance the efficiency of a particular virtual machine by keeping high activity pages (such as virtual page zero) in real storage. If the number of page frames available for paging is limited, you should not issue the LOCK command without the approval of your system programmer.

2. If too many pages of real storage are locked, the remaining virtual machines may not have enough page frames available to operate efficiently. System throughput may be severely degraded in all virtual machines because of the excessive contention for the remaining available page frames.
3. Once a page is locked, it remains locked until you log off the system or until you issue a CP UNLOCK command for that page. If you have the "locked pages" option in effect and you re-IPL your system or load another system, your locked pages are unlocked and are made available to the system being loaded.
4. Locking V = R and V = F storage has no effect on their performance since their storage is already fixed.
5. When the page range specified covers one or more RIO370 area frames, the command is a NO-OP for these frames. These frames are marked as being CP-locked when the RIO370 area is initialized.
6. The CP SYSTEM CLEAR command clears any locked pages to binary zeros.
7. The virtual pages locked in processor storage are 4K (4096) byte blocks. This block of storage need not represent all of the user's virtual storage. The LOCK command may be issued as many times as required for one virtual machine to lock noncontiguous pages of storage. The remaining virtual machine storage blocks remain pageable.
8. If you omit the MAP option from a LOCK command and you later want to find the real page, reissue the LOCK with the MAP option. No intervening UNLOCK is required. The MAP option is the default if the 'symbol' form is used.
9. The SYSTEM and MAP operands are useful when you want to examine a page that is normally not resident (such as console function routines). The MAP operand locates the real page for display purposes.
10. Page numbers are defined as follows:

Storage Locations (in hexadecimal)	Page Number
000-0FFF	0
1000-1FFF	1
2000-2FFF	2
.	.
.	.
12000-12FFF	12
.	.
.	.

Examples

1. For firstpage and lastpage, only the page numbers are specified. For example, to lock USERA's virtual storage locations X'12000' thru X'2CFFF' in real storage, the following command should be given:

```
LOCK USERA 12 2C
```

To lock the first three pages of a USERA's storage, enter:

```
LOCK USERA 0 2
```

LOCK

2. To lock a CP page for display purposes, key in the following lines:

```
LOCK SYSTEM 2A 2A MAP
VIRTPAGE REALPAGE (response to the MAP operand)
0002A000 0004F000
COMMAND COMPLETE (response to the LOCK command)
DISPLAY H4F038-4F048
etc.
UNLOCK SYSTEM 2A 2A
```

Responses

COMMAND COMPLETE

indicates the successful completion of the command.

Migration Notes

VM/SP HPO

1. In VM/XA SP you may specify a CP system symbol to lock the page containing that symbol.
2. VM/XA SP issues an error message if an attempt is made to lock nonpageable storage, unless you use the symbol operand.

LOGOFF

Privilege Class: Any

Use the LOGOFF command to terminate virtual machine execution and to remove your virtual machine from the VM/XA SP system.

LOGoff LOGout	[HOLD]
------------------	----------

where:

HOLD

causes the TTY display terminal telecommunication connection to the 7171 ASCII DACU to remain in effect after your session is ended.

Usage Notes

1. When you LOGOFF, all active spool files are closed, temporary disks are relinquished, dedicated devices are detached, and an accounting record is created for your session.
2. The LOGOFF command terminates all IUCV connections and releases the IUCV buffer.
3. Always log off your display in addition to turning off the power; turning off your display is not the same as logging off.

If you turn off the power at your display without logging off, your virtual machine is logged off 15 minutes after VM/XA SP determines that your display has been turned off. If you again turn on the display, your virtual machine will be disconnected. You can reconnect the display to your virtual machine by logging on within the 15 minute interval; if you do not log on within the interval, your virtual machine is automatically logged off.

4. If you specify the HOLD option on an SNA/CCS terminal, the command is invalid and an error message is issued. If you specify the HOLD option on a TTY display terminal connected to a 7171 ASCII DACU, it is accepted and processed. The HOLD option is accepted but ignored on all other terminals.
5. If your installation has coded the E3270HLD feature in the system real I/O configuration file for your TTY display terminal, the telecommunication connection always remains in effect, regardless of the HOLD option.

Response 1

```
CONNECT= hh:mm:ss VIRTCPU= mmm:ss.hs TOTCPU= mmm:ss.hs
LOGOFF AT hh:mm:ss zone weekday mm/dd/yy
```

where:

CONNECT = hh:mm:ss

is the actual clock time in hours:minutes:seconds spent in the current session.

VIRTCPU = mmm:ss.hs

is the virtual processor time in minutes:seconds.hundredths of seconds used in the current session.

LOGOFF

TOTCPU = mmm:ss.hs

is the total processor time in minutes:seconds.hundredths of seconds (including virtual and overhead) used in the current session.

These times are either the elapsed time for the entire session or the elapsed time since the ACNT command was entered for this user.

LOGOFF AT

is the response you receive for your logoff.

hh:mm:ss

is the current time in hours:minutes:seconds.

zone

is the time zone.

weekday

is the current day of the week.

mm/dd/yy

is the date in month/day-of-month/year format.

Response 2

```
{ type { rdev } } LOGOFF AS userid USERS = nnnnn  
  { ldev }  
  { luname }
```

This response is sent to the primary system operator when you logoff VM/XA SP.

If the virtual machine had been logged on to the VM/XA SP system using the CP AUTOLOG command, this response is sent to both the primary system operator and to the user who issued the AUTOLOG command.

where:

type

is the user's display type.

rdev

ldev

is the real or logical device number of the user's display.

luname

identifies an SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

AS userid

is the identification of the user who is logging off.

USERS = nnnnn

is the number of users still on the system.

Response 3

XSTORE DETACHED FROM userid

This response is sent to the primary system operator when Expanded Storage is detached from a user who has just logged off (if the primary system operator is not the one logging off).

Migration Notes

VM/SP HPO: The VM/XA SP response has five digits for the number of users.

VM/XA SF: VM/XA SP uses GRAF rather than DISP in its response.

LOGON**Privilege Class:** Any

Use the LOGON command to identify yourself to VM/XA SP and to access the system.

Logon Login	userid [password] [Ipl { vdev { sysname } }] [#console-input-data] [NOipl] [ACcount nnnnnnnn] [NORUN] [APL] [TEXT] [QUIET] [Storage { nnnnnnnk } { nnnM }]
----------------	--

where:

userid

is the identifier assigned to you in the system directory.

password

is the userid's LOGON password from the user's directory. VM/XA SP accepts passwords up to 40 characters in length. Refer to usage note 2 on page 245 for details on when this operand cannot be used.

**IPL { vdev
 { sysname } }**

specifies the virtual device number or named saved system that you want IPLed upon completion of your logon processing. This option overrides the IPL statement (if any) specified in your entry in the system directory. The validity of the IPL operand that you specify is determined by the IPL command processor after your logon is completed; it is not verified as part of the LOGON command processing.

NOIPL

specifies that the IPL device specified in your entry in the system directory should not be used for an automatic IPL.

ACCOUNT nnnnnnnn

specifies an alternate account number to be assigned to your virtual machine at logon time. This account number must be valid in order for the LOGON command to complete successfully. If you do not specify this option, the primary (or first) account number specified in your entry in the system directory is used.

NORUN

specifies that upon successful completion of the user logon the virtual machine is placed into a terminal CP READ state and RUN is set OFF.

**APL
TEXT**

specifies that upon successful completion of the user logon the control program is to use an APL or TEXT translation table for translating console input and output rather than the standard translation table. This option of the LOGON command provides the same function as the TERMINAL APL ON or the TERMINAL TEXT ON commands.

STORAGE { **nnnnnnnK** }
{ **nnnM** }

specifies an alternate storage size to be assigned to your virtual machine. The value that you specify for this option must be valid in order for the logon to complete successfully. If you do not specify a value, the minimum storage size indicated in your entry in the system directory is used. The minimum value that you can specify is 64K; the maximum value allowed is the maximum value specified on the USER statement for your userid in the system directory and cannot exceed 999M. If the value supplied is not a multiple of 64K, it is rounded up to the next 64K boundary if the value is less than 16M or it is rounded up to the next 1M boundary if the value is greater than 16M.

QUIET

suppresses the display of all informational LINK messages, system log-messages, and information regarding spooled virtual reader, printer, and punch files.

#console-input-data

is data that you want used as virtual console input in response to the first read your virtual machine issues. You may use the remainder of the command line to enter console data; the entire line including any logical line-end characters (#) are passed to your virtual machine.

Usage Notes

1. Upon successful logon, VM/XA SP creates a virtual machine configuration for you based upon information contained in the system directory.
2. If your installation is using the password suppression facility for the LOGON command (controlled by the SET PASSWORD command), you cannot specify your password on the command line. You must wait for the "ENTER PASSWORD" prompt before you type in the password. The password is automatically masked, thus improving system security. If your installation is not using the password suppression facility for the LOGON command, you can specify your password on the command line or you can wait for the "ENTER PASSWORD" prompt before you type in the password.
3. Do not use any line-editing symbols when you enter the LOGON command name. You can use the system default line-editing symbols when you enter operands. However, you cannot use line-editing symbols that are defined specifically for your virtual machine.
4. If you were disconnected from your virtual machine because of a console error, you have 15 minutes to log on again. If you do not log on within 15 minutes, your virtual machine is automatically logged off. If this happens, you may have to reconstruct files and restart jobs that were interrupted by the error.
5. If your virtual machine is running in disconnected mode, and you issue the LOGON command to reconnect your console, certain SET and TERMINAL command operands are forced OFF. You can issue the CP QUERY SET and CP QUERY TERMINAL commands to check which functions are currently active.

LOGON

6. When you issue the LOGON command to reconnect to your virtual machine, the options NOIPL, IPL, ACCOUNT, and STORAGE are ignored. The console-input data line is also ignored.
7. The NORUN option is intended when RUN is set ON for users who wish to reconnect and get into CP READ. However, the option is always accepted with no effect in cases other than when RUN is set ON. The option is particularly helpful for users who need to enter CP commands and who are in full-screen mode with no break-key and with RUN set ON. These users can power off their terminals to be disconnected and then reconnect via LOGON with NORUN.
8. Keyboard/printer terminals are supported as SNA/CCS terminals via VTAM service machines. Although CP automatically causes the VTAM service machine to mask your password on these typewriter terminals to ensure installation security, the MASK operand is accepted on the LOGON command to provide a LOGON command syntax compatible with VM/SP HPO.

Response 1

ENTER PASSWORD:

indicates that your userid has been accepted. Enter your password.

Response 2

LOGMSG - hh:mm:ss zone day-of-week mm/dd/yy

indicates the time and date at which the system log message was recently revised and is displayed only when there are log messages to be displayed. All lines of the log message for which the first character is an asterisk are displayed. If you wish to see all of the system log messages, issue the CP QUERY LOGMSG command.

where:

hh:mm:ss

is the time when the log message was last updated.

zone

is the time zone.

day-of-week

is the day when the log message was last updated.

mm/dd/yy

is the month/day/year when the log message was last updated.

Response 3

FILES: nnnn RDR, nnnn PRT, nnnn PUN
 NO NO NO

where nnnn is the number of spool files of each type.

Response 4

LOGON AT hh:mm:ss zone weekday mm/dd/yy

— or —

RECONNECTED AT hh:mm:ss zone weekday mm/dd/yy

indicates the time, day of the week, and date at which the logon or reconnect is complete.

Response 5

```
{ type { rdev } } LOGON AS userid    USERS = nnnnn
  { ldev }
  { luname }
```

– or –

```
{ type { rdev } } RECONNECT userid  USERS = nnnnn
  { ldev }
  { luname }
```

is the response to the primary system operator.

type

is the display type.

rdev

ldev

is the real or logical device number of the user's display.

luname

identifies an SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

userid

is the userid of the user who has just completed logging on to the system.

USERS = nnnnn

is the number of users on the VM/XA SP system.

Response 6

```
XSTORE ATTACHED, SIZE=nnnnnnnnM
```

This response confirms that Expanded Storage has been added to your virtual machine configuration. You receive it when you have an XSTORE directory statement and Expanded Storage is available at logon.

Response 7

```
XSTORE ATTACHED TO userid
```

This response is sent to the primary system operator (if the operator is not the one logging on), indicating that Expanded Storage has been attached to the specified user's virtual machine.

Migration Notes

Because of the new password operand syntax, users' passwords cannot be any of the allowable VM/XA LOGON options if password suppression is OFF (or, in other words, if the password can be on the command line).

VM/XA SF

1. VM/XA SP uses GRAF rather than DISP in its response.

MESSAGE

Use the MESSAGE command described in this section to transmit message text to the virtual consoles of other logged-on users.

Complete Format for MESSAGE

The format box below shows all of the operands available with the MESSAGE command. Complete descriptions of the operands appear on the following pages, in the same sequence as the operands appear here.

		Class
Message	ALL msgtext	A/B
Msg	$\left. \begin{array}{l} \text{userid} \\ * \\ \text{Operator} \end{array} \right\} \text{msgtext}$	Any

General Usage Notes for MESSAGE

1. If a message is issued from a CMS environment, the command and the message text cannot be longer than 240 characters.

If a message is issued from a CP environment, the command and message length is limited by the input area of the terminal.

2. If you send a message to a user who is not logged on and has not defined a secondary user, or to a user who has suppressed the receiving of messages (that is, a user who has issued the CP SET MSG OFF command), your message will not be transmitted and you will receive an information message. The message that was transmitted will not be saved. You must send it again when the user is receiving messages.
3. The MESSAGE command is processed to route the message to the secondary user of a disconnected virtual machine. If the receiver is disconnected and has designated a secondary userid, the message is routed according to the status of that user. If the secondary userid is logged on and neither logging off nor disconnected, output is sent to the secondary user's console. The same is true if the secondary user is disconnected but handling console messages through the MSG system service. An error message indicating that a userid is not receiving or is disconnected is written to the issuer only if neither of the above conditions is satisfied.
4. Messages are displayed only when the display device is ready to receive output. This is when the status area shows the system is RUNNING. If you have a CP READ or a VM READ, messages are held until you press ENTER.

Note, however, that if you are using an SNA/CCS terminal, messages are displayed when in CP READ or VM READ.
5. If you receive a message on a 3270 virtual console that has the alarm feature, the alarm will sound.

Migration Notes

VM/SP HPO: VM/XA SP provides the time stamp and message text on the same line without blank lines in between as in VM/SP HPO.

MESSAGE ALL

Privilege Class: A, B

Use MESSAGE ALL to transmit message text to all logged on users.

Message Msg	ALL msgtext
----------------	-------------

where:

ALL msgtext

transmits the message you specify in msgtext to all logged on users. The length of the message is limited by the number of characters remaining on the input line after the command and appropriate operand are entered. You must include at least one blank between the ALL operand and the message text.

If a message is issued from a CMS environment, the command and the message text cannot be longer than 240 characters. If a message is issued from a CP environment, command and message lengths are limited by the input area of the terminal.

Usage Notes

For additional information, refer to the "General Usage Notes for MESSAGE" on page 248.

Responses

hh:mm:ss * MSG FROM userid : msgtext

where:

hh:mm:ss

is the time in hours:minutes:seconds when the message was sent to the user.

userid

is the user identification of the user sending the message.

msgtext

is the message text.

MESSAGE (user)**Privilege Class:** Any

Use MESSAGE (user) to transmit message text to a specified userid or to the primary system operator.

Message Msg	{userid * OPERATOR}	msgtext
----------------	---------------------------	---------

where:

userid

*

OPERATOR

is the identification of the user to whom you are sending a message. If you specify OPERATOR, the message is sent to the primary system operator, regardless of userid. You may specify userid as an asterisk (*) to send a message to yourself.

msgtext

is the text of the message that you are sending. The length of the message is limited by the number of characters remaining on the input line after the command and appropriate operand are entered.

If a message is issued from a CMS environment, the command and the message text cannot be longer than 240 characters. If a message is issued from a CP environment, command and message lengths are limited by the input area of the terminal.

Usage Notes

1. MSG * is useful in identifying the current userid of an abandoned display, or it can be used in a program intended to be executed under several userids.
2. You do not have to be logged on to issue the MESSAGE USER command.
3. For additional information, refer to the "General Usage Notes for MESSAGE" on page 248.

Response 1

```
hh:mm:ss * MSG FROM OPERATOR: msgtext
```

is the response received by the user when the system operator sends a message.

Response 2

```
hh:mm:ss * MSG FROM {userid  
LOGNxxxx  
LOGLxxxx  
luname}: msgtext
```

is the response received by the user when another user sends a message.

where:

hh:mm:ss

is the time in hours:minutes:seconds when the message was sent to the user.

MESSAGE (user)

userid

LOGNxxxx

LOGLxxxx

luname

is the user identification of the user sending the message. **LOGNxxxx** indicates the ID of a user in the process of logging on to a real device. **LOGLxxxx** indicates the ID of a user in the process of logging on to a logical device. *luname* indicates the logical unit name of the SNA/CCS terminal to which a user is in the process of logging on; it is the name from the VTAM definition of the terminal as an SNA logical unit. If the user is not logged on, **LOGNxxxx** or **LOGLxxxx** indicates the line number of the device from which the message originated.

msgtext

is the message text.

MONITOR

Use the MONITOR command to control the selection, collection, and reporting of data from the system.

Complete Format for MONITOR

MONitor	Event [DISable PROCessor SCheduler STORage ENable] {Block m} { STArt [Block m][PARTition n] STOP }	(I/O { DEvIce { rdev rdev... } { rdev-rdev... } CLass class-list TYpe type-list VOLume { volid-list } { CPVOL } ALL } { USERID userid-list } { ALL } } } { SEEKS { DEvIce { rdev rdev... } { rdev-rdev... } TYpe type-list VOLume { volid-list } { CPVOL } ALL } })
---------	--	---

MONITOR

MONitor	SAMPlE [<table style="display: inline-table; vertical-align: middle;"> <tr> <td style="vertical-align: middle;">I/O</td> <td style="vertical-align: middle;"> { DEvIce { rdev rdev rdev... rdev-rdev... } ClAss class-list TYPe type-list } </td> </tr> <tr> <td style="vertical-align: middle;">DISable</td> <td style="vertical-align: middle;"> { VOLume { volid-list CPVOL } ALL } </td> </tr> <tr> <td style="vertical-align: middle;">ENable</td> <td style="vertical-align: middle;"> { USER { USERID userid-list ALL } PRoCessor STORage ALL } </td> </tr> </table> { INterval n [MINutes] [SEConds] } { RATE { n [SEConds] } STOP } { STArt STOP } { STArt STOP } [BLock m] [PARTition n]	I/O	{ DEvIce { rdev rdev rdev... rdev-rdev... } ClAss class-list TYPe type-list }	DISable	{ VOLume { volid-list CPVOL } ALL }	ENable	{ USER { USERID userid-list ALL } PRoCessor STORage ALL }
I/O	{ DEvIce { rdev rdev rdev... rdev-rdev... } ClAss class-list TYPe type-list }						
DISable	{ VOLume { volid-list CPVOL } ALL }						
ENable	{ USER { USERID userid-list ALL } PRoCessor STORage ALL }						

General Usage Notes:

1. Do the following before trying to start event or sample monitoring:
 - Define and save a saved segment for MONITOR.
 - Load the monitor saved segment and connect to the *MONITOR CP system service via IUCV, using an application program running in a virtual machine.
2. Both the event and sample profiles can be changed after monitoring has started. The only exception is that the partition size of the saved segment cannot be changed unless event monitoring is stopped. New event domains will be activated upon completion of the EVENT command, and new sample domains will be activated when the next interval elapses.
3. For a detailed description of data collected, refer to the monitor records in the *VM/XA SP CP Programming Services* manual.
4. The valid classes and types for use with SEEKS and I/O domains are listed in the following table:

Table 8. Valid classes and types for SEEKS and I/O domains				
CLASS	TYPE	Other TYPES included	Valid For SEEKS	Valid For I/O
TAPE	3420			X
	3430			X
	3480			X
DASD	2305			X
	3330	3333	X	X
	3340	3344	X	X
	3350		X	X
	3370		X	X
	3375		X	X
	3380		X	X
	3380		X	X
SPOOL	1403			X
	2501			X
	2540			X
	3203			X
	3211			X
	3262			X
	3505			X
	3525			X
	3800			X
	4245			X
	4248			X
GRAF	GPRT ¹			X
	GTERM ¹			X

¹GPRT represents all graphic display printers; GTERM represents all display terminals. For a list of graphic devices supported, see *VM/XA SP General Information*.

Migration Notes

VM/SP HPO: VM/XA SP uses different commands, produces different data records, and has different collection mechanisms.

ENABLE

enables the specified domains and their elements for event data collection. Event data shall be collected only from those domains that have been enabled. ENABLE is the default.

Note: The MONITOR domain is always enabled for event monitoring.

I/O

specifies that the I/O domain is to be enabled or disabled for event data collection. An I/O domain is comprised of all real I/O devices. Event monitoring may be enabled or disabled on all devices, or it may be restricted by their device numbers, classes, types, or volumes. I/O-domain events include such activities as VARY ON and VARY OFF, attachment and detachment of devices.

I/O DEVICE rdev**I/O DEVICE rdev rdev...****I/O DEVICE rdev-rdev...**

specifies the device number of the real device to be enabled or disabled for event data collection in the I/O domain. You may specify a list, a range of devices, or a combination of both. If you specify a range, all valid devices in the range, inclusively, will be enabled or disabled. For additional information, see the specific usage notes for I/O.

I/O CLASS class-list

specifies a list of device classes that are to be enabled or disabled for event monitoring in the I/O domain. A class list is one or more device classes, separated by blanks. The valid classes are DASD, TAPE, SPOOL, and GRAF. All real devices of the specified class or classes will be monitored.

I/O TYPE type-list

specifies a list of device types that are to be enabled or disabled for event monitoring in the I/O domain. A type list is one or more device types (such as 3380), separated by blanks. For valid device types, see Table 8 on page 255. All real devices of the specified type or types will be monitored.

I/O VOLUME valid-list**I/O VOLUME CPVOL**

specifies a list of volumes that are to be enabled or disabled for monitoring in the I/O domain. A valid list is one or more volume identifiers, separated by blanks. If CPVOL is specified, then all CP owned volumes are monitored. A CP owned volume is any volume specified on the SYSCPVOL macro of HCPSYS ASSEMBLE.

I/O ALL

specifies that all real devices are to be enabled or disabled for monitoring in the I/O domain.

USER

specifies that the USER domain is to be enabled or disabled for event data collection. An USER domain is comprised of all users. All or selected users may be specified for monitoring.

User domain events include such activities as logon and logoff, defining and detaching a virtual CPU, and an end of transaction.

USER USERID userid-list

specifies a list of users that are to be enabled or disabled in the USER domain. A userid list is one or more user identifiers, separated by blanks.

USER ALL

specifies that all users are to be enabled or disabled for event data collection in the USER domain.

PROCESSOR

specifies that the PROCESSOR domain is to be enabled or disabled for event data collection. Events such as VARY ON or VARY OFF of a processor will be monitored.

SCHEDULER

specifies that the SCHEDULER domain is to be enabled or disabled for event data collection. Event data for this domain include statistics such as scheduler queue manipulation, flow of work through the system, and resource allocation strategies of the scheduler and the dispatcher.

SCHEDULER USERID userid-list

specifies a list of users that are to be enabled or disabled for event data collection in the SCHEDULER domain. A userid list is one or more user identifiers, separated by blanks.

SCHEDULER ALL

specifies that all users are to be enabled or disabled for event data collection in the SCHEDULER domain.

STORAGE

specifies that the STORAGE domain is to be enabled or disabled for event data collection. Events such as defining and purging of a named saved system or a saved segment, and attaching or detaching a CP volume are monitored.

SEEKS

specifies that the SEEKS domain is to be enabled or disabled for event data collection. A SEEKS domain is comprised of all DASD devices. Event monitoring may be enabled or disabled for all DASD devices, or it may be restricted by their device numbers, types, or volumes.

Event data will be collected on every I/O request on such enabled DASD devices.

SEEKS DEVICE rdev

SEEKS DEVICE rdev rdev...

SEEKS DEVICE rdev-rdev...

specifies the device number of a DASD device to be enabled or disabled for event data collection. You may specify a list, a range of devices, or a combination of both. If you specify a range, all valid DASD devices in the range, inclusively, are enabled or disabled. See the specific SEEKS usage notes for additional information.

SEEKS TYPE type-list

specifies a list of DASD device types that are to be enabled or disabled for SEEKS monitoring. A type list is one or more device types (such as a 3380), separated by blanks. All DASD devices of the specified type or types will be monitored. For valid device types, see Table 8 on page 255.

SEEKS VOLUME valid-list

SEEKS VOLUME CPVOL

specifies a list of volumes that are to be enabled or disabled for SEEKS monitoring. A valid list is one or more volume identifiers, separated by blanks. If CPVOL is specified, then all CP owned volumes are enabled or disabled. A CP owned volume is any volume specified on the SYSCPVOL macro of HCPSYS ASSEMBLE.

SEEKS ALL

specifies that all real DASD devices are to be enabled or disabled for SEEKS monitoring.

ALL

specifies that all domains and all their elements are to be enabled or disabled for event data collection. This includes all six domains: I/O (all devices), USER (all users), SEEKS (all DASD devices), SCHEDULER (all users), PROCESSOR, and STORAGE.

Note: The MONITOR domain is enabled for event data collection at all times.

BLOCK m

is described as part of the MONITOR EVENT START command below. If the command is issued when event monitoring is not active, the value of *m* will not be verified until monitoring has been activated by the MONITOR START EVENT or the MONITOR START commands.

START

activates event-driven data collection for the domains specified in preceding MONITOR EVENT commands. At startup, the profile records from the monitor domain are also generated.

BLOCK m

specifies the number of 4Kb shared-segment frames (*m*) that will be accumulated before the virtual machine is notified that the event data is available for data collection. The BLOCK value may be modified via the MONITOR EVENT BLOCK command while event monitoring is active.

PARTITION n

specifies that the number of 4Kb shared-segment frames (*n*) to be reserved for event data. The rest of the shared segment will then be used for sample data.

STOP

deactivates the event-driven data collection. The monitor event profile is not affected by this command, except for PARTITION which is set to 0.

General Usage Notes

1. Both event and sample monitoring can be active at the same time. Monitoring of both types of data can be started by using the MONITOR START command, or for event data by issuing the MONITOR EVENT START command. Similarly, monitoring can be stopped by the appropriate stop commands, MONITOR EVENT STOP or MONITOR STOP.
2. The following must be done before event monitoring can be started:
 - A saved segment must be defined and saved for MONITOR.
 - An application program running in a virtual machine must load the monitor saved segment and connect to the *MONITOR CP system service via IUCV.
3. The event profiles can be changed after monitoring has started. The only exception is that the partition size of the saved segment cannot be changed unless event monitoring is stopped. New event domains will be activated upon completion of the event command.
4. Event monitoring can result in large volumes of data generated, and in increased CP overhead, particularly in the SEEKS and SCHEDULER domains. These domains should be enabled only when problems are suspected.

Specific Usage Notes for I/O and SEEKS

1. Any overlapping of I/O devices can exist in a series of MONITOR EVENT commands on the I/O domain and the SEEKS domain. For example, if this command:
`MONITOR EVENT ENABLE I/O TYPE 3380`
is followed by this command:
`MONITOR EVENT ENABLE I/O CLASS DASD`
then all DASD devices will be enabled for monitoring. However, if this is followed by this command:
`MONITOR EVENT DISABLE I/O TYPE 3380`
monitoring will cease for all 3380 devices, but will continue for all other types of DASD devices.
2. A message will be issued for any device which does not exist or, for the SEEKS domain, is not a DASD device.
3. Devices which are enabled but are found to be offline will be monitored once they have been varied online or attached to the system.
4. A summary message will be issued indicating how many devices are enabled for monitoring, and how many are offline.
5. The VOLUME operand can only be used for volumes attached to the system since CP associates a volume label to a device number only at the time it is attached. If you want to enable a DASD device not attached to the system, use the DEVICE option.
6. For a multiple-exposure device, all exposures associated with the volume are enabled or disabled.
7. The volume to be monitored is maintained by its device address. Enabling or disabling a volume for monitoring remains valid as long as its device address does not change. For example, if a volume is detached and a volume with the same label at a different address is attached, the volume must be enabled again to be monitored and its previous device address disabled.

Specific Usage Notes for BLOCK

1. If BLOCK is not specified, the default is four 4Kb frames.
2. The minimum value for BLOCK is four 4Kb frames.
3. The maximum value for BLOCK is half the size of the PARTITION.
4. The value of BLOCK will be saved when monitoring is stopped.

Specific Usage Notes for PARTITION

1. If PARTITION is not specified, the initial value is half the size of the saved segment.
2. The minimum PARTITION size is eight 4Kb frames.
3. The maximum PARTITION size is the size of the saved segment. If sample monitoring is active, the maximum PARTITION size is the size of the saved segment minus one 4Kb frame.
4. The value of PARTITION will not be saved when monitoring is stopped.

Response 1

The command, MONITOR EVENT START, activates event-driven data collection based on the previously specified event profile if an application program is retrieving event records. The response is:

MONITOR EVENT STARTED

Response 2

MONITOR EVENT STOP deactivates all current event-driven data collection. The response when event-driven data collection is active is:

MONITOR EVENT STOPPED

The following response is issued for all other valid commands entered:

COMMAND COMPLETE

MONITOR SAMPLE

Privilege Class: A,E

Use the MONITOR SAMPLE command to establish a profile for sample data collection, including single sample data and high frequency sample data. Single sample data are a set of counters and data that characterize a system operation. High frequency sample data are a set of counters and data that represent the state of the system at the moment they are sampled.

As established by this command, a sample profile lists the data domains and their elements to be monitored, the rate at which high frequency sample data is to be collected, and the time interval at which sample data is to be reported.

This command can also be used to activate or deactivate sample data collection.

MONitor	SAMple <div style="display: inline-block; vertical-align: middle; margin-left: 10px;"> $\left[\begin{array}{l} \text{DISable} \\ \text{ENable} \end{array} \right. \left\{ \begin{array}{l} \text{I/O} \left\{ \begin{array}{l} \text{DEvIce} \left\{ \begin{array}{l} \text{rdev} \\ \text{rdev rdev...} \\ \text{rdev-rdev...} \end{array} \right\} \\ \text{Class} \quad \text{class-list} \\ \text{Type} \quad \text{type-list} \end{array} \right\} \\ \text{VOLUME} \left\{ \begin{array}{l} \text{volid-list} \\ \text{CPVOL} \end{array} \right\} \\ \text{ALL} \end{array} \right\} \\ \left\{ \begin{array}{l} \text{USER} \left\{ \begin{array}{l} \text{USERID} \quad \text{userid-list} \\ \text{ALL} \end{array} \right\} \\ \text{PROcEssor} \\ \text{STORage} \\ \text{ALL} \end{array} \right\} \end{array} \right.$ </div> <div style="margin-top: 10px;"> $\left\{ \begin{array}{l} \text{INTerval} \quad n \quad \begin{array}{l} \text{MINutes} \\ \text{SECONDS} \end{array} \\ \text{RATE} \quad \left\{ \begin{array}{l} n \quad \text{SECONDS} \\ \text{STOP} \end{array} \right\} \\ \text{STArt} \\ \text{STOP} \end{array} \right\}$ </div>
---------	--

where:

SAMPLE

indicates that the command is for single sample and high frequency sample data collection types.

DISABLE

disables the specified domains or their elements from monitoring.

Note: The MONITOR domain and the SYSTEM domain cannot be disabled from sample data collection.

ENABLE

enables the specified domains and their elements for sample data collection. Sample data shall be collected only from those domains that have been enabled. ENABLE is the default.

Note: The MONITOR domain and the SYSTEM domain are always enabled for sample data monitoring.

I/O

specifies that the I/O domain is to be enabled or disabled for sample data collection. The I/O domain is comprised of all real I/O devices; monitoring may be enabled or disabled on all devices, or it may be restricted by their device numbers, classes, types, or volumes.

I/O DEVICE rdev**I/O DEVICE rdev rdev...****I/O DEVICE rdev-rdev...**

specifies the device number of the real device to be enabled or disabled for sample data collection in the I/O domain. You may specify a list, a range of devices, or a combination of both. If you specify a range, all valid devices in the range, inclusively, will be enabled or disabled. For additional information, see the specific usage notes for I/O.

I/O CLASS class-list

specifies a list of device classes that are to be enabled or disabled for sample data collection. A class list is one or more device classes, separated by blanks. The valid classes are DASD, TAPE, SPOOL, and GRAF. All real devices of the specified class or classes will be monitored.

I/O TYPE type-list

specifies a list of device types that are to be enabled or disabled for sample data collection. A type list is one or more device types (such as the 3380), separated by blanks. For valid device types see Table 8 on page 255. All real devices of the specified type or types will be monitored.

I/O VOLUME valid-list**I/O VOLUME CPVOL**

specifies a list of volumes that are to be enabled or disabled for sample data collection. A valid list is one or more volume identifiers, separated by blanks. If CPVOL is specified, then all CP owned volumes are monitored. A CP owned volume is any volume specified on the SYSCPVOL macro of HCPSYS ASSEMBLE.

I/O ALL

specifies that all real devices are to be enabled or disabled for sample data collection.

USER

specifies that the USER domain is to be enabled or disabled for sample data collection. The USER domain is comprised of all users. Data collected for this domain consists of user resource utilization and status data for all or selected users.

USER USERID userid-list

specifies a list of users that are to be enabled or disabled for sample data collection. A userid list is one or more user identifiers, separated by blanks.

USER ALL

specifies that all users are to be enabled or disabled for sample data collection.

MONITOR SAMPLE

PROCESSOR

specifies that the PROCESSOR domain is to be enabled or disabled for sample data collection. Data related to work dispatched on each processor is collected.

STORAGE

specifies that the STORAGE domain is to be enabled or disabled for sample data collection. Status and utilization data of real, virtual, expanded, shared, and auxiliary storage are collected.

ALL

specifies that all domains and all their elements are to be enabled or disabled for sample data collection. This includes all the four domains: I/O (all devices), USER (all users), PROCESSOR, and STORAGE.

Note: SYSTEM and MONITOR domains are enabled for sample monitoring at all times.

SAMPLE INTERVAL n MINUTES

SAMPLE INTERVAL n SECONDS

specifies the time interval at which the sample data is to be reported. MINUTES is the default. For additional information, see the specific usage notes for INTERVAL.

SAMPLE RATE n SECONDS

specifies the frequency at which the high frequency sample data is to be collected. High frequency sample data is a set of counters and data which represent the state of the system at the moment they are sampled.

High frequency sampling will be automatically activated whenever the MONITOR SAMPLE START command is issued, unless the RATE value has been previously set to STOP. For additional information, see the specific usage notes for RATE.

SAMPLE RATE STOP

immediately terminates the collection of high frequency sampling data. Data for the current interval is lost.

SAMPLE START

activates sample data collection. Only those domains and their elements that have been enabled shall be monitored. At startup, system configuration records from the monitor domain are generated. These records include information concerning the memory, processor, paging and real I/O configurations, as well as the initial profile settings. High frequency sampling data collection, if enabled, as well as single sample data collection will be started.

SAMPLE STOP

deactivates creation and collection of sample data. This also stops the collection of high frequency sampling data, if it is active. All data collected for the current interval are purged. The monitor sample profile is not affected by this command.

Usage Notes

1. The following must be done before sample monitoring can be started:
 - A saved segment must be defined and saved for MONITOR.
 - An application program running in a virtual machine must load the monitor saved segment and connect to the *MONITOR CP system service via IUCV.

- Both event and sample monitoring can be active at the same time. Monitoring of both types of data can be started by use of the MONITOR START command, or for sample data, by issuing the MONITOR SAMPLE START command. Similarly, monitoring can be stopped by the appropriate stop command, MONITOR SAMPLE STOP or MONITOR STOP.

Specific Usage Notes for I/O

- Any overlapping of I/O devices can exist on a series of MONITOR SAMPLE commands on the I/O domain. For example, if this command:

```
MONITOR SAMPLE ENABLE I/O TYPE 3380
```

is followed by this command:

```
MONITOR SAMPLE ENABLE I/O CLASS DASD
```

then all DASD devices will be enabled for monitoring. However, if this is followed by this command:

```
MONITOR SAMPLE DISABLE I/O TYPE 3380
```

monitoring will cease for all 3380 devices, but will continue for all other types of DASD devices.
- A message will be issued for any devices which do not exist.
- Devices which are enabled but are found to be offline, will be monitored when varied online attached to the system.
- A summary message will be issued indicating how many devices are enabled for monitoring, and how many are offline.
- The VOLUME operand can only be used for volumes attached to the system since CP associates a volume label to a device number only at the time it is attached. If it is desired to enable a DASD device not attached to the system, it can be done by using the DEVICE option.
- For a multiple-exposure device, all exposures associated with the volume are enabled or disabled.
- The volume to be monitored is maintained by its device address. Enabling or disabling a volume for monitoring remains valid as long as its device address does not change. For example, if a volume is detached and a volume with the same label at a different address is attached, the volume must be enabled again to be monitored and its previous device address disabled.

Specific Usage Notes for INTERVAL

- The valid range of values for INTERVAL is 6 to 3600 for SECONDS, and 1 to 60 for MINUTES. INTERVAL values must be specified in whole numbers.
- If this command is not issued, an initial value of 10 minutes (600 seconds) will be used when collecting sample data.
- If an interval is currently in effect, the new interval will not take effect until the next sample data record is produced.
- The SAMPLE INTERVAL value must be greater than or equal to the SAMPLE RATE value.
- It is possible that data may not be reported at the time the interval expires. This will happen when it takes longer than an interval to collect the data. If this should happen, the interval time should be adjusted to allow sufficient time to collect and report the data.

MONITOR SAMPLE

Usage Notes for RATE

1. The valid range of value for RATE is 0.01 to 30.00 seconds. Up to two digits may be specified after the decimal point. Leading zeros are not required, and trailing zeros are not required after the decimal point.
2. If this command is not issued, an initial RATE of 5 seconds will be used.
3. If a rate is currently in effect, the new rate will not take effect until the current high frequency rate has elapsed.
4. The SAMPLE RATE value specified must be less than or equal to the SAMPLE INTERVAL value.

Response 1

MONITOR SAMPLE START causes sample data to be collected periodically based on the previously specified sample profile if an application program is retrieving sample records. The response is:

MONITOR SAMPLE STARTED

Response 2

MONITOR SAMPLE STOP deactivates all current sample data collection. The response when sample data collection is active is:

MONITOR SAMPLE STOPPED

Response 3

The following response is issued for all other valid commands entered:

COMMAND COMPLETE

MONITOR START/STOP

Privilege Class: A,E

Use the MONITOR START command to activate event and sample monitoring based on the profiles established by the preceding MONITOR EVENT and MONITOR SAMPLE commands. Use the MONITOR STOP command to deactivate all monitoring.

MONITOR	$\left\{ \begin{array}{l} \text{START [Block m][PARTition n]} \\ \text{STOP} \end{array} \right\}$
---------	---

where:

START

activates event and sample monitoring based on the profiles established by the MONITOR EVENT and MONITOR SAMPLE commands. Event monitoring will be started first.

STOP

deactivates the event and sample monitoring. The monitor event and sample profiles are not affected by this command, except for PARTITION which is set to zero. Event monitoring will be stopped first.

BLOCK *m*

specifies the number of 4Kb shared-segment frames (*m*) that will be accumulated before the virtual machine is notified that the event data is available for data collection. The BLOCK value may be modified via the MONITOR EVENT BLOCK command while event monitoring is active.

PARTITION *n*

specifies the number of 4Kb shared-segment frames (*n*) that are reserved for event data. The rest of the shared segment shall then be used for sample data.

Usage Notes

1. Specific usage notes for BLOCK:
 - a. If BLOCK is not specified, the default is four 4Kb frames.
 - b. The minimum value for BLOCK is four 4Kb frames.
 - c. The maximum value for BLOCK is half the size of the PARTITION value.
 - d. The value of BLOCK will be saved when monitoring is stopped.
2. Specific usage notes for PARTITION:
 - a. If PARTITION is not specified, the default value is half the size of the saved segment.
 - b. The minimum PARTITION size is eight 4Kb frames.
 - c. The maximum PARTITION size is the size of the saved segment. If sample monitoring is active, the maximum PARTITION size is the size of the saved segment, minus one 4Kb frame.
 - d. The value of PARTITION will not be saved when monitoring is stopped.

MONITOR START/STOP

Response 1

For MONITOR START, either one line or both lines may appear:

MONITOR EVENT STARTED

MONITOR SAMPLE STARTED

Response 2

For MONITOR STOP, either one line or both lines may appear:

MONITOR EVENT STOPPED

MONITOR SAMPLE STOPPED

MSGNOH

Privilege Class: B

Use of the MSGNOH command provides the ability for a service virtual machine to send messages to specified users without the standard header associated with the MESSAGE command.

MSGNOH	$\left. \begin{array}{c} \text{userid} \\ * \\ \text{ALL} \\ \text{OPerator} \end{array} \right\} \{ \text{msgtext} \}$
--------	---

where:

userid

*

is the identification of the user to whom the message is being sent. If you wish to send a message to your own userid, you may specify that with an asterisk (*).

ALL

transmits the message to all logged on users.

OPerator

transmits the message to the operator.

msgtext

is the text of the message that is being sent. The length of the message is limited by the number of characters remaining on the input line after the command and appropriate operand are entered.

Usage Notes

1. If a message is issued from a CMS environment, the command and the message text cannot be longer than 240 characters.
If the message is issued from a CP environment, the command and message length is limited by the input area of the terminal.
2. See "General Usage Notes for MESSAGE" on page 248.

Response

msgtext

is the response received by the specified user or users.

where:

msgtext

is the message text.

NOTREADY

NOTREADY

Privilege Class: G

Use the NOTREADY command to make a virtual spooling device appear as if it had changed from “ready” to “not ready” status.

NOTReady	vdev
----------	------

where:

vdev

is the virtual device number of the device to be made “not ready.”

Usage Notes

1. Use this command to change the status of spooled unit record devices and virtual consoles only.
2. Any I/O operation in progress to the specified device at the time the command is issued will be completed. The “not ready” condition will be in effect for the next I/O operation.

Responses

None.

ORDER**Privilege Class:** D, G

Use the ORDER command to change the order of closed spool files on a specified queue. The order of the files determines the sequence in which they will be selected for output on real printers or punches, or read on virtual readers.

The class G user can issue the ORDER command to order any spool files that the user originated or currently owns.

The class D user can issue the ORDER command to order any spool files in the system or files belonging to any user on the system.

ORDER	$ \begin{array}{l} \text{[SYSTEM]} \left\{ \begin{array}{l} \text{Reader} \\ \text{Printer} \\ \text{PUunch} \end{array} \right\} \left\{ \begin{array}{l} \text{Class c1} \\ \text{FORM form1} \end{array} \right\} \left[\begin{array}{l} \text{Class c2 ...} \\ \text{FORM form2 ...} \end{array} \right] \left. \vphantom{\begin{array}{l} \text{Reader} \\ \text{Printer} \\ \text{PUunch} \end{array}} \right\} \\ \\ \text{[userid]} \left\{ \begin{array}{l} \text{Reader} \\ \text{Printer} \\ \text{PUunch} \end{array} \right\} \left\{ \begin{array}{l} \text{Class c1} \\ \text{FORM form1} \\ \text{spoolid1} \end{array} \right\} \left[\begin{array}{l} \text{Class c2 ...} \\ \text{FORM form2 ...} \\ \text{spoolid2 ...} \end{array} \right] \left. \vphantom{\begin{array}{l} \text{Reader} \\ \text{Printer} \\ \text{PUunch} \end{array}} \right\} \\ \\ \text{[*]} \left\{ \begin{array}{l} \text{Reader} \\ \text{Printer} \\ \text{PUunch} \end{array} \right\} \left\{ \begin{array}{l} \text{Class c1} \\ \text{FORM form1} \\ \text{spoolid1} \end{array} \right\} \left[\begin{array}{l} \text{Class c2 ...} \\ \text{FORM form2 ...} \\ \text{spoolid2 ...} \end{array} \right] \left. \vphantom{\begin{array}{l} \text{Reader} \\ \text{Printer} \\ \text{PUunch} \end{array}} \right\} \end{array} $
-------	---

Note: You may enter a combination of CLASS, FORM, and spoolid on the same command line.

where:

SYSTEM

is restricted to class D users. Use the SYSTEM operand to order any file or group of files on the system queue that you specify without indicating the owner's userid. This is not valid with the "spoolid" option.

userid

*

is the user identification of the user whose spool files are to be ordered. If you are ordering your own files, you may specify userid as an asterisk (*).

The userid option allows a class D user to order any file, or group of files, owned by a particular user. The files are repositioned only with respect to other spool files owned by the userid that you specified.

The userid option allows a class G user to order any file that he or she originated, and that is now owned by the user whose userid is specified on the command line. The files will be ordered only in relation to any other files that the class G user originated which are owned by the userid specified in the command line.

READER**RDR****PRINTER****PRT****PUNCH****PCH**

indicates the spool file queue that you want to order.

ORDER

CLASS c1

CLASS c2 ...

are the file classes that you want to rearrange, and the sequence in which they are to be processed. You must enter the keyword CLASS for each class that you want to order. c1, c2, ... are one character alphanumeric fields (from A to Z or from 0 to 9) that represent spooling classes.

FORM form1

FORM form2 ...

are the forms to be rearranged, and the sequences in which they are to be processed. 'form1' and 'form2' are specified as 1-to-8 characters. For the class G user, the form is the user form name. For the class D user, the form is the user form name when you specify "*" or default to your userid. Otherwise, the form is the operator form number.

spoolid1

spoolid2 ...

are the spool identifications of the files that you want to order. This is not valid with the SYSTEM keyword.

Usage Notes

1. Class G users can order only their own files and any files that they originated.

The order that you specify will not affect the position of any other files that are in the same system queue as your files, but will only reposition your files with respect to any other files that you may own.

For example, suppose the system printer queue contains the following files:

Spoolid	Owning user
1	USER1
22	USER2
333	USER2
4444	USER2

Assume further that USER2 issues the following ORDER command:

```
ORDER PRT 4444
```

The following queue order will result:

Spoolid	Owning user
1	USER1
4444	USER2
22	USER2
333	USER2

The class G user may order files that he or she originated and that another userid now owns. If so, the user will only change the order of the files in relation to any other files that he or she originated and are now owned by the other userid. For example, assume there are two users, USER1 and USER2, and USER2's printer queue is as follows:

Spoolid	Owning User	Originator
123	USER2	USER2
456	USER2	USER1
789	USER2	USER1
9101	USER2	USER2

Assume further that USER1 issues the following ORDER command:

```
ORDER USER2 PRT 789
```

The following print queue order will result for USER2:

Spoolid	Owning User	Originator
123	USER2	USER2
789	USER2	USER1
456	USER2	USER1
9101	USER2	USER2

- The class D user can reorder any files in the system by specifying the SYSTEM or userid option.

For example, if you issue the following command:

```
ORDER SYSTEM PRINTER CLASS C
```

all class C printer files are placed ahead of any other printer files in the system, regardless of who owns them or originated them.

The ORDER command is also useful to the class D user who wants to place files owned by a particular user ahead of all other files on a specified queue.

For example, if you issued the following command:

```
ORDER USER2 PUNCH CLASS B
```

you place all of USER2's class B punch files ahead of all other punch files on the system.

- Use the QUERY READER/PRINTER/PUNCH command to determine spool file attributes, such as class and spoolid, for the files that you want to rearrange. The display produced by the command will show the current order of the spool files on the queue that you specify.
- You can use a combination of class, forms, and spoolid specifications to rearrange files. For example:

```
ORDER PRINTER CLASS A 1963 CLASS C FORM LISTING
```

orders the command issuer's printer queue files in the following order: all class A files, the file with spoolid 1963, all class C files, and all files with the form "LISTING." They will be followed by any other printer spool files not specified in the ORDER command line.

- A file cannot be ordered more than once in a given command. For example, assume the following is a user's spool print queue.

Spoolid	Class
1234	A
4567	B
7890	A
5555	A
8888	B

ORDER

If the user issued "ORDER PRT 8888 CLASS A CLASS B," the following order would result:

Spoolid	Class
8888	B
1234	A
7890	A
5555	A
4567	B

Note that the spoolid 8888 was only ordered once.

Responses

```
{nnnnnn} FILE(S) ORDERED  
{ NO}
```

indicates the number of files ordered.

Note: This response is suppressed if you have issued the CP SET IMSG OFF command.

Migration Notes

VM/SP HPO

1. VM/XA SP does not allow users to specify SYSTEM and spoolid on the same command line. VM/SP HPO (Release 5) users do not see this difference.
2. VM/XA SP does not support the DEST option.
3. VM/XA SP allows room for a 7-digit response.

VM/XA SF: VM/XA SP does not allow users to specify SYSTEM and spoolid on the same command line.

PER**Privilege Class: G**

Within VM/XA SP, the TRACE command is used to monitor events that occur in your virtual machine. Please refer to the TRACE command in this reference book for this information.

Migration Notes*VM/SP HPO*

1. VM/XA SP will accept the PER command as a synonym for TRACE.
2. VM/XA SP provides equivalent function differently for the following:
 - SAVE – provided by GOTO and APPEND
 - GET – provided by GOTO and APPEND
 - END CURRENT – provided by applying CLEAR to the active trace set
 - END ident – provided by DELETE
 - END set – provided by CLEAR
 - END traptype – same result by deleting each trap separately
 - GUEST R – provided by NODAT
 - GUEST V – provided by DAT.
3. VM/XA SP has no equivalent for MASK without DATA.

PURGE

PURGE

Use the PURGE command described in this section to remove closed spool files and system data files from the system.

Complete Format for PURGE

The format box below shows all the operands available with the PURGE command. Complete descriptions of the operands appear on the following pages, in the same sequence as the operands appear here.

		Class
PURge	[SYSTEM] { Reader Printer PUnch UR } { ALL Class c1 [Class c2 ...] FORM form1 [FORM form2 ...] }	D
	[userid _*] { Reader Printer PUnch UR } { ALL Class c1 [Class c2 ...] spoolid1 spoolid2 ... FORM form1 [FORM form2 ...] }	D,G
	IMG { ALL spoolid1 [spoolid2 ...] NAME fn1 [NAME fn2 ...] }	A,B,C,D,E
	NLS { ALL spoolid1 [spoolid2 ...] NAME fn1 [NAME fn2 ...] }	B
	NSS [ASSOCiates] { ALL spoolid1 [spoolid2 ...] NAME fn1 [NAME fn2 ...] }	E
	[SYSTEM] [TRFiles { ALL NAME fn1 [NAME fn2 ...] }	A,C,D,E,G
	[userid _*] [TRFiles { ALL NAME fn1 [NAME fn2 ...] spoolid1 [spoolid2 ...] }	
	UCR { ALL spoolid1 [spoolid2 ...] NAME fn1 [NAME fn2 ...] }	A,B,C

General Usage Notes for PURGE

- If you specify more than one file selection criteria, processing of the PURGE command stops upon encountering an invalid option. None of the remaining options are processed.

Migration Notes*VM/SP HPO*

1. VM/XA SP has no default for selection criteria, unlike HPO which uses a default of ALL.
2. VM/XA SP uses UR to specify all device types instead of ALL in VM/SP HPO.
3. VM/XA SP does not allow you to specify SYSTEM and spoolid on the command line. VM/SP HPO (Release 5) users will not see this difference.
4. VM/XA SP does not support the DEST and FORCE options.
5. VM/XA SP allows room for a 7-digit response.

VM/XA SF: VM/XA SP does not allow you to specify SYSTEM and spoolid on the command line.

PURGE

Privilege Class: D,G

Use the PURGE command to remove closed spool files from the system before they are selected for output on a real printer or punch, or before they are read by a virtual card reader. The PURGE by spoolid command can be used to purge a file after it has begun printing or punching on a real device; this causes printing or punching of the file to cease, and causes the file to be purged.

Class G users can purge any spool files that they own or originate.

The class D user can purge any spool file in the system, or any files that belong to a particular user on the system.

PURge	<pre>[SYSTEM] { Reader } { ALL { Printer } { Class c1 [Class c2 ...] { PUnch } { FORM form1 [FORM form2 ...] { UR }</pre>
	<pre>[userid] { Reader } { ALL { * } { Printer } { Class c1 [Class c2 ...] { _ } { PUnch } { spoolid1 [spoolid2 ... { } { UR } { FORM form1 [FORM form2 ...] }</pre>

Note: You may enter a combination of classes, spoolids, and forms on the command line.

where:

SYSTEM

allows the class D user to manipulate all files in the system, regardless of ownership. This is not allowed with *spoolid*.

userid

*

is the user identification for the user whose files are to be purged.

The userid option allows the class D user to purge files that belong to any user on the system. If you are a class G user, you can purge your own files or files owned by another userid that you originated.

Any user, class D or class G, can purge their own files using the asterisk (*) option. If you do not specify userid or *, the default is to purge your own files.

READER

RDR

PRINTER

PRT

PUNCH

PCH

UR

indicates which spool file queue contains the files that you want to purge. UR indicates that all reader, printer, and punch files are to be purged.

ALL

specifies that ALL files on the queue of the device type specified (READER, PRINTER, PUNCH, UR) are to be purged.

CLASS c1 CLASS c2

designates a spool file class where *c1* and *c2* are one-character alphanumeric fields from A to Z or 0 to 9. All the files of the specified class or classes for the specified device type or types will be purged.

spoolid1

purges the files of the specified spool identification, for the specified device type. This is not valid with the SYSTEM keyword.

FORM form1 FORM form2

purges all files with the specified form names or form numbers. *form1* and *form2* are specified as 1- to 8-characters. For the class G user, the form is the user form name. For the class D user, the form is the user form name when you specify * or default to your userid. Otherwise, the form is the operator form number.

Usage Notes

1. The class D user can use the PURGE command to purge any file in the system by specifying the SYSTEM or userid option. For example, if you issue the following command:

```
PURGE SYSTEM PRINTER CLASS C
```

you purge all class C printer files in the system, regardless of who owns them or who originated them.

The class D user can use the userid option to purge any file or files that are owned by a particular user. For example, if you issue the following command:

```
PURGE GENUSER READER ALL
```

you purge all reader files owned by the userid GENUSER.

2. You can purge files using a combination of CLASS and spoolid on the same command line. For example:

```
PURGE PRINTER CLASS A 1932 FORM DOCUMENT CLASS D 619
```

specifies that CP should purge all the command user's class A and class D printer files, all the command user's files with form DOCUMENT, and the command user's printer files having spoolids 1932 and 619.

3. If a command syntax error occurs when purging closed spool files from the system, only the files encountered prior to the error are processed.
4. An active spool file (one which is being printed or punched) may only be purged by specifying its spoolid. Printing or punching of the file will stop, and the file will be purged.

Responses

```
{ nnnnnn } FILE(S) PURGED
{      NO }
```

indicates the number of files purged.

Note: This response is suppressed if you have issued the CP SET IMSG OFF command.

PURGE

Migration Notes

VM/SP HPO

1. VM/XA SP has no default for selection criteria, unlike HPO which uses a default of ALL.
2. VM/XA SP uses UR to specify all device types instead of ALL in VM/SP HPO.
3. VM/XA SP does not allow you to specify SYSTEM and spoolid on the command line. VM/SP HPO (Release 5) users will not see this difference.

VM/XA SF: VM/XA SP does not allow you to specify SYSTEM and spoolid on the command line.

PURGE IMG

Privilege Class: A,B,C,D,E

Use the PURGE IMG command to remove system data files that contain image libraries.

PURGe	IMG { ALL spoolid1 [spoolid2 ...] NAME fn1 [NAME fn2 ...] }
-------	---

Note: You may enter a combination of spool identification numbers and names on the command line.

where:

IMG

specifies that image library files should be purged.

ALL

allows you to purge all image library files.

spoolid [spoolid2 ...]

is the number used to identify the file. It specifies the file to be purged.

NAME fn [NAME fn2 ...]

specifies the filename that identifies the files to be purged.

Usage Notes

1. In response to the PURGE command, CP purges the designated files *unless* the files are currently in use. If a file is currently in use, CP places it in a “pending purge” state (class P), and purges it as soon as it is no longer being used or the system is shut down or re-IPLed.
2. To determine if a file is in a pending purge state, issue the QUERY IMG command. If the file is in a pending purge state, CP’s response will show that the file is class “P.”
3. Any combination of spoolid and NAME may be specified on the PURGE IMG command line. For example:

```
PURGE IMG 1234 NAME IMAG3800 3421
```

will purge IMG files 1234 and 3421, and any IMG files with the filename “IMAG3800.”

Response 1

The response from the PURGE IMG command displays the count of files purged.

```
{ nnnn } FILE(S) PURGED
{ NO }
```

where:

nnnn

indicates the number of files purged.

Note: This response is suppressed if you have issued the CP SET IMSG OFF command.

PURGE IMG

Response 2

This response displays the count of files pending a purge and is issued only when pending purge files result.

```
{nnnn} FILE(S) PENDING PURGE
```

where:

nnnn

indicates the number of the files which are purged when they are no longer in use.

For example, in the situation where five image files exist, of which two are in use, issue the command:

```
PURGE IMG ALL
```

The responses will be:

```
0003 FILES PURGED  
0002 FILES PENDING PURGE
```

Migration Notes

VM/SP HPO: This function is not provided in *VM/SP HPO*.

PURGE NLS

Privilege Class: B

Use the PURGE NLS command to purge message repository files.

PURGe	NLS { ALL spoolid1 [spoolid2 ...] NAME fn1 [NAME fn2 ...] }
-------	---

where:

NLS

specifies that message repository files should be purged.

ALL

specifies that all message repository files should be purged.

spoolid

specifies the spool identification number of the file to be purged.

NAME fn

specifies the name of the file to be purged.

Usage Notes

Any combination of spoolid and NAME may be specified on the PURGE NLS command line. For example:

```
PURGE NLS 1234 NAME NLS101 3421
```

purges message repository files 1234 and 3421, and any message repository files with the filename NLS101.

Response 1

The response from the PURGE NLS command displays the count of files purged.

```
{nnnn} FILE(S) PURGED  
{ NO }
```

where:

nnnn

indicates the number of files purged.

Note: This response is suppressed if you have issued the CP SET IMSG OFF command.

Response 2

This response displays the count of files pending a purge and is issued only when pending purge files result.

```
{nnnn} FILE(S) PENDING PURGE
```

where:

nnnn

indicates the number of files that are purged when they are no longer in use.

PURGE NLS

| For example, say that out of five existing NLS files two are in use, and you issue the
| command:

| PURGE NLS ALL

| The responses will be:

| 0003 FILES PURGED

| 0002 FILES PENDING PURGE

PURGE NSS

Privilege Class: E

Use the PURGE NSS command to remove system data files that contain named saved systems (NSS) and saved segments.

PURge	NSS [ASSOCIates] { ALL spoolid1 [spoolid2 ...] NAME fn1 [NAME fn2 ...] }
-------	--

Note: You may enter a combination of spool identification numbers and names on the command line.

where:

NSS

specifies that named saved systems (NSS) and saved segments should be purged.

ASSOCIates

specifies that the files associated with a segment space or member segment will be purged. If you purge a segment space, its associated members are also purged or set to pending purge or their segment space entries are compressed. If you purge a member segment, it is purged or set to pending purge and its associated segment spaces are purged or set to pending purge or their member entries are compressed.

ALL

allows you to purge all NSS and saved segment files.

spoolid

is the number used to identify the file. It specifies the file to be purged. It can be an NSS, a DCSS, a segment space, or a member saved segment.

NAME fn

specifies the filename that identifies the file to be purged. The file can be an NSS, a DCSS, a segment space, or a member saved segment.

Usage Notes

1. In response to the PURGE command, CP purges the designated files *unless* the files are currently in use. If a file is currently in use, CP places it in a “pending purge” state (class P), and purges it as soon as it is no longer being used or the system is shut down or re-IPLed. If the file being purged is a member of a segment space, the system places it in pending purge status if any user is accessing any member of the segment space. The system purges the member when the segment space is no longer in use.
2. If you do not specify ASSOCIATES and the target of the PURGE command is a member saved segment, CP purges only that member. It does not purge any other file and it does not update any segment space descriptions. If the target of the PURGE command is a DCSS, CP purges only the DCSS. If the target of the PURGE command is a segment space, CP purges only that segment space. It does not purge any other file and it does not update members of that segment space.

PURGE NSS

3. To determine if a file is in a pending purge state, issue the QUERY NSS command. If the file is in a pending purge state, CP's response will show that the file is class "P."
4. Any combination of spoolid and NAME may be specified on the PURGE NSS command line. For example:

```
PURGE NSS 1234 NAME CMS 3421
```

will purge NSS and saved segment files 1234 and 3421, and any NSS and saved segment files with the filename "CMS."

Response 1

The response from the PURGE NSS command displays the count of files purged.

```
{nnnn} FILE(S) PURGED  
{ NO }
```

where:

nnnn
indicates the number of files purged.

Note: This response is suppressed if you have issued the CP SET IMSG OFF command.

Response 2

This response displays the count of files pending a purge and is issued only when pending purge files result.

```
{nnnn} FILE(S) PENDING PURGE
```

where:

nnnn
indicates the number of files which are purged when they are no longer in use.

For example, in the situation where five NSS or saved segment files exist, of which two are in use, issue the command:

```
PURGE NSS ALL
```

The responses will be:

```
0003 FILES PURGED  
0002 FILES PENDING PURGE
```

Migration Notes

VM/SP HPO: This function is not provided in VM/SP HPO.

PURGE TRFILES

Privilege Class: A,C,D,E,G

Use the PURGE TRFILES command to remove users' system trace files from the system. System trace files are system data files created by the TRSAVE command for CP or virtual machine tracing.

PURge	[SYSTEM] [TRFiles { ALL NAME fn1 [NAME fn2 ...] }]
	[userid] [TRFiles { ALL NAME fn1 [NAME fn2 ...] spoolid1 [spoolid2 ...] }]

where:

SYSTEM

specifies that files should be purged regardless of who owns or originated them. Class G users cannot use the SYSTEM option. It is invalid with the *spoolid* option.

userid

*

specifies the owner of the files to be purged. System trace files cannot be purged while they are in use. Class G users may purge only files that they own or that they originate, and that are not in use. Class A, C, D, and E users may purge any system trace files that are not in use.

You can specify * to purge your own system trace files.

TRFiles

specifies that system trace files should be purged.

ALL

specifies that all system trace files should be purged.

spoolid

specifies the spool file identification number of the file to be purged. This option is invalid with the SYSTEM option.

NAME fn

specifies the name of the file to be purged. This is the file name specified on the command TRSAVE... ON DASD NAME *fname*... when the trace was started. More than one file can exist with the same filename for a given owner.

Usage Notes

1. Any combination of spoolid and NAME may be specified on the PURGE TRFILES command line. For example:
PURGE TRF 1234 NAME GTRACE 3421
purges system trace files 1234 and 3421, and any trace files with the filename GTRACE.
2. If you specify *spoolid* to purge files, the file you purge may be part of a group of files with the same file name as that created by the TRSAVE..ON DASD NAME *fname*.. command. If this is the case, some system trace data information is purged.

PURGE TRFILES

3. If you specify the NAME option to purge files and there are multiple groups of trace files with the same name, you may accidentally purge files you do not want to purge. Use the QUERY TRFILES command to display all of the trace files and determine which files you want to discard. If more than one file exists with the same name, use *spoolid* to specify the correct file to be purged and to avoid losing data.
4. If you are authorized to use the SYSTEM option and you use it with the NAME option, you purge all trace files in the system with the filename specified.
5. If a system trace file has a class of W (meaning the file is being written to), it is not purged. You must issue the TRSAVE OFF, TRSAVE CANCEL, or TRSOURCE DISABLE command to purge the class W file.

Responses

```
{nnnnnn} FILE(S) PURGED  
{      NO }
```

indicates the number of files purged.

Note: This response is suppressed if you have issued the CP SET IMSG OFF command.

PURGE UCR

Privilege Class: A,B,C

Use the PURGE UCR command to purge a UCR file created by the system programmer.

PURge	UCR	{ ALL NAME fn1 [NAME fn2 ...] spoolid1 [spoolid2 ...] }
-------	-----	---

UCR

specifies that UCR files should be purged.

ALL

allows you to purge all UCR files.

spoolid

is the numbers used to identify the file to be purged.

NAME fn

specifies the filenames of the file to be purged.

Usage Notes

You may specify any combination of spoolid and NAME on the PURGE UCR command line. For example:

```
PURGE UCR 1234 NAME UCR101 3421
```

will purge UCR files 1234 and 3421, and any UCR files with the filename UCR101.

Responses

```
{ nnnn } FILE(S) PURGED  
{ NO }
```

displays the count of files purged by the PURGE UCR command.

nnnn

indicates the number of files purged.

Note: This response is suppressed if you have issued the CP SET IMSG OFF command.

Migration Notes

VM/SP HPO: This function is not provided in VM/SP HPO.

QUERY

Use the QUERY command described in this section to determine your system status and machine configuration.

The format box below shows all of the operands available with the QUERY command. Complete descriptions of the operands appear on the following pages, in the same sequence as the operands appear here.

		Class
Query	<p>ABEND</p> <p>ACCount</p> <p>ALLOC [<u>ALL</u> [volid1 [volid2 ...]]]</p> <p>CACHE { rdev rdev ... rdev-rdev }</p> <p>CHPID yy</p> <p>CMDLIMit [XAUTOLog AUTOLog]</p> <p>COMMAnds [cmd1 cmd2 cmd3 ...]</p> <p>CPLANGLIST</p> <p>CPLANGuage</p> <p>CPLEVEL</p> <p>CPTRace [{FOR <u>NONSPECIFIC</u> } FOR { SPECIFIC SYSTEM ALL userid }] [[<u>ALLcodes</u> code1 code2 ...] ...]</p> <p>CPUid</p> <p>DASd RESERVE [userid]</p>	<p>B</p> <p>G</p> <p>D</p>

Query	<pre> { type [ACTIVE FREe OFFline ATTach [userid] SYStem ALL] rdev [STATUS] rdev rdev-rdev [rdev rdev ... rdev-rdev rdev-rdev ...] DAsd RESERVE [userid] [DAsd] valid } DUMP Files [SYSTEM] [[Class c][FORM form]] [HOld [NOHOld SYShold USERhold] [XFER{userid}][AVAIL] {ALL}] [userid] [[Class c][FORM form]] [HOld * [spoolid [ALL] [NOHOld SYShold USERhold] [XFER{userid}][AVAIL] {ALL}] FRames Hold [LOG] IMG [spoolid] [COUNT] [NAME fn [Class c [ALL]] </pre>	<p>B</p> <p>B</p> <p>D</p> <p>D,G</p> <p>A,B,E</p> <p>B,D</p> <p>A,B,C, D,E</p>
-------	--	---

QUERY

Query	IOAssist	B
	Links vdev	G
	LOGICAL { ldev [ldev ...] [ldev-ldev [ldev-ldev ...]] }	B
	LOGmsg	Any
	MAXSPOOL [SYSTEM userid * _]	D,G
	MAXUsers	A
	MITime	A,B
	MONData	C
	MONitor [ALL Block Event INTerval PARTition RATE SAMPLe]	A,E
	Names	Any
	NLS [ALL NAME fn spoolid Class c] [COUNT]	E
	NSS [spoolid] [MAP] NAME fn [COUNT] Class c ALL	E
	[USERS filename]	

Query	PASSWORD { LINK LOGON XAUTOLOG AUTOLOG }	G
	PATHS [TO] rdev	A,B
	PF[nn]	G
	PROcessors	A,B,C,E
	QUICKDSP userid	A,E
	Reader [SYSTEM][[Class c][FORM form]] [AVAIL] [HOLD NOHOLD SYShold USERhold] [[EXP] [XFER {userid}]]	D
	Printer	
	PUnch	
	Reader [userid][[Class c][FORM form]] [AVAIL] [HOLD NOHOLD SYShold USERhold] [[EXP] [XFER {userid}]]	D,G
	Printer [*] [spoolid]	
	PUnch	
	RECORDing	A,B,C,E,F
	SCReen	G
	SDF	A,B,C,D E,G
	SECuser	G
	Set	G
	SHARE userid	A,E

QUERY

Query	SRM { <u>ALL</u> IABias DSPSlice LDUBuf STORBuf DSPBuf } STORage SYSTEM [rdev] [TDisk] [<u>ALL</u>] [TDsk] TAG { [DEV] { Printer PUnch CONsole vdev } } { FILE spoolid } TDisk TDsk TERMinal Time TRace [<u>ACTive</u> ALL name SETS RETURNS] TRfiles [SYSTEM] [<u>ALL</u> [NAME fn]] [COUNT] [userid] [<u>ALL</u> [NAME fn]] [COUNT] [*] [spoolid]]	A,E B B G B G G G A,C,D,E,G
-------	--	---

Query	<pre> TRSAVE [FOR] [CP [ID] traceid ALL] TRSource [ALL ID traceid SET traceset TYPE (IO) (GT) (DATA) (NONE) USER } userid *] UCR [ALL NAME fn spoolid] [COUNT] { USERID } { userid } Users [userid] VECTor [users] Virtual [ALL] [Virtual] CONsole [Virtual] CPUS [Virtual] CTCa [Virtual] DAsd [DETAILS] [Virtual] { vdev { vdev-vdev } } [DETAILS] [Virtual] {GRaf} </pre>	<p>A,C,E,G</p> <p>A,C,E,G</p> <p>A,B,C</p> <p>G</p> <p>Any</p> <p>A,B,C,E</p> <p>G</p> <p>G</p> <p>G</p> <p>G</p> <p>G</p> <p>G</p> <p>G</p> <p>G</p>
-------	---	---

QUERY

Query	Virtual {Printer} Prt }	G
	Virtual {PUnch} PCh }	G
	Virtual {Reader} Rdr }	G
	[Virtual] STORage	G
	[Virtual] TApes	G
	[Virtual] UR	G
	[Virtual] VECtor	G
	[Virtual] {XSTore } {XSTorage }	G
	V=R	B
	{ XSTore } [USER [userid]] { XSTorage } [MAP]	B

QUERY ABEND

Privilege Class: B

Use QUERY ABEND to display the current status of the SOFT ABEND function. Using QUERY ABEND, you can determine whether soft abends will cause full system termination, or whether soft abends will be processed normally.

Query	ABEND
-------	-------

where:

ABEND

requests the display of the current status of the SOFT ABEND function.

Usage Notes

None.

Responses

ABEND { SOFT
HARD
HARD aaannn }

where:

SOFT

indicates that soft abends will not force full system termination; normal system processing will continue after recovery action has been taken.

HARD

indicates that no soft abends are to attempt recovery action; all soft abends will cause full system termination as if they were hard abends.

HARD aaannn

specifies the soft abends that you have selected to be treated as hard abends; they will cause full system termination. The code aaannn is made up of a 3 character module name and a 3 character abend code.

Note: ABEND HARD aaannn can display up to five entries on a single output response line.

QUERY ACCOUNT

QUERY ACCOUNT

Privilege Class: G

Use QUERY ACCOUNT to display the account number currently being used for your session charges.

Query	ACCount
-------	---------

where:

ACCOUNT

requests the display of your current session account number.

Usage Notes

None.

Responses

ACCOUNT acctnum

where:

acctnum

is your current session account number.

QUERY ALLOC

Privilege Class: D

Use QUERY ALLOC to display the number of allocated, in use, and available cylinders for DASD volumes attached to the system.

Query	ALLOC [ALL valid1 [valid2 ...]]
-------	--------------------------------------

where:

ALLOC

requests the display of the number of allocated, in-use, and available cylinders.

ALL

specifies that you want allocation information for all system volumes. This is the default.

valid1 [valid2 ...]

specifies a list of one or more specific volume identifier or identifiers for which you want information displayed.

Usage Notes

Examples of QUERY ALLOC:

QUERY ALLOC - displays information about all system volumes

QUERY ALLOC ALL - displays information about all system volumes

QUERY ALLOC VOL001 - displays information about a single system volume whose valid is VOL001

QUERY ALLOC VOL001 DASD99 - displays information about two system volumes whose valids are VOL001 and DASD99.

Responses

```
DASD rdev valid type TDISK TOTAL=nnnn INUSE=nnnn AVAIL=nnnn
      PAGE TOTAL=nnnn INUSE=nnnn AVAIL=nnnn
      SPOOL TOTAL=nnnn INUSE=nnnn AVAIL=nnnn
      DRCT TOTAL=nnnn INUSE=nnnn AVAIL=nnnn <,ACTIVE>
```

is displayed for each system volume.

where:

rdev

is the real device number of the CP-owned volume.

valid

is the volume identification of the CP-owned volume.

QUERY ALLOC

type

shows the DASD device type (for example, 3380).

nnnn

shows the number of cylinders allocated in each category.

<,ACTIVE>

appears on the DRCT response line if that volume contains the currently active user directory.

Migration Notes

VM/SP HPO: This function is not provided by HPO.

VM/XA SF: The **<,ACTIVE>** option which appears with the volume containing the active user directory is new.

QUERY CACHE

Privilege Class: B

Use the QUERY CACHE command to display caching status for 3880 Model 11, 13, 21 and 23 subsystems.

Query	CACHE	{	rdev	}	
			rdev ...		
			rdev-rdev		

where:

rdev

is the real device number of one of the devices in the subsystem.

rdev ...

is a list of real device numbers.

rdev-rdev

is a range of real device numbers.

Usage Notes

For a multiple-exposure device (3880 models 11 and 21), only the base address can be queried. For example, to display the caching status for the cache associated with device 340, issue:

```
QUERY CACHE 340
```

Response 1

```
rdev SUBSYSTEM STORAGE
cccccccK BYTES CONFIGURED
aaaaaaaaK BYTES AVAILABLE
ooooooooK BYTES OFFLINE
ppppppppK BYTES UNAVAILABLE
```

is the response returned for each specified real device attached to a 3880 Model 11 subsystem.

where:

rdev

is the real device number of one of the devices in the subsystem.

ccccccc

is the configured size of subsystem storage in kilobytes.

aaaaaaaa

is the size in kilobytes of subsystem storage available to the storage director for allocation.

oooooooo

is the size in kilobytes of subsystem storage that is not available to the storage director for allocation. The storage space is not available because of storage failures that occurred during attempts to transfer data from subsystem storage.

QUERY CACHE

pppppppp

is the size in kilobytes of subsystem storage that is not available to the storage director for allocation due to DASD exception conditions that prevent successful completion of a transfer from subsystem storage to DASD.

Note: cccccccc, aaaaaaaaa, ooooooooo, and pppppppp are decimal numbers.

Response 2

```
rdev CACHE {AVAILABLE } FOR SUBSYSTEM
            {UNAVAILABLE}
cccccccK BYTES CONFIGURED
aaaaaaaK BYTES AVAILABLE
oooooooK BYTES OFFLINE
pppppppK BYTES UNAVAILABLE
```

is the response returned for each real device attached to a 3880 Model 21 subsystem.

where:

rdev

is the real device number of one of the devices in the subsystem.

ccccccc

is the configured size of subsystem storage in kilobytes.

aaaaaaa

is the size in kilobytes of subsystem storage available to the storage director for allocation.

ooooooo

is the size in kilobytes of subsystem storage that is not available to the storage director for allocation. The storage space is not available because of storage failures that occurred during attempts to transfer data from subsystem storage.

ppppppp

is the size in kilobytes of subsystem storage that is not available to the storage director for allocation due to DASD exception conditions that prevent successful completion of a transfer from subsystem storage to DASD.

Note: cccccccc, aaaaaaaaa, ooooooooo, and pppppppp are decimal numbers.

Response 3

```
rdev CACHE {AVAILABLE } FOR SUBSYSTEM
            {UNAVAILABLE}
cccccccK BYTES CONFIGURED
aaaaaaaK BYTES AVAILABLE
oooooooK BYTES OFFLINE
pppppppK BYTES BOUND
```

is the response returned for each real device attached to a 3880 Model 13 subsystem.

rdev CACHE { ACTIVATED } FOR DEVICE
 { DEACTIVATED }

is the response to be repeated for each device specified in the QUERY CACHE command.

where:

rdev

is the real device number of one of the devices in the subsystem.

ccccccc

is the configured size of subsystem storage in kilobytes.

aaaaaaaa

is the size in kilobytes of subsystem storage available to the storage director for allocation.

oooooooo

is the size in kilobytes of subsystem storage that is not available to the storage director for allocation. The storage space is not available because of storage failures that occurred during attempts to transfer data from subsystem storage.

pppppppp

is the size in kilobytes of subsystem storage currently occupied by tracks that are bound to the cache within one or more high-performance storage areas.

Note: ccccccc, aaaaaaaaa, ooooooooo, and pppppppp are decimal numbers.

Response 4

rdev CACHE { AVAILABLE } FOR SUBSYSTEM
 { UNAVAILABLE }
 cccccccK BYTES CONFIGURED
 aaaaaaaK BYTES AVAILABLE
 oooooooK BYTES OFFLINE

is the response returned for each real device attached to a 3880 Model 23 subsystem.

rdev CACHE { ACTIVATED } FOR DEVICE
 { DEACTIVATED }

is the response to be repeated for each device specified in the QUERY CACHE command.

where:

rdev

is the real device number of one of the devices in the subsystem.

ccccccc

is the configured size of subsystem storage in kilobytes.

aaaaaaaa

is the size in kilobytes of subsystem storage available to the storage director for allocation.

QUERY CACHE

0000000

is the size in kilobytes of subsystem storage that is not available to the storage director for allocation. The storage space is not available because of storage failures that occurred during attempts to transfer data from subsystem storage.

Note:

ccccccc, aaaaaaa, and ooooooo are decimal numbers.

QUERY CHPID

Privilege Class: A, B

Use the QUERY CHPID command to display all the devices which have channel path yy installed to them. It also displays the current status of the path as it pertains to that device (online or offline).

Query	CHPID yy
-------	----------

where:

yy
is the path whose device allocation information is requested.

Examples

To display all the devices on path 1A type:

```
QUERY CHPID 1A
```

A typical response may be:

```
PATH 1A ONLINE TO DEVICES 0190, 0191, 0193
PATH 1A OFFLINE TO DEVICES 0192
```

Usage Notes

The QUERY CHPID command shows all devices associated with a specific channel path. For information about which channel paths are associated with a specific device, see the QUERY PATHS command.

Response 1

Path Status:

```
PATH yy {ONLINE } TO DEVICES xxxx(1) .... xxxx(n)
PATH yy {OFFLINE} TO DEVICES zzzz(1) .... zzzz(n)
```

indicates that path yy to device xxxx(1) through device xxxx(n) is logically available (online) and path yy to device zzzz(1) through device zzzz(n) is logically unavailable (offline).

where:

yy
is the logical path.

xxxx

zzzz

are the real device numbers.

QUERY CHPID

Response 2

No Devices on Path

NO DEVICES ON PATH yy

indicates there are no devices on the specified channel path.

where:

yy
is the path.

Response 3

Nonexistent CHPID

INVALID CHPID yy

indicates that a nonexisting channel path ID was specified.

where:

yy
is the nonexisting channel path ID that was entered.

Response 4

Path Status:

PATH STATUS FOR DEVICE rdev: ONLINE - y1,yn OFFLINE - x1,xn

indicates the online or offline status for each channel path ID to the specified device.

where:

rdev
is the real device number.

y1,yn
are the logically online channel paths to the device.

x1,xn
are the logically offline channel paths to the device.

QUERY CMDLIMIT

Privilege Class: A, B

Use QUERY CMDLIMIT to determine the system limit value that disables the AUTOLOG and XAUTOLOG commands. If you enter more incorrect passwords than this system limit value, the AUTOLOG and the XAUTOLOG commands are disabled for the remainder of your logon session.

Query	CMDLIMIT [XAUTOLOG] AUTOLOG]
-------	------------------------------------

where:

AUTOLOG

XAUTOLOG

requests a display showing the system limit value for the AUTOLOG and XAUTOLOG commands. This parameter is optional.

Usage Notes

1. The SET CMDLIMIT command is used to change the system limit value for the number of incorrect passwords for the AUTOLOG and XAUTOLOG commands.
2. For security reasons, CP limits the number of incorrect passwords allowed in your logon session. Each invalid password entered for an AUTOLOG or XAUTOLOG command is counted. Once you have exceeded the limit, you must log off and log on again to enter more AUTOLOG or XAUTOLOG commands.
3. AUTOLOG is an accepted synonym for XAUTOLOG. The minimum abbreviation is AUTOL.

Responses

XAUTOLOG/AUTOLOG incorrect password limit is nnn.

where:

nnn

indicates the number of incorrect passwords that can be entered before the AUTOLOG and XAUTOLOG commands are disabled for the rest of your logon session.

QUERY COMMANDS

Privilege Class: Any

Use the QUERY COMMANDS command to display the commands and DIAGNOSE codes which you are authorized to use.

Query	COMMANDS [cmd1 cmd2 cmd3 ...]
-------	-------------------------------

where:

cmd1 cmd2 cmd3 ...

are the names of specific commands which you are trying to find out if you have authorization to use.

Usage Notes

1. The same function is available via the CP COMMANDS command.
2. Some commands have more than one version, but the user may not be authorized to use all the versions of a particular command. If the system rejects a command that is listed in the response, it is probably because the user is not authorized to use the version that includes the operand or operands that were entered.
3. For an indication of which versions of a command you are authorized to issue, use "QUERY COMMANDS cmd ...," where "cmd" is the name of the command or commands you are interested in. You will be shown the default IBM privilege classes of the command which you are authorized to issue.
4. You may specify any number of commands on the "QUERY COMMANDS cmd..." command. If a command name is entered twice, it will be displayed twice in the response. No checking is made for duplicate entries.

Response 1

```
comm1  comm2  comm3  comm4  comm5  comm6
comm7  .....
.
.
diag1  diag2  diag3  diag4  diag5  diag6
diag7  .....
```

indicates the commands and DIAGNOSE codes that the user is authorized to issue.

where:

commx

is the name of a command you are authorized to use.

diagx

identifies a DIAGNOSE code you are authorized to use.

Response 2

```
cmd1    IBMCLASS=c1,c2,c3
cmd2    IBMCLASS=ANY
cmd3    IBMCLASS=NONE
```

is the response when you specifically enter commands to check authorization for them. The response includes one line of output for each command listed.

where:

cmdx

is the name of the command you entered on the command line.

IBMCLASS = c1,c2,c3

are the IBM-defined privilege classes of this command.

IBMCLASS = ANY

indicates that the command is a class ANY command, and all users may issue the command.

IBMCLASS = NONE

indicates that you are not authorized to issue any version of the command.

Migration Notes

VM/SP HPO

1. VM/SP HPO has **COMMANDS**, not **QUERY COMMANDS**.
2. VM/XA SP allows specification of a particular command.
3. VM/XA SP commands can be up to 12 characters in length; HPO commands can be up to 8 characters.
4. VM/XA SP does not include the * (asterisk) in the list of commands.

QUERY CPLANGLIST

Privilege Class: G

Use the QUERY CPLANGLIST command to display all the language identifiers (langids) that can be set for CP in your virtual machine. You can use this command to determine whether or not a certain language is valid for your virtual machine.

Query	CPLANGLIST
-------	------------

where:

CPLANGLIST

specifies that information should be displayed about the system CP default language, the issuing user's current CP language, and all other CP languages available to the user.

Usage Notes

None.

Responses

System default language: xxxxx
User's current language: xxxxx
Other available languages: xxxxx

.
.
.

where:

System default language:

is the language identifier of the default language for CP.

User's current language:

is the language identifier of the language that is currently active for CP in your virtual machine.

Other available languages:

are other available CP languages (or NONE).

QUERY CPLANGUAGE

Privilege Class: G

Use the QUERY CPLANGUAGE command to display the language CP is currently using to display CP messages for the virtual machine session.

Query	CPLANGuage
-------	------------

where:

CPLANGUAGE

specifies that information should be displayed about the current CP language being used by the virtual machine.

Usage Notes

None.

Responses

language

where:

language

is the CP language currently being used by the virtual machine.

QUERY CPLEVEL

Privilege Class: G

Use the QUERY CPLEVEL command to determine the following CP attributes:

- Software release number
- Software service level number
- Date and time the CP system software was written to the nucleus
- Date and time the CP was last started.

Query	CPLEVEL
-------	---------

where:

CPLEVEL

displays the software release number, service level, generation date and time, and IPL date and time.

Usage Notes

None.

Responses

```
VM/XA SP RELEASE r, SERVICE LEVEL ssss
GENERATED AT mm/dd/yy hh:mm:ss timezone
IPL AT mm/dd/yy hh:mm:ss timezone
```

where:

r

is the software release number.

SERVICE LEVEL ssss

is the software service level number. The number indicates which service tape has been applied to the CP software; it cannot identify which individual updates have been incorporated into CP.

GENERATED AT mm/dd/yy hh:mm:ss timezone

indicates the date and time the CP system software was written to the nucleus.

IPL mm/dd/yy hh:mm:ss timezone

indicates the date and time the CP system software was last started.

QUERY CPTRACE

Privilege Class: A, C, E

Use QUERY CPTRACE to display the current setting of the tracing of real system events in the CP trace tables. This setting is specified by the SET CPTRACE command.

Query	CPTRace	FOR	$\left(\begin{array}{l} \text{NONSPECIFIC} \\ \text{SPECIFIC} \\ \text{SYSTEM} \\ \text{ALL} \\ \text{userid} \end{array} \right)$	[$\left[\begin{array}{l} \text{ALLcodes} \\ \text{code1 code2 ...} \end{array} \right] \dots$]
-------	---------	-----	---	---	---	---

where:

CPTRACE

displays the current setting of the tracing of real system events in the CP trace tables.

FOR NONSPECIFIC

FOR specifies which categories of trace settings are to be displayed.

NONSPECIFIC indicates the specified tracing applies to all virtual machines not specifically set or named by a previous SET CPTRACE command.

NONSPECIFIC is the default. All virtual machines begin in the NONSPECIFIC group. All members of the NONSPECIFIC group have the same codes traced.

FOR $\left(\begin{array}{l} \text{SPECIFIC} \\ \text{SYSTEM} \\ \text{ALL} \\ \text{userid} \end{array} \right)$

SPECIFIC indicates the specified tracing applies only to those virtual machines specifically set or named by a previous SET CPTRACE command. Members of the SPECIFIC group may each have a unique set of codes traced.

SYSTEM indicates the specified tracing applies to system work. SYSTEM becomes a member of the SPECIFIC group even though the specified tracing may not match the virtual machines already in the group. Therefore, SYSTEM is similar to "userid" in that it names the work for which tracing applies, and SYSTEM can also be a part of either the SPECIFIC or NONSPECIFIC groups.

ALL indicates the specified tracing applies to both the SPECIFIC and NONSPECIFIC groups of virtual machines and SYSTEM.

userid indicates the specified tracing applies only to this specific virtual machine. userid becomes a member of the SPECIFIC group, even though the specified tracing may not match others in the group.

ALLCODES

displays the setting of all CPTRACE categories.

code1 code2 ...

is either a single 4-digit trace code or the name of a category of codes. Only the setting of this code or category of codes is displayed.

QUERY CPTRACE

Usage Notes

1. The CP SET CPTRACE command determines which events are recorded. Using QUERY CPTRACE will let you know what tracing events have been set on or off. To check for only NONSPECIFIC categories, issue the command:

```
QUERY CPTRACE
```

To check for only SPECIFIC categories, issue the command:

```
QUERY CPTRACE FOR SPECIFIC
```

2. You can use the QUERY CPTRACE response to recreate the CPTRACE setting at some later point. For each response line, in the order displayed, the indicated code or category is set to ON if ENABLED or ENABLED EXCEPT, and is set to OFF if DISABLED or DISABLED EXCEPT. In the case of Example 4 (see "Example 4" on page 315) the CPTRACE setting can be recreated by issuing:

```
SET CPTRACE FOR NONSPECIFIC OFF ALLCODES
SET CPTRACE FOR NONSPECIFIC ON SVC
SET CPTRACE FOR NONSPECIFIC OFF 0200
SET CPTRACE FOR NONSPECIFIC OFF 3600
SET CPTRACE FOR NONSPECIFIC ON REALIOS
SET CPTRACE FOR NONSPECIFIC OFF SSCH
```

Response 1

The QUERY CPTRACE response shows an indented hierarchy of trace categories describing, as concisely as possible, the current CPTRACE setting. The output takes the form:

```
FOR qualifier
category ENABLED/DISABLED EXCEPT
  subcategory DISABLED/ENABLED EXCEPT
    subcategory ENABLED/DISABLED
      subcategory DISABLED/ENABLED
```

Response 2

For categories not specifically traced, the output is:

```
FOR qualifier NO SPECIFIC TRACE SET
```

Response 3

For a query on the SPECIFIC qualifier, when no specific tracing is in effect, the output is:

```
NO SPECIFIC TRACE SET
```

This response is repeated for each qualifier the trace setting is to be displayed for. In the following examples, no specific tracing is assumed unless otherwise indicated.

Examples

Example 1: If all CP TRACE entries were enabled via the command SET CPTRACE ON (or all CP TRACE entries were disabled via the command SET CPTRACE OFF), you receive the following response for the QUERY CPTRACE command:

```
FOR NONSPECIFIC
ALLCODES ENABLED (or DISABLED)
```

Example 2: All CP TRACE entries were enabled via the command SET CPTRACE ON. If you wanted to display the setting of the REALIOS trace category after deactivating trace for CLEAR SUBCHANNEL (X'1000' and X'1003') via the command SET CPTRACE OFF CSCH, you would issue the command QUERY CPTRACE REALIOS. You would receive the following response:

```
FOR NONSPECIFIC
REALIOS  ENABLED EXCEPT
CSCH     DISABLED
```

Example 3: If all CP TRACE entries were enabled via the command SET CPTRACE ON and you wanted to disable the CP TRACE entries for CALL-WITH-SAVE AREA (X'2800'), RETURN-WITH-SAVE AREA (X'2C00'), RUN USER (X'0A00'), and IUCV (X'1500'- X'1511') via the command "SET CPTRACE OFF CALLRET 0A00 IUCV," you receive the following response for the QUERY CPTRACE command:

```
FOR NONSPECIFIC
ALLCODES ENABLED EXCEPT
CALLRET  DISABLED
0A00     DISABLED
IUCV     DISABLED
```

Example 4: All CP TRACE entries were disabled via the command SET CPTRACE OFF. Perhaps you want to enable all the CP TRACE entries in category SVC and REALIOS except for SVC INTERRUPTION (X'0200'), EXIT TO THE DISPATCHER (X'3600'), and START SUBCHANNEL (X'1030'- '1033'). You would then issue the command SET CPTRACE ON SVC REALIOS OFF 0200 3600 SSCH. You will then receive the following responses for the QUERY CPTRACE command:

```
FOR NONSPECIFIC
ALLCODES DISABLED EXCEPT
SVC     ENABLED EXCEPT
0200    DISABLED
3600    DISABLED
REALIOS ENABLED EXCEPT
SSCH    DISABLED
```

Example 5: All tracing is enabled for the nonspecific and disabled for BATCH1 by issuing SET CPTRACE ON and SET CPTRACE FOR BATCH1 OFF. You will receive the following response for QUERY CPTRACE for all codes:

```
FOR NONSPECIFIC
ALLCODES ENABLED
FOR BATCH1
ALLCODES DISABLED
```

QUERY CPTRAP

VM/XA SP does not support the QUERY CPTRAP command. Equivalent functions are provided by the following CP commands:

- **QUERY TRSAVE** — Displays the destination of traces defined by TRSOURCE or displays the current status of recording for CP system tracing (traces controlled by the SET CPTRACE command).
- **QUERY TRSOURCE** — Displays the current status of the various traces that have been defined using the TRSOURCE command.
- **QUERY CPTRACE** — Displays the current setting of the tracing of real system events in the CP trace tables.

In addition, the following commands may be helpful:

- **TRSOURCE** — Supports definitions and control of I/O, data, and guest tracing.
- **TRSAVE** — Specifies where CP trace data or data from traces defined by the TRSOURCE command are to be saved. CP trace data may be saved either in system trace files or on tape. Data from traces defined by the TRSOURCE command may be saved only in system trace files.
- **SET CPTRACE** — Activates or deactivates CPTRACE entry recording of real machine events in the CP TRACE tables.

Migration Notes

VM/SP HPO: Using any abbreviation for the QUERY CPTRAP command may result in the QUERY CPTRACE command being invoked. This results in syntax errors on the CPTRACE command.

QUERY CPUID

Privilege Class: G

Use QUERY CPUID to display the 16-digit processor identification that is in use by your virtual machine.

Query	CPUid
-------	-------

where:

CPUID

requests the display of the processor identification that is in use by your virtual machine.

Usage Notes

None.

Responses

CPUID= aassssssccccdddd

where:

aa

is the version code. These two digits are set to X'FF' to identify that your virtual machine is running under VM/XA SP.

sssss

is the processor identification number. This field contains six hexadecimal digits. This is the only part of the CPUID that can be modified by means of the CP SET CPUID command or set by the system directory's OPTION control statement.

cccc

is the model number. This field is set to the model number of the real machine.

dddd

is the machine check extended logout length. This field is set to X'0000'.

QUERY DASD RESERVE

Privilege Class: B

Use the CP QUERY DASD RESERVE command to display the status of all reserved DASD devices that are defined as full pack mini-disks.

Query	DAsd RESERVE [userid]
-------	-----------------------

where:

userid

indicates that only DASD reserved to this user should be displayed. If this operand is not specified, then all reserved DASD devices will be displayed. If the userid is specified as '*', then the issuing user's ID will be used.

Usage Notes

1. Only DASD defined as full-pack minidisks with the SHARED option and an active or pending reserve are displayed.
2. This command may be used to determine the reserve status of system DASD.
3. If either the "RESERVED BY SYSTEM" response or the "RESERVE PENDING FOR SYSTEM" response consistently appear for the same device over a period of time, this may indicate a problem.

Responses

DASD rdev INT REQUIRED (if the device requires intervention)

DASD rdev { CP SYSTEM } nnnn valid { RESERVED BY } { USER userid }
 { CP OWNED } { RESERVE PENDING FOR } { SYSTEM }

are the responses you receive upon the successful completion of the command. If intervention is required, you will receive both responses.

where:

rdev

specifies the real device number of the device.

CP SYSTEM

indicates that this is a system DASD device.

CP OWNED

indicates that this is a CP-owned volume.

nnnn

specifies the number of links to this pack.

valid

is the volume ID of the DASD device. This field may be blank.

RESERVED BY

indicates that there is currently a reserve on the DASD device.

RESERVE PENDING FOR

indicates that a user (or the system) is waiting for a reserve on the DASD device.

USER userid

indicates that the real reserve is held by the system for this user.

SYSTEM

indicates that the real reserve is held by the system.

QUERY (real device)

Privilege Class: B

Use QUERY (real device) to display the status of real devices.

Query	<pre> type [ACTive FREe OFFline ATTach [userid] SYStem ALL] rdev [STATUS] rdev rdev-rdev [rdev rdev ... rdev-rdev rdev-rdev ...] [DAsd] valid </pre>
-------	---

where:

type

is one of the following:

Type	Meaning
DAsd	Direct access storage device
TApes	Magnetic tape
UR	Unit record devices
GRaf	Display device
CTCa	Channel-to-channel adapter
ALL	Requests a display of all devices and the status of real storage.

Note: If you want to display the status of real storage (from the QUERY ALL command), refer to the QUERY STORAGE command.

ACTIVE

indicates the active devices of the type specified. An active device is one that is in use by a user or the system and is neither a FREE nor an OFFLINE device.

FREE

indicates the free devices of the type specified. A free device is one that is not currently in use by a user or the system, and is neither an ACTIVE nor an OFFLINE device.

OFFLINE

indicates the offline devices of the type specified. An offline device is neither an ACTIVE nor a FREE device.

ATTACH [userid]

indicates the devices that are dedicated to a user on the system. If you specify a userid, only devices attached to that user's virtual machine are indicated. If you do not specify a userid, you receive the status of all devices dedicated to any user on the system. Attached devices are always active.

SYSTEM

indicates the devices in use by the system. A system device is always active.

ALL

indicates all devices of the type specified. The status is displayed in “active,” “free,” and “offline” order, and is equivalent to the response you would receive if you entered:

QUERY type ACTIVE
 QUERY type FREE
 QUERY type OFFLINE

rdev [STATUS]

displays the status of a real DASD device. The status presented indicates whether the device is operational, nonoperational, has an equipment check, has a channel error, or is offline.

rdev

rdev-rdev

indicates the real device address or range of real devices that are defined.

DASD valid

is the volume identification of a DASD device whose status is to be displayed. If a volume has a nonstandard label, the reported valid is unpredictable.

Usage Notes

1. If you want to display the status of real storage, refer to the QUERY STORAGE command.
2. SYSTEM is not a valid option for the QUERY GRAF and QUERY CTCA commands.
3. STATUS is a valid option only for the DASD type devices.

QUERY (REAL) ACTIVE Responses

Note: If you issue the QUERY DASD valid or the QUERY UR ACTIVE command when there are no active devices of this type, you receive one of the following responses:

UR ACTIVE NOT FOUND

- or -

DASD valid NOT FOUND

where:

valid

is the volume identification specified.

If the system is using any device that is in an intervention required state when you issue QUERY (real), you receive the following response:

type rdev INT REQ

where:

type

is the real device type.

R/O

R/W

is the access status for the device. R/O means read-only access, while R/W means read-write access.

valid

is the volume label of the device.

nnnnn

is the number of links to minidisks contained on the volume.

...

includes additional response information for shared and/or reserved DASD.

SYSTEM

indicates that the device is allocated to the system for use as user's minidisks.

OWNED

indicates that the device is used by the system for paging and spooling activity.

SHARED

indicates that the device is shared between multiple real and virtual systems.

RESERVE PENDING FOR

indicates that a user (or the system) is waiting for a reserve on the DASD device.

RESERVED BY

indicates that a reserve is active on the DASD device.

USER userid

indicates that the real reserve is held by the system for this user.

SYSTEM

indicates that the real reserve is held by the system.

If the device is a multiple-exposure paging subsystem, the following response is also displayed:

rdev rdev rdev ATTACHED userid vdev vdev vdev $\left. \begin{matrix} \{R/O\} \\ \{R/W\} \end{matrix} \right\}$ valid

where:

rdev

is the real device number of each of the three page-mode exposures.

vdev

is the virtual device number of each of the three page-mode exposures.

valid

is the volume label of the device.

ACTIVE TAPE Responses

TAPE rdev ATTACHED userid vdev $\left. \begin{matrix} \{R/O\} \\ \{R/W\} \end{matrix} \right\}$

is displayed if the tape drive is dedicated to a user.

— or —

TAPE rdev ASSIGNED SYSTEM DUMP

is displayed if the tape drive is assigned to receive system dumps.

QUERY (REAL DEVICE)

– or –

TAPE rdev ASSIGNED SYSTEM SPTAPE

is displayed if the tape is currently being used by the SPTAPE command for dumping, loading, or scanning.

TAPE rdev ASSIGNED SYSTEM TRSAVE

where:

rdev

is the real device number.

userid

is the owner's virtual machine identification.

vdev

is the virtual machine's virtual device number.

R/O

R/W

is the access status for the device. R/O means read-only access, R/W means read-write access.

SYSTEM DUMP

indicates that the tape is assigned to receive system dumps.

SYSTEM SPTAPE

indicates that the tape is currently being used by the SPTAPE command for dumping, loading, or scanning.

SYSTEM TRSAVE

indicates that the tape is in use by the TRSAVE function.

ACTIVE READER Responses

RDR rdev ATTACHED userid vdev

indicates that the card reader is dedicated to a user.

where:

rdev

is the real device number.

userid

is the device owner's virtual machine identification.

vdev

is the virtual machine's virtual device number.

– or –

RDR rdev { STARTED } { SYSTEM }
 { DRAINING } { userid }
 { DRAINED }
 { INT REQ }

is displayed for each card reader assigned to the system for spooling activity.

where:

rdev

is the real device number.

STARTED

indicates that the device is available for spooling activity.

DRAINING

indicates that the device is processing a file, but upon completion the device is drained.

DRAINED

indicates that the device is not available for spooling activity. Issue the CP START command to activate the device.

INT REQ

indicates that the device is not ready and requires manual intervention before spool file processing can resume.

SYSTEM

indicates that the device is available for input with user identification cards.

userid

indicates that the device is available for input and that no user identification cards are required. All input decks are assigned to the user identified as userid.

If the card reader is currently active with a spool file, the following response is also displayed:

RDR rdev READING userid FILE spid

where:

rdev

is the real device number.

userid

is the name of the virtual machine spool file owner.

spid

is the spool file number.

ACTIVE PRINTER Responses

{PRT} rdev ATTACHED userid vdev

indicates that the printer is dedicated to a user.

where:

rdev

is the real device number.

ATTACHED

indicates that the device is dedicated to a user.

userid

is the device owner's virtual machine identification.

vdev

is the virtual machine's virtual device number.

QUERY (REAL DEVICE)

```
PRT rdev { STARTED
          DRAINING
          DRAINED
          INT REQ
          SETUP REQ
          MOUNT REQ } {SYSTEM} CLASS class
                    {userid}
```

- or -

```
PRT rdev FORM form { MANUAL } {NOSEP} {NO3800}
                   { AUTO } {SEP}  {BEG3800}
                   { SETUP } {ANY3800}
```

For 3800 printers only:

```
{PRT} rdev FLASH name IMAGE imagelib CHARS char FCB name {HOLD }
                                                           {PURGE }
```

For impact printers only (except for a 1403 without UCS feature):

```
{PRT} rdev {NOFOLD} IMAGE imagelib CHARS chars [FCB {name}] [INDEX nn]
            { FOLD }
```

is the response when the printer is in use as a system spooling device.

where:

rdev

is the real device number.

STARTED

indicates that the device is available for spooling activity.

DRAINING

indicates that the device is processing a file, but upon completion the device is drained.

DRAINED

indicates that the device is not available for spooling activity. Issue the CP START command to activate the device.

INT REQ

indicates that the device is not ready, and requires manual intervention before spool file processing can resume.

SETUP REQ

indicates that the printer is ready to print an alignment page to allow the operator to properly align the forms on an impact printer. The operator must hit the STOP and START buttons to print another alignment page, or enter the START command to begin printing the entire file.

MOUNT REQ

indicates that the printer is waiting for the operator to mount a new form.

SYSTEM

indicates that the devices process files for any userid.

userid

indicates that the device process files only for the specified userid.

CLASS class

specifies the classes serviced by the device. An output device can service up to eight output classes.

rdev

is the real device number.

FORM form

specifies the form that an output device can service. "form" is a 1- to 8-character operator form number. This operand is ignored when the printer is in AUTO mode.

MANUAL

indicates that the operator schedules output forms. Only spool files with the form currently active on the device are processed. When the last spool file with the current form is printed or punched, the output device stops.

AUTO

indicates that the operator wants CP to schedule forms output. The currently active form is processed first. When all files with the current form have been printed or punched, CP automatically selects a new form for the device.

SETUP

indicates that the operator wants CP to print a page so that forms alignment can be verified. (This option is valid only for impact printers.) The operator gets a message indicating that the form should be set up. When the form has been set up, the operator should press the STOP button, then the START button. An alignment page is then printed. It is the first page of the file, with letters replaced with X's and numbers replaced by 9's.

Each time the operator presses the STOP and START buttons after a full alignment page is printed, another alignment page is printed. When the forms are aligned, the operator can then issue the START command for the device. The printer then returns to its previous mode (AUTO or MANUAL), and the file prints in its entirety.

If neither MANUAL, AUTO, nor SETUP is specified and the printer is currently in SETUP mode, it returns to the previous mode, AUTO or MANUAL. If no START command has been issued with the MANUAL, AUTO, or SETUP options since a system cold start, the default is MANUAL.

SEP

NOSEP

indicates whether you want a file separator for output files. This option is invalid for readers.

NO3800

specifies that no spool files that contain 3800 load CCWs are printed. Since load CCWs issued to 3800 printers can take a great deal of time, this option insures a fast output queue.

BEG3800

specifies that only spool files without 3800 load CCWs, or that contain 3800 load CCWs only at the beginning of the file, are printed on this device. This allows the user an initial setup with which the entire file can be printed.

QUERY (REAL DEVICE)

ANY3800

places no restrictions on the presence or absence of 3800 load CCWs within spool files. This option can potentially cause a slower output queue, depending on the number of 3800 load CCWs issued by the selected spool files.

FLASH name

specifies the forms overlay negative loaded into the device.

IMAGE imagelib

specifies the name of the system image library file that contains the load images (character sets, FCBs, graphic character modifications, or copy modifications) that are used for printing files and separators.

CHARS chars

specifies the name of the character set or sets used when printing the separator page.

FCB name

is the name of the forms control buffer (FCB) being used when printing the file or the separator page.

Note: This does not appear for 1403-type printers.

HOLD

indicates that spool files causing 3800 load checks are put in system HOLD status.

PURGE

indicates that spool files causing 3800 load checks are deleted.

NOFOLD

FOLD

indicates whether lowercase characters are being translated (folded) in to uppercase for printing.

INDEX nn

indicates the character position on the page at which to begin printing.

Note: This only appears for 3211 type printers.

If the device is currently active with a spool file, the following response is also displayed:

```
PRT rdev PRINTING userid FILE spid RECS { nnnn } COPY [*]nnn SEQ sss
                                     { nnnK }
                                     { nnnM }
```

where:

rdev

is the real device number.

PRINTING

indicates that the device is printing a file.

userid

is the name of the virtual machine spool file owner.

FILE spid

is the spool file number.

QUERY (REAL DEVICE)

DRAINING

indicates that the device is processing a file, but upon completion the device will be drained.

DRAINED

indicates that the device is not available for spooling activity. Issue the CP START command to activate the device.

INT REQ

indicates that the device is not ready, and requires manual intervention before spool file processing can resume.

SETUP REQ

indicates that the punch is ready to generate an alignment page to allow the operator to properly align the forms on an impact printer. The operator must hit the STOP or START buttons to print another alignment page, or enter the START command to begin printing the entire file.

MOUNT REQ

indicates that the device is waiting for the operator to mount a new form.

SYSTEM

indicates that the devices process files for any userid.

userid

indicates that the device process files only for the specified userid.

CLASS class

specifies the classes serviced by the device. An output device can service up to eight output classes.

If the device is currently active with a spool file, the following response is also displayed:

```
PUN rdev PUNCHING userid FILE spid RECS { nnnn } COPY [*]nnn SEQ sss
                                         { nnnK }
                                         { nnnM }
```

where:

rdev

is the real device number.

PUNCHING

indicates that the device is punching a file.

userid

is the name of the spool file owner.

FILE spid

is the spool file number.

```
RECS { nnnn }
      { nnnK }
      { nnnM }
```

is the number of records remaining for output in the file. If the number of records is greater than 9999, this is shown as nnnK, where K is the number of records rounded to the nearest 1000. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

is the number of copies remaining for output, where 001 indicates the last copy.

The asterisk (*) is for the 3800 printer and indicates that the printer prints each page nnn times before going on to the next page.

SEQ sss

is the sequence for the file on the punch.

ACTIVE GRAF Responses

GRAF rdev ATTACHED TO userid vdev

indicates that the real device is dedicated to a user.

GRAF rdev LOGON AS userid vdev

GRAF rdev DIALED TO userid vdev

indicates that a virtual machine user is logged on at the device or dialed through the device.

where:

rdev

is the real device number.

userid

is the owner's virtual machine user identification.

vdev

is the virtual machine's virtual device number.

Note: The status of any 328x display printer may be shown in the response to the QUERY (REAL) GRAF command.

ACTIVE GRAF/DISPLAY Responses (SNA/CCS Terminals)

vvvvvvvv CONTROLS THE FOLLOWING LU'S

luname LOGON AS userid vdev

luname DIALED TO userid vdev

where:

vvvvvvvv

identifies a VTAM service machine that is recognized by CP. Any logical units controlled by this VTAM service machine that are logged on to or dialed in to the system are described in the luname response lines following this line.

luname

identifies a SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

userid

identifies the userid of the user logged on through the SNA/CCS terminal or the userid of the virtual machine to which the SNA/CCS terminal is dialed.

vdev

is the virtual device number of the terminal.

QUERY (REAL DEVICE)

QUERY (REAL) FREE Responses

type rdev { FREE
 ENABLED
 DISABLED
 DRAINED
 valid }, . . . up to 4 responses per line

is displayed if the real device is not in use by a user or the system.

where:

type

is the real device type.

rdev

is the real device number.

FREE

indicates that the device is available.

ENABLED

indicates that a display device is enabled for user access, but is not currently in use.

DISABLED

indicates that a display device is disabled for user access.

DRAINED

indicates that a unit record device has been drained of system spooling activity.

valid

indicates that a DASD device is available and has the volume identified as valid mounted.

In the above response, type is one of the following:

Type	Meaning
DASD	Direct access storage device
TAPE	Magnetic tape
RDR	Card reader
PRT	Printer
PUN	Card punch
GRAF	Display device
CTCA	Channel-to-channel adapter
CTLR	3704 or 3705 Communications Controller
MSC	Mass storage controller
DEV	Any other device.

QUERY (REAL) OFFLINE Response

type rdev OFFLINE , . . . up to 4 responses per line

is displayed for each offline device.

where:

type

is the real device type.

rdev
is the real device number.

OFFLINE
shows the offline status of the device.

In the above response, type is one of the following:

Type	Meaning
DASD	Direct access storage device
TAPE	Magnetic tape
RDR	Card reader
PRT	Printer
PUN	Card punch
GRAF	Display device
CTCA	Channel-to-channel adapter
CTLR	3704 or 3705 Communications Controller
MSC	Mass Storage Controller
DEV	Any other device.

QUERY (REAL) Miscellaneous Devices Responses

type rdev ATTACHED userid vdev

is displayed if the real device is dedicated to a user.

where:

type
is the real device type.

rdev
is the real device number.

userid
is the owner's virtual machine user identification.

vdev
is the virtual machine's virtual device number.

In the above response, type is one of the following:

Type	Meaning
CTCA	Channel-to-channel adapter
CTLR	3704 or 3705 Communications Controller
MSC	Mass Storage Controller
DEV	Any other device.

QUERY rdev STATUS Responses

In all of the following responses, *rdev* is the real device number by which the user refers to the device.

When the device is ready and available, the system responds with:

DEVICE rdev OPERATIONAL

QUERY (REAL DEVICE)

| When the device is either disabled or not connected to the channel subsystem, the
| system responds with:

| DEVICE rdev NOT OPERATIONAL

| When the device sense data indicates an equipment check, the system responds with:

| DEVICE rdev EQUIPMENT CHECK

| When the device has indicated it has hardware problems, the system responds with:

| rdev CHANNEL ERROR

| When the device is unavailable to CP, the system responds with:

| DASD rdev OFFLINE

Migration Notes

VM/SP HPO: VM/XA SP provides real device information in numerical order by device address.

VM/XA SF

1. VM/XA SP uses GRAF rather than DISP in its response.
2. The third line in the response for impact printers is new for VM/XA SP.

QUERY DUMP

Privilege Class: B

Use QUERY DUMP to display the device number and type of the device assigned to receive CP abnormal termination dumps, and to display the current settings of the DUMP options.

Query	DUMP
-------	------

where:

DUMP

requests the display of the device number and type of the device assigned to receive system ABEND dumps, and the setting of the DUMP options.

Usage Notes

If you have assigned a device to receive CP dumps, VM/XA SP responds with the device type and the real device number. If you have not assigned a device to receive CP dumps, the response depends on the dump assignment made by VM/XA SP during initialization. If VM/XA SP was able to find enough contiguous DASD storage to assign a DASD dump, it responds to QUERY DUMP with DASD and the real device number. If VM/XA SP was unable to find enough contiguous DASD storage, it responds with PRINTER. PRINTER indicates that, at the time it is ready to create the dump, VM/XA SP will assign the dump to the first printer available in the order in which the printers were defined at initialization. If no printers are available when VM/XA SP is ready to create the dump, the dump is not created.

Responses

```
{type rdev} DUMP UNIT {V=R} {IPL} [CYL nnn]
{PRINTER}      {CP}  {NOIPL}
                {ALL}
```

where:

type rdev

indicates the device type and the real device number of the dump device. *type* can be DASD, TAPE, or PRINTER.

PRINTER

indicates that no dump device has been assigned and that VM/XA SP was unable to find enough contiguous DASD space at initialization to assign a DASD dump receiver. By default, when VM/XA SP is ready to create the dump, it assigns the dump to the first printer available in the order in which the printers were defined at initialization. If no printer is available when VM/XA SP is ready to create the dump, no dump is created.

V=R

CP

ALL

indicates whether the entire V=R area, CP-owned pages and free storage, or all pages in real storage will be included in your dump.

QUERY DUMP

IPL

NOIPL

indicates whether the system will automatically restart when the dump is completed. NOIPL indicates that the system will enter a wait state upon completion of the dump.

CYL nnn

is the number cylinders of spooling space allocated for system dumps. This value is displayed for DASD dumps only.

Migration Notes

VM/SP HPO:

1. In VM/XA SP, the response does not indicate TEMP or DUMP.
2. VM/XA SP indicates PRINTER if no dump device has been assigned.
3. The VM/XA SP response provides additional information regarding restart after a dump and spooling space cylinders for DASD dumps.

QUERY FILES

Privilege Class: D,G

Use QUERY FILES to obtain a count of reader, printer, and punch spool files. The file counts include files that are open on virtual devices or are being processed on real printers or punches.

Class G users can issue QUERY FILES to display the number of reader, printer, or punch files that they currently own or have originated.

A class D user can issue QUERY FILES to obtain a count of all spool files in the system, or files that belong to a particular user on the system.

Query	Files	[SYSTEM] [Class c] [FORM form] [ALL]	[HOLD NOHOLD SYSHOLD USERhold]	[XFER{userid2} {ALL}]	[AVAIL]
	[userid *]	[Class c] [FORM form] spoolid [ALL]	[HOLD NOHOLD SYSHOLD USERhold]	[XFER{userid2} {ALL}]	[AVAIL]

where:

SYSTEM

obtains a count of all spool files in the system. The SYSTEM option is for class D users only. It is invalid with the *spoolid* option.

userid

*

is the user identification for the user whose spool files are to be queried. You can specify userid as an asterisk (*) for information on your own files. For a class G user, the * (asterisk) is the default. Class G users cannot specify userids other than their own.

ALL

queries all of your files and is the default.

CLASS c

designates an existing spool file class. *c* is a 1-character alphanumeric field from A to Z, or 0 to 9.

spoolid

is the system-assigned spoolid number of the file that is to be queried. This option is not valid with the SYSTEM keyword, nor can it be used if a class D user is defaulting to SYSTEM. Class D must be used with a userid or an asterisk (*).

FORM form

indicates that you want to count the number of spool files with the specified form. *form* is 1-to-8 characters. For the class G user, *form* is the user form name. For the class D operator, *form* is the user form name when * is specified. Otherwise, the form is the operator form number.

AVAIL

displays only a count of available files, not those files that are open or in use.

QUERY FILES

HOLD
NOHOLD
SYSHOLD
USERHOLD

indicates that you want to query about files that are in a particular **HOLD** status. Specify **HOLD** to query files that are in user or system hold. Specify **USERHOLD** to query files that only have user hold in effect; specify **SYSHOLD** to query files that only have system hold in effect. Specify **NOHOLD** to query files that are not in user or system hold.

A file that is in hold status can not be processed by a spooling device; you can use the **CP CHANGE** command to alter the file's hold status.

XFER { **userid2** }
 { **ALL** }

XFER *userid2* allows you to query the number of spool files that you originated, and which are currently owned by the user specified by the *userid*. **XFER ALL** allows you to query the total number of spool files originated by you.

Usage Notes

If the *userid* option is used with the **XFER** option by a class D user, then the *userid* option is treated as the origin id of the files. The result will be a count of the files originated by the first user, and will include those files currently owned by the user but specified after the **XFER** option.

For example, if a class D user issues the command:

```
QUERY FILES USER1
```

a count is returned of the files that **USER1** currently owns.

If a class D user issues the command:

```
QUERY FILES XFER USER2
```

a count is returned of the files that **USER2** currently owns, but which **USER2** did not originate.

If a class D user issues the command:

```
QUERY FILES USER1 XFER USER2
```

a count is returned of the files that **USER1** originated, but which are currently owned by **USER2**.

Responses

For the class D user:

```
FILES: { nnnnnnn } RDR, { nnnnnnn } PRT, { nnnnnnn } PUN  
          { NO }           { NO }           { NO }
```

where *nnnnnnn* is the number of spool files of each type.

For the class G user:

```
FILES: { nnnn } RDR, { nnnn } PRT, { nnnn } PUN  
          { NO }           { NO }           { NO }
```

where *nnnn* is the number of spool files of each type.

Migration Notes*VM/SP HPO*

1. VM/XA SP does not allow users to specify SYSTEM and spoolid on the same command line. VM/SP HPO (Release 5) users do not see this difference.
2. VM/XA SP does not support the DEST option.
3. VM/XA SP allows room for a 7-digit response.

VM/XA SF: VM/XA SP does not allow users to specify SYSTEM and spoolid on the same command line.

QUERY FRAMES

Privilege Class: A,B,E

Use QUERY FRAMES to display the status of real storage.

Query	FRames
-------	--------

where:

FRAMES

displays the status of real storage.

Usage Notes

None.

Responses

```

SYSGEN  REAL  USABLE  OFFLINE
nnnnn   nnnnn nnnnn   nnnnn

V=R     RESNUC  PAGING  TRACE  RI0370
nnnnn   nnnnn nnnnn   nnnnn  nnnnn

AVAIL   PAGNUC  LOCKRS  LOCKCP  SAVE  FREE  LOCKRIO
nnnnn   nnnnn nnnnn   nnnnn  nnnnn nnnnn nnnnn
    
```

where:

nnnnn

is the number of 4K (4096 byte) frames.

SYSGEN

is the number of frames specified at system generation.

REAL

is the number of 4K frames of storage within the addressing capability of the processor.

USABLE

is the number of frames of usable storage.

OFFLINE

is the number of frames not configured and frames taken offline because of error.

Note: If $SYSGEN < REAL$ and $SYSGEN = USABLE + OFFLINE$, it is because pages below SYSGEN have been assigned to the hardware.

The second line of the response provides a breakdown of usable storage.

V=R

is the number of frames of virtual = real (V = R) area allocated, if any.

RESNUC

is the number of frames in the resident CP nucleus.

PAGING

is the number of frames available for paging.

FREE

is the number of frames currently in use by the free storage management routines.

TRACE

is the number of frames in the internal CP trace table.

RIO370

is the number of frames allocated for the RIO370 area, if any were requested at system generation. These frames are reserved for use by the DIAGNOSE X'98' page lock function for virtual machines running in 24-bit addressing mode. If this value is 0, programs running in 24-bit addressing mode may not use DIAGNOSE X'98'. If you issue the UNLOCK RIO370 command, this area is deallocated and its frames are added to the AVAILABLE list.

The third line of the response provides the current breakdown of the use of storage in the PAGING area.

AVAIL

is the number of frames available for use; that is, frames that are not any of the following types:

PAGNUC

is the number of frames currently in use by the pageable portion of the CP nucleus.

LOCKRS

is the number of frames currently locked for I/O operations.

LOCKCP

is the number of frames currently locked using the CP LOCK command.

EXTEND

is the number of frames allocated as an extension of the FREE storage area.

SAVE

is the number of frames currently in use by the save area manager for dynamic save areas or CPEBKs.

LOCKRIO

is the number of frames in the RIO370 area currently in use by DIAGNOSE X'98'. If an RIO370 area exists with no frames in use, you may deallocate the area by issuing the UNLOCK RIO370 command. The difference between RIO370 and LOCKRIO is the number of frames still available to satisfy additional lock requests.

The following relationships hold:

USABLE + OFFLINE = REAL (or SYSGEN, whichever is less)
 USABLE = V=R + RESNUC + PAGING + TRACE + RIO370
 PAGING = AVAIL + PAGNUC + LOCKRS + LOCKCP + SAVE + FREE
 RIO370 = LOCKRIO + (RIO370 area frames that are not in use)

Note that frames used to satisfy DIAGNOSE X'98' lock requests for 370-XA mode virtual machines are part of the normal PAGING area and are counted as LOCKRS frames.

QUERY FRAMES

Migration Notes

VM/XA SF: This information was previously given by the QUERY STORAGE command.

QUERY HOLD

Privilege Class: B, D

Use the QUERY HOLD command to display a list of users whose output has been placed in system HOLD status, or who have been placed in logon HOLD status.

If you are a class B user, you can display logon HOLD information. If you are a class D user, you can display output HOLD data. If you have both privilege classes, you can display either logon HOLD or output HOLD information.

Query	Hold [LOG]
-------	------------

where:

HOLD

requests a display of a list of all users whose output or logon has been placed in system HOLD status. If you are a class B user and specify LOG, you will see the logon HOLD information. If you are a class D user (and therefore cannot specify LOG), you will see the output HOLD information. If you have both privilege classes, you must specify LOG to see the logon HOLD information.

LOG

requests a display of logon HOLD information. This option is restricted to users with class B privileges. The logon HOLD information is displayed.

Usage Notes

Users in logon HOLD status can also have output files in HOLD status.

Response 1

$$\text{HOLD: } \left\{ \begin{array}{c} \text{nnnnnnn} \\ \text{NO} \end{array} \right\} \text{ PRT, } \left\{ \begin{array}{c} \text{nnnnnnn} \\ \text{NO} \end{array} \right\} \text{ PUN}$$

where:

nnnnnnn

NO

is the number of files of the type indicated (PRT or PUN), or NO (if none) in HOLD status. For the class B user, this is the number of files belonging to users in logon HOLD status. For the class D user, this is the number of files in system HOLD status. For a user with both class B and D, it is the combined total.

This response is always the first of two responses issued.

Response 2

NO USERS HELD

For the command QUERY HOLD (class D), this means there are no users currently in the output HOLD status. For the command QUERY HOLD LOG (class B), this means there are no users currently in the logon HOLD status.

QUERY HOLD

Response 3

userid - $\left\{ \begin{array}{l} \text{ALL} \\ \text{PRT} \\ \text{PUN} \end{array} \right\}, \dots$ (up to four users per line)

is displayed in response to the QUERY HOLD (class D) command for each user in output HOLD status.

where:

userid

is the identification of the user in output HOLD status.

ALL

PRT

PUN

indicates which type of output is being held for the user. PRT for printer files only, PUN for punch files only, and ALL for both printer and punch files.

Response 4

userid - LOG, ... (up to four users per line)

is displayed in response to the QUERY HOLD LOG (class B) command for each user in logon HOLD status.

where:

userid

is the identification of the user logon HOLD status.

If you have class D privileges also, any user in both logon and output HOLD status will receive a response as follows:

userid - LOG $\left\{ \begin{array}{l} \text{ALL} \\ \text{PRT} \\ \text{PUN} \end{array} \right\}, \dots$

where:

ALL

PRT

PUN

indicates in which type of output HOLD the user may be.

Migration Notes

VM/SP HPO: VM/XA SP allows for a 7-digit response.

VM/XA SF: In VM/XA SP, you must specify the LOG operand to display users whose logon has been placed in HOLD status.

QUERY IMG

Privilege Class: A,B,C,D,E

Use the QUERY IMG command to display information about image library (IMG) files. It is equivalent to the QUERY RDR/PRT/PUN command for spool files.

Query	IMG	[spoolid NAME fn Class c <u>ALL</u>]	[COUNT]
-------	-----	---	---------

where:

IMG

specifies that information should be displayed on image library files. This operand is valid only for a class A, B, C, D, or E user.

spoolid

is the system-assigned spoolid number of the file to be queried.

COUNT

displays only the number of image library files that match the criteria specified on the command. No other information is provided.

NAME fn

specifies the filename of the file for which information is displayed.

CLASS c

specifies the class of files to be queried. *c* is a 1-character alphanumeric field. The classes for these files are A and P where A indicates that the file is in the available state and P indicates that the file is in the pending purge state.

ALL

allows you to display information on all files of the specified type. This is the default.

Usage Notes

None

Response 1

```
OWNERID  FILE TYPE CL RECS DATE  TIME      FILENAME FILETYPE ORIGINID
*IMG     spid typ  c  nnnn mm/dd hh:mm:ss filename filetype originid
```

where:

***IMG**

indicates that the file is an IMG file owned by the system. *IMG is not a userid; it is used to indicate on what queue the file resides.

spid

is the file identification number.

typ

indicates the type of system data file. This is IMG for image library files.

QUERY IMG

c

is the file class. For the list of these classes, refer to the class listings on page 371.

nnnn

is the number of logical records in the spool file. If the number is greater than 9999, then the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

mm/dd

is the month and day that the file was closed. If the file has not yet been closed, this is the date the file was opened.

hh:mm:ss

is the time the file was closed in hours:minutes:seconds. If the file has not yet been closed, this is the time the file was opened.

filename

is the filename assigned to the file.

filetype

is the filetype (IMG) assigned to the file.

originid

is the userid of the originator of the IMG file.

Response 2

If no files as specified by the command line exist, the following response is given:

```
FILES:  NO IMG
```

Response 3

If the COUNT option is specified on the QUERY IMG command, only the number of image library files that match the criteria specified is displayed. The following is the response:

```
FILES: {nnnn} IMG
        { NO }
```

where:

nnnn

is the number of image library files. If no files are found that match the selection criteria, then NO replaces nnnn.

Migration Notes

VM/SP HPO: This function is not provided in HPO.

VM/XA SF: The format of this response has changed.

QUERY IOASSIST

Privilege Class: B

Use QUERY IOASSIST to determine the status of the interpretive execution assist for the V=R and V=F users. Class G users should use the QUERY SET command to determine the status.

Query	IOAssist
-------	----------

where:

IOASSIST

requests a display of the status of the interpretive execution assist for all V=R and V=F users.

Usage Notes

None.

Responses

ALL USERS SET - { ON }
 { OFF }

userid	IOASSIST SETTING	ACTIVE
	{ ON }	{ YES }
	{ OFF }	{ NO }
.		
.		
.		

where:

ALL USERS SET - { ON }
{ OFF }

indicates the status of the interpretive execution assist for all users. When this is set OFF, no users will use the assist. When this is set ON, the individual userid setting determines whether the user can use the assist.

userid

indicates the userid of a V=R or V=F virtual machine.

IOASSIST SETTING { ON }
{ OFF }

indicates the current status of the interpretive execution assist for this userid. ON means that the user is eligible to use the assist if the ALL USERS SET value is also ON. OFF means that the user is not using the assist.

The value given for "ALL USERS SET" determines what the status is for all users, regardless of the individual userid setting. It is changed via the SET IOASSIST ON/OFF FOR ALL command.

ACTIVE { YES }
{ NO }

indicates whether the user was actually using the start interpretive execution assist at the time the command was issued. It is only an instantaneous look at the user's status.

QUERY LINKS

Privilege Class: G

Use QUERY LINKS to display information about existing links to a virtual device.

Query	Links	vdev
-------	-------	------

where:

LINKS vdev

requests the display of information about links to the virtual device located at virtual device number vdev.

Usage Notes

None.

Responses

userid vdev $\left. \begin{matrix} \{R/W\} \\ \{R/O\} \end{matrix} \right\}$, . . . up to four responses per line

where:

userid

is the userid of the user who has a link to the specified virtual device.

vdev

is the virtual device number by which the user (userid) refers to the device.

R/O

R/W

is the type of access the user (userid) has to the virtual device. R/O is read-only and R/W is read-write access.

QUERY LOGICAL

Privilege Class: B

Use QUERY LOGICAL to display the status of logical devices.

Query	{ ldev [ldev ...] ldev-ldev [ldev-ldev ...] }
-------	--

ldev

ldev ...

ldev-ldev

ldev-ldev ldev-ldev ...

requests the display of the status of the logical device at address ldev, or the range of logical devices specified as ldev-ldev. Specify a logical device number as an L followed by four hexadecimal digits.

Response 1

If the system is using any logical device that is in an intervention required state when you issue QUERY LOGICAL, you receive the following response:

```
{ PRT } ldev INT REQ
{ GRAF }
{ DEV }
```

Response 2

If the logical device is dedicated to a user when you issue QUERY LOGICAL, you receive the following response:

```
{ PRT } ldev ATTACHED userid vdev
{ GRAF }
```

Response 3

If a virtual machine user is logged on at the device when you issue QUERY LOGICAL, you receive the following response:

```
GRAF ldev LOGON AS userid vdev
```

Response 4

If the device is free when you issue QUERY LOGICAL, you receive the following response:

```
{ GRAF } ldev { ENABLED } ... up to three
{ PRT }      { DISABLED } responses per line
{ DEV }
```

Note: In the above responses, PRT refers to a printer, GRAF refers to a display station, and DEV refers to any other device.

Migration Notes

VM/XA SF: References to DISP changed to GRAF in VM/XA SP.

QUERY LOGMSG

Privilege Class: A,B,C,D,E,F,G

Use QUERY LOGMSG to display the daily system log message.

Query	LOGmsg
-------	--------

where:

LOGMSG

displays the system log message.

Usage Notes

See the SET LOGMSG command for information on how to change the current system log message.

Response 1

[nn] logmsg text

(one line is displayed for each non-null logmsg line)

where:

nn

is the log message line number. This value can range from 01 to 50. A line number is displayed only if the user has class "B" privileges.

logmsg text

is the message text that has been assigned to the indicated log message line number.

Response 2

THERE IS NO LOGMSG DATA

is displayed when no message text has been assigned to a specified log message line number.

Migration Notes

VM/SP HPO

1. The VM/XA SP response provides an initial line of header information.
2. The VM/XA SP response for class B users includes the log message line number.

QUERY MAXUSERS

Privilege Class: A

Use the QUERY MAXUSERS command to display the current limit of logged on users.

Query	MAXUsers
-------	----------

where:

MAXUSERS

requests a display of the current user logon limit.

Usage Notes

The number of users actually logged on may be greater than the maximum value displayed. This can happen when the user logon limit is lowered to a point below the current number of users logged on. Also, the total users could be one more than the maximum user limit when the operator logs on and the maximum number of users is already logged on.

Response 1

MAXUSERS = nnnnn

is the response to display the current user logon limit.

where:

nnnnn

is the maximum number of users allowed to be logged on to the system at any one time.

Response 2

NO MAXUSERS LIMIT IS SET

is the response if no user logon limit is set.

QUERY MITIME

Privilege Class: A, B

Use QUERY MITIME to determine the current time interval settings for the missing interrupt detector.

Query	MITime
-------	--------

where:

MITIME

requests the display of the current time interval settings for the missing interrupt detector.

Usage Notes

1. When you issue QUERY MITIME, each current MITIME setting will be displayed in a form similar to the one you used when you issued the SET MITIME command to invoke the specifications.
2. When multiple responses are displayed for specific devices because of the overlap in their range and class, the first response will take precedence, since the command displays previous SET MITIME settings in a last-in, first-out sequence.

Example of Multiple Responses:

Assume the valid range of DASD devices addresses is 190-400 and several SET MITIME commands have been previously entered. If you wished to determine the current missing interrupt time interval for DASD devices 251 and 291, you would issue the command, QUERY MITIME. If the response was as follows:

```
MITIME ON 250-252 00:10
MITIME ON 190-255 01:40
MITIME ON DASD 00:30
```

then the current time interval for device 251 is 10 seconds, and the current time interval for device 291 is 30 seconds.

Responses

```
MITIME {ON } { devclass } [mm:ss]
        {OFF} { rdev
              { rdev-rdev
              { ALL
```

where:

ON

OFF

indicates whether time intervals for the specified devices have been set on. ON specifies that the intervals have been set; OFF specifies that time intervals have been set off and that no monitoring for missing interrupts is being done for the specified devices.

QUERY MITIME

devclass

indicates the device class for which you want to display the missing interrupt detector settings. "devclass" includes DASD, TAPE, GRAF, UR, or MISC. GRAF does not include the 328X, and UR does not include the 3800. MISC includes 328X, 3800, and any DASD defined as a mass storage subsystem device.

rdev

rdev-rdev

is a real device number or range or real device numbers for which you want to display the missing interrupt detector settings.

ALL

indicates that the missing interrupt detector setting is in effect for all of your devices.

mm:ss

indicates the time interval in minutes and seconds. The maximum value for mm is 99, while the maximum value for ss is 59. This is only displayed if MITIME is set ON.

Migration Notes

VM/SP HPO: The VM/XA SP response provides interval settings on a single line for each device or class of devices.

QUERY MONDATA

Privilege Class: C

Use the QUERY MONDATA command to determine if terminal event records for monitor include input and output from the user display device.

Query	MONData
-------	---------

where:

MONDATA

requests a display of whether the user display data is being included with the terminal event records for monitor.

Usage Notes

To change the MONDATA option, issue the SET MONDATA command.

Responses

MONDATA { ON }
 { OFF }

ON indicates that input from and output to user display stations is being included with the terminal event records. It is the responsibility of the installation to protect the security of this data.

OFF indicates that input from and output to user display stations is not being included with the terminal event records.

QUERY MONITOR

Privilege Class: A,E

Use the QUERY MONITOR command to display the current profile information for monitor.

Query	MONitor	[ALL Block Event INTERval PARTition RATE SAMPLe]
-------	---------	--

where:

ALL

requests a display of the domain specifications for both event and sample data. This is the default value.

BLOCK

requests a display of the current monitor block value.

EVENT

requests a display of all current settings established by the MONITOR EVENT commands. This display will show all MONITOR EVENT controls in effect and will show all active monitor event domain specifications. If an application program is connected for receiving monitor records, the userid of that machine is displayed and an indication is given if the IUCV path has been quiesced.

INTERVAL

requests a display of the current monitor sample interval value.

PARTITION

requests a display of the current monitor partition value.

RATE

requests a display of the current monitor high frequency rate value.

SAMPLE

requests a display of all current settings established by MONITOR SAMPLE commands. This display will show all sampling controls in effect and will show all active monitor sample domain specifications. If an application program is connected for receiving monitor records, the userid of that machine is displayed and an indication is given if the IUCV path has been quiesced.

Usage Notes

1. If QUERY MONITOR or QUERY MONITOR ALL is issued, data on EVENT monitoring is displayed first, followed by SAMPLE data.
2. Regarding the responses:
 - Any leading zeros are converted to blanks
 - Multiple blanks and indenting are intentional and are included for readability.

Examples:

For QUERY MONITOR SAMPLE:

```

MONITOR SAMPLE ACTIVE
          INTERVAL  30 SECONDS  PENDING INTERVAL  20 SECONDS
          RATE      11.21 SECONDS
MONITOR DCSS NAME - MONDCSS
USER CONNECTED TO *MONITOR - USER45
MONITOR  DOMAIN ENABLED
SYSTEM   DOMAIN ENABLED
PROCESSOR DOMAIN DISABLED
STORAGE  DOMAIN DISABLED
USER     DOMAIN ENABLED
  THE FOLLOWING USERIDS ARE ENABLED:
    ABLEFOX MAINT  USER01  USER02  USER03  USER04  USER05
    WILLIAMS WILSON
I/O      DOMAIN ENABLED
  THE FOLLOWING DEVICES ARE ENABLED:
    01A0-01AF  08C0-08FF  0900      0934-093F  0999-1010  1030-104F
    2222      3000-32FF  3400-3404
  
```

where:

SAMPLE

describes the monitor sample domain.

ACTIVE

indicates that SAMPLE monitoring is being performed.

INTERVAL 30 SECONDS

indicates that the current monitor sampling interval is set to 30 seconds.

PENDING INTERVAL 20 SECONDS

indicates that a new sampling interval of 20 seconds will take effect after the current sampling interval has occurred.

RATE 11.21 SECONDS

indicates that high-frequency sampling is active, with a current rate of 11.21 seconds.

The **first** leading zero in the RATE value is converted to a blank, if applicable (for example, 8.21 rather than 08.21).

MONITOR DCSS NAME - MONDCSS

indicates that the name of the DCSS used to collect monitor data is MONDCSS.

USER CONNECTED TO *MONITOR - USER45

indicates that USER45 is the virtual machine ID that is receiving IUCV notification of the monitor records.

MONITOR DOMAIN ENABLED

indicates that the monitor domain is enabled.

SYSTEM DOMAIN ENABLED

indicates that the system domain is enabled.

PROCESSOR DOMAIN DISABLED

indicates that the processor domain is disabled.

STORAGE DOMAIN DISABLED

indicates that the storage domain is disabled.

QUERY MONITOR

USER DOMAIN ENABLED

indicates that the user domain is enabled.

ABLEFOX MAINT USER01 USER02 USER03 USER04 USER05 WILLIAMS WILSON

indicates that there are nine users currently being monitored: ABLEFOX, MAINT, USER01, USER02, USER03, USER04, USER05, WILLIAMS, and WILSON.

Userids are displayed in alphabetical order.

I/O DOMAIN ENABLED

indicates that the I/O domain is enabled.

01A0-01AF 08C0-08FF 0900 0934-093F 0999-1010 1030-104F 2222 3000-32FF 3400-3404

indicates that device 0900 and device 2222 are being monitored. In addition, all devices in the following ranges are being monitored:

01A0-01AF
08C0-08FF
0934-093F
0999-1010
1030-104F
3000-32FF
3400-3404

Devices are shown in ascending order.

All leading zeros, if applicable, are shown. For example, device number 3 would be displayed as 0003.

Responses

To make it easier to find the various responses, every response is listed by option (EVENT, SAMPLE, BLOCK, INTERVAL, PARTITION, RATE and ALL).

Response 1 (QUERY MONITOR EVENT)

If QUERY MONITOR EVENT is specified, the response is as follows:

Note: If QUERY MONITOR ALL is specified, this response is issued first, followed by Response 2:

```
MONITOR EVENT {ACTIVE } BLOCK m PARTITION n  
               {INACTIVE}
```

```
MONITOR DCSS NAME - [dcssname]
```

```
USER CONNECTED TO *MONITOR - {userid} [IUCV PATH QUIESCED]
```

where:

EVENT

describes the status of event monitoring.

ACTIVE

indicates that EVENT monitoring is being performed.

INACTIVE

indicates that EVENT monitoring is not being performed.

BLOCK m

indicates the number of 4K blocks that will be accumulated before an IUCV message is sent to the virtual machine, indicating there is event data to be read.

Leading zeros, if any, are removed.

PARTITION n

indicates the number of 4K blocks reserved to hold event data.

Leading zeros, if any, are removed.

dcssname

indicates the name of the DCSS that is being used. If there is no DCSS name defined, the dcssname is specified as "NO DCSSNAME DEFINED."

userid

is the userid of the user connected to monitor.

IUCV PATH QUIESCED

indicates that monitor data collection has been temporarily suspended for both sample and event. However, high frequency sample data continues to be updated.

The following response lines are given based on the specifications in the event profile:

```
MONITOR    DOMAIN ENABLED
PROCESSOR  DOMAIN {DISABLED}
              {ENABLED }
STORAGE    DOMAIN {DISABLED}
              {ENABLED }
SCHEDULER  DOMAIN {DISABLED}
              {ENABLED }
```

Note: The following additional responses are issued only if the SCHEDULER domain is enabled, based on the specifications for the SCHEDULER domain in the event profile:

ALL USERS ENABLED

- or -

ALL USERS EXCEPT:
userid [userid ...]

- or -

THE FOLLOWING USERIDS ARE ENABLED:
userid [userid ...]

Userids are displayed in alphabetical order.

```
SEEKS      DOMAIN {DISABLED}
              {ENABLED }
```

Note: The following additional responses may be issued only if the SEEKS domain is enabled, based on the specifications for the SEEKS domain in the event profile:

ALL DEVICES ENABLED

QUERY MONITOR

- or -

ALL DEVICES EXCEPT:
rdev[-rdev]

- or -

THE FOLLOWING DEVICES ARE ENABLED:
rdev[-rdev]

USER DOMAIN { DISABLED }
 { ENABLED }

Note: The following additional responses may be issued only if the USER domain is enabled, based on the specifications for the USER domain in the event profile:

ALL USERS ENABLED

- or -

ALL USERS EXCEPT:
userid [userid ...]

- or -

THE FOLLOWING USERIDS ARE ENABLED:
userid [userid ...]

I/O DOMAIN { DISABLED }
 { ENABLED }

Note: The following additional responses may be issued only if the I/O domain is enabled, based on the specifications for the I/O domain in the event profile:

ALL DEVICES ENABLED

- or -

ALL DEVICES EXCEPT:
rdev[-rdev]

- or -

THE FOLLOWING DEVICES ARE ENABLED:
rdev[-rdev]

Response 2 (QUERY MONITOR SAMPLE)

If QUERY MONITOR SAMPLE is specified, the response is as follows:

Note: If QUERY MONITOR ALL is specified, this response follows Response 1.

MONITOR SAMPLE { ACTIVE }
 { INACTIVE }
 INTERVAL { m MINUTES } [PENDING INTERVAL { m MINUTES }]
 { n SECONDS } [{ n SECONDS }]
 RATE { m SECONDS } [PENDING RATE { n SECONDS }]
 { STOP } [{ STOP }]

MONITOR DCSS NAME - [dcssname]

USER CONNECTED to *MONITOR - {userid} [IUCV PATH QUIESCED]

where:

SAMPLE

describes the monitor sample domain.

ACTIVE

indicates that SAMPLE monitoring is being performed.

INACTIVE

indicates that SAMPLE monitoring is not being performed.

INTERVAL { m MINUTES }
 { m SECONDS }

describes the current monitor sampling interval, where m is the current interval value (1-60 for MINUTES, 6-3600 for SECONDS). If an INTERVAL value does not fall on a whole minute, the suffix SECONDS is displayed.

PENDING INTERVAL { n MINUTES }
 { n SECONDS }

indicates that a new sampling interval is to take effect after the current sampling interval has occurred. If the new INTERVAL value does not fall on a whole minute, the suffix SECONDS is displayed.

RATE { m SECONDS }
 STOP

describes the current monitor high frequency sampling rate, where m is a value between 0.01 and 30 seconds. STOP indicates that high frequency sampling is inactive.

PENDING RATE { n SECONDS }
 STOP

indicates that a new high frequency sampling rate is to take effect after the current high frequency sampling rate has occurred. n is a value between 0.01 and 30 seconds. STOP indicates that high frequency sampling will be inactive when the current RATE expires.

dcssname

indicates the name of the DCSS that is being used. If there is no DCSS name defined, the dcssname is specified as "NO DCSSNAME DEFINED."

IUCV PATH QUIESCED

indicates that monitor data collection has been temporarily suspended for sample. However, high frequency sample data continues to be updated.

The following additional response lines are given based on the specifications in the sample profile:

```
MONITOR  DOMAIN  ENABLED
SYSTEM   DOMAIN  ENABLED
PROCESSOR DOMAIN {DISABLED}
           {ENABLED}
STORAGE  DOMAIN {DISABLED}
           {ENABLED}
USER     DOMAIN {DISABLED}
           {ENABLED}
```

QUERY MONITOR

Note: The following additional responses may be issued only if the USER domain is enabled, based on the specifications for the USER domain in the sample profile:

ALL USERS ENABLED

- or -

ALL USERS EXCEPT:
userid [userid ...]

- or -

THE FOLLOWING USERIDS ARE ENABLED:
userid [userid ...]

I/O DOMAIN {DISABLED}
 {ENABLED }

Note: The following additional responses may be issued only if the I/O domain is enabled, based on the specifications for the I/O domain in the sample profile:

ALL DEVICES ENABLED

- or -

ALL DEVICES EXCEPT:
rdev[-]

- or -

THE FOLLOWING DEVICES ARE ENABLED:
rdev[-rdev]

Response 3 (QUERY MONITOR BLOCK)

If QUERY MONITOR BLOCK is specified, the response is as follows:

MONITOR BLOCK m

where:

BLOCK m

shows the number of 4Kb blocks of event records that accumulate before an IUCV message is sent to the virtual machine to indicate that there is event data to be read.

Response 4 (QUERY MONITOR INTERVAL)

If QUERY MONITOR INTERVAL is specified, the response is as follows:

MONITOR INTERVAL { m MINUTES } PENDING INTERVAL [m MINUTES]
 { n SECONDS } [n SECONDS]

where:

INTERVAL { m MINUTES }
 { n SECONDS }

describes the current monitor sampling rate. where m is the current interval value (1-60 for MINUTES, 6-3600 for SECONDS). If an INTERVAL value does not fall on a whole minute, the suffix SECONDS is displayed.

PENDING INTERVAL $\left\{ \begin{array}{l} m \text{ MINUTES} \\ n \text{ SECONDS} \end{array} \right\}$

indicates that a new sampling interval is to take effect after the current sampling interval has occurred. If the new INTERVAL value does not fall on a whole minute, the suffix SECONDS is displayed.

Response 5 (QUERY MONITOR PARTITION)

If QUERY MONITOR PARTITION is specified, the response is as follows:

MONITOR PARTITION n

where:

PARTITION n

indicates the number of 4K blocks reserved to hold event data.

Response 6 (QUERY MONITOR RATE)

If QUERY MONITOR RATE is specified, the response is as follows:

MONITOR RATE $\left\{ \begin{array}{l} m \text{ SECONDS} \\ \text{STOP} \end{array} \right\} \left[\begin{array}{l} \text{PENDING RATE } n \text{ SECONDS} \\ \text{SECONDS} \end{array} \right]$

where:

RATE $\left\{ \begin{array}{l} m \text{ SECONDS} \\ \text{STOP} \end{array} \right\}$

describes the current monitor high frequency sampling rate, m is a value between 0.01 and 30 seconds. STOP indicates that high frequency sampling is inactive.

PENDING RATE $\left\{ \begin{array}{l} n \text{ SECONDS} \\ \text{STOP} \end{array} \right\}$

indicates that a new high frequency sampling rate is to take effect after the current high frequency sampling rate has occurred, where n is a value between 0.01 and 30 seconds. STOP indicates that high frequency sampling will be inactive when the current RATE expires.

Response 7 (QUERY MONITOR ALL)

When you issue QUERY MONITOR ALL, Response 1 and Response 2 are issued.

QUERY NAMES

Privilege Class: A, B, C, D, E, F, G

Use QUERY NAMES to display a list of all the users logged on and the real or logical device number of the display to which each user is connected.

Query	Names
-------	-------

where:

NAMES

requests a display of a list of all logged on users and the real or logical device number of the display where the user is connected.

Usage Notes

Multiple, concurrent VTAM service machines are supported; therefore, multiple listings of logical unit names (lunames) owned by VTAM service machines may be displayed in the response. This portion of the QUERY NAMES response follows the rdev/ldev listings of users connected to the system; it is only displayed if there is an active VTAM service machine connected to the system.

Responses

$\left. \begin{matrix} \text{userid} \\ \text{LOGNxxxx} \\ \text{LOGLxxxx} \end{matrix} \right\} - \left. \begin{matrix} \text{rdev} \\ \text{DSC} \\ \text{ldev} \end{matrix} \right\} \dots \text{up to four responses per line.}$

```

VSM - userid
userid - luname, userid -luname, ...
userid - luname, userid -luname, ...
.
.
.
    
```

where:

userid

LOGNxxxx

LOGLxxxx

is the user identification of the virtual machine. If a user has not completed logon processing to a real device, userid is displayed in the form LOGNxxxx, where xxxx is the real device number of the display that being used. If a user has not completed logon processing to a logical device, userid is displayed in the form LOGLxxxx, where xxxx is the logical device number of the display being used.

rdev

DSC

ldev

is the device number where the user is logged on. The device may be real (rdev) or logical (ldev). DSC is displayed if the user is disconnected and does not have a real display.

VSM - userid

identifies the userid of the VTAM service machine managing the following users.

userid - luname

identifies the userids and the SNA logical unit names (lunames) of their terminals for all users accessing the system from an SNA/CCS terminal managed by that VTAM service machine.

Migration Notes*VM/SP HPO*

1. The VM/XA SP response uses LOGN instead of LOGO as in VM/SP HPO.
2. The VM/XA SP response uses LOGL instead of LOGOL as in VM/SP HPO.

VM/XA SF: The VM/XA SP response uses DSC instead of DISC as in VM/XA SF.

QUERY NLS

Privilege Class: E

Use the QUERY NLS command to display detailed information about one or more message repository files or to display the count of message repository files. Message repository files are created by the system programmer.

Query	NLS	[ALL NAME fn spoolid Class c]	[COUNT]
-------	-----	--	---------

where:

NLS

specifies that information should be displayed about message repository files.

ALL

queries all message repository files. This is the default.

spoolid

specifies the system-assigned number of the file to be queried.

NAME fn

specifies the filename of the file or files to be queried. This is the filename specified when the file was created. More than one file can exist with the same filename, although only one may be class A.

CLASS c

specifies the class of files to be queried. *c* is a 1-character alphanumeric field. The classes for these files are A or P where A indicates that the repository file is in the available state and P indicates that the file is in the pending purge state.

COUNT

displays only the number of message repository files that match the criteria specified on the command.

Usage Notes

None.

Response 1

OWNERID	FILE	TYPE	CL	RECS	DATE	TIME	FILENAME	FILETYPE	ORIGINID
*NLS	1234	NLS	A	0001	08/28	11:21:05	FRNCH	NLS	NLSMAINT
*NLS	9012	NLS	P	0001	09/28	06:22:10	ENGLS	NLS	NLSMAINT

where:

OWNERID

indicates that the file is an NLS file owned by the system. *NLS is not a userid. It indicates the queue on which the file resides.

FILE

is the spool file identification number.

TYPE

is always NLS and indicates a message repository file.

CL

is either A or P. Class A indicates that the file is available. Class P indicates that this file is in the pending purge state.

RECS

is the number of 4Kb pages in the file.

DATE

is the month and day that the file was closed.

TIME

is the time the file was closed in hours:minutes:seconds.

FILENAME

is the filename associated with the file.

FILETYPE

is the filetype associated with the file.

ORIGINID

is the userid of the originator of the file.

Response 2

FILES: NO NLS

appears if none of the specified files exist.

Response 3

FILES: { nnn } NLS
 { NO }

appears if the COUNT option is specified on the QUERY NLS command. Only the number of repository files that match the specified criteria is displayed.

where:

nnnn

is the number of repository files. If no files are found that match the selection criteria, then NO replaces nnnn.

QUERY NSS

Privilege Class: E

Use the QUERY NSS command to display information about named saved systems (NSS), and saved segments. It is equivalent to the QUERY RDR/PRT/PUN command for spool files.

The format of the QUERY NSS command is:

Query	NSS [spoolid] [MAP] [NAME fn] [COUNT] [Class c] [ALL] [USERS filename]
-------	--

where:

NSS

specifies that information should be displayed for NSS and saved segment files.

spoolid

is the spool file identification number of the file to be queried.

NAME fn

specifies the filename of the file for which information is displayed.

CLASS c

specifies the class of files to be queried. 'c' is a 1-character alphanumeric field. The classes for these files are A, P, S, or R.

ALL

allows you to display information on all NSS and saved segment files. This is the default.

MAP

indicates that information specific to NSS and saved segment files should be displayed, such as beginning and ending page numbers of the saved NSS and saved segments.

COUNT

displays only the number of files that match the criteria specified on the command. No other information is provided.

USERS filename

displays the userids that are actively using the NSS or saved segment.

Response 1

```
OWNERID FILE TYPE CL RECS DATE TIME FILENAME FILETYPE ORIGINID
*NSS spid type c nnnn mm/dd hh:mm:ss filename filetype originid
```

where:

OWNERID

indicates that the file is an NSS or saved segment file owned by the system. *NSS is not a userid; it indicates the queue on which the file resides.

FILE

is the file identification number.

TYPE

indicates the type of system data file. This is NSS for named saved system or saved segment files.

CL

is the file class. For the list of these classes, refer to the class listings on page 371.

RECS

is the number of logical records in the spool file. If the number is greater than 9999, then the number is shown as nnnK, where K represents the number of lines rounded to the nearest thousandth. If the number is greater than 999 499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

DATE

is the month and day that the file was closed. If the file has not yet been closed, this is the date the file was opened.

TIME

is the time the file was closed in hours:minutes:seconds. If the file has not yet been closed, this is the time the file was opened.

FILENAME

is the filename assigned to the file.

FILETYPE

is the filetype assigned to the file.

ORIGINID

is the userid of the last user to write to the file.

Response 2

If no files as specified by the command line exist, the following response is given:

FILES: NO NSS

Response 3 (QUERY NSS ALL MAP)

Query output with the ALL and MAP options is as follows:

FILE	FILENAME	FILETYPE	MINSIZE	BEGPAG	ENDPAG	TYPE	CL	#USERS	PARMREGS	VMGROUP
1001	CMS	NSS	000256K	00000	00004	EW	A	00015	00-15	NO
				0000E	00021	EW				
				00E00	00FFF	SR				
1002	USERSEG	DCSS	N/A	00100	001FF	SR	P	00001	N/A	N/A
1003	USERSEG	DCSS	N/A	00100	001FF	SR	R	00000	N/A	N/A
1004	SEG1	DCSS	N/A	00100	001FF	EN	S	00000	N/A	N/A
				00200	002FF	SR				NO
1006	GCS	NSS	000004M	00000	00008	EW	R	00004	00-15	YES
				00300	005FF	SW				
				00600	00FFF	SN				
1009	CPSYSTEM	CPNSS	000004M	00000	00900	EW	R	00000	NONE	N/A
1010	MONSEG	CPDCSS	N/A	00100	001FF	SC	R	00001	N/A	N/A
2006	SYSTEM1	NSS	000300K	00000	00008	EW	R	00000	OMITTED	NO
				00300	005FF	SW				
2007	SYSTEM1	NSS	000300K	00000	00008	EW	S	00000	NONE	NO
				00300	005FF	SW				
2008	SYSTEM2	NSS	000300K	00000	00008	EW	R	00000	03-08	NO
				00300	005FF	SW				
2009	SYSTEM3	NSS	000300K	00000	00008	EW	R	00000	10-10	NO
				00300	005FF	SW				
3001	BIGSEG	DCSS-S	N/A	00700	009FF	--	A	00000	N/A	N/A
3014	SEGE	DCSS-M	N/A	00700	0077F	SR	A	00001	N/A	N/A
				00780	008FF	SW				

where:

FILE

is the file identification number of the NSS or saved segment.

FILENAME

indicates the name of the NSS or saved segment.

FILETYPE

indicates the type of system data file. "NSS" is for a named saved system, "CPNSS" is for a CP system service NSS, "CPDCSS" is for a CP system service DCSS, "DCSS" is for a segment defined without the SPACE operand, "DCSS-S" is for a segment space for which members were defined, and "DCSS-M" is for a member of a segment space.

MINSIZE

indicates the minimum storage size of the virtual machine into which an NSS can be loaded. This field does not apply to a saved segment, for which "N/A" is displayed.

nnnnnK = storage size in decimal kilobytes

000nnnM = storage size in decimal megabytes

BEGPAG

indicates the beginning page number of one page range in the NSS or saved segment. For a segment space, this is the beginning page number of the entire segment space.

ENDPAG

indicates the ending page number of one page range in the NSS or saved segment. For a segment space, this is the ending page number of the entire segment space.

TYPE

is the page descriptor code of the page range in the NSS or saved segment. It indicates the type of virtual machine access permitted to pages in the range. See the DEFSEG or DEFSYS commands for a more detailed discussion of page descriptor codes. Valid codes are:

- EW - exclusive read/write access
- EN - exclusive read/write access, no data saved
- ER - exclusive read-only access
- SW - shared read/write access
- SN - shared read/write access, no data saved
- SR - shared read-only access.
- SC - CP writeable pages, shared read-only access by virtual machine, no data saved

The page range type may not apply to the entire range of a segment space; therefore “—” is used when describing these files.

CL

indicates the file class.

The file classes are:

Class Definition

- A** indicating this system data file is in the available state. This means that the system data file has been defined and saved. To determine whether the system data file is in use, examine the #USERS field in the response from the QUERY NSS command with the MAP option.
- P** indicating this system data file is in the pending purge state. It will be purged when the last virtual machine releases the NSS or saved segment, or during the next system IPL or RESTART.
- R** indicating this system data file has restricted access and is in the available state. To determine whether it is in use, examine the #USERS field in the response from the QUERY NSS command with the MAP option. Access to an NSS, DCSS, or segment space name with this class requires a NAMESAVE directory entry for the NSS, DCSS, or segment space name.
- S** indicating this system data file is in the skeleton state. This means that a DEFSEG or DEFSYS command has been executed for the system data file. The SAVESEG or SAVESYS can now be executed to complete this system data file.

#USERS

indicates the number of users attached to the NSS or saved segment.

PARMREGS

indicates a range of registers in which parameters are passed to the virtual machine at IPL time. Possible values are:

- m-n** displays the range of registers, where m and n are decimal integers between 00 and 15 (inclusive) and m is less than or equal to n.
- NONE** displayed when PARMREGS = NONE was specified on the DEFSYS command.
- OMITTED** indicates that the PARMREGS option was not specified on the DEFSYS command.
- N/A** displayed for a saved segment. Registers are not saved for saved segments; therefore, this field does not apply.

VMGROUP

indicates, for NSS files only, whether the VMGROUP attribute was specified on the DEFSYS command when the named saved system was created. For DCSS files, this field is not applicable (N/A).

Response 4 (QUERY NSS NAME MAP)

QUERY NSS NAME output with the MAP option is as follows when the named system data file is a segment space:

FILE	FILENAME	FILETYPE	MINSIZE	BEGPAG	ENDPAG	TYPE	CL	#USERS	PARMREGS	VMGROUP
4111	STUFF	DCSS-S	N/A	00600	008FF	--	A	00001	N/A	N/A
4212	PIECE1	DCSS-M	N/A	00600	0067F	SR	A	00000	N/A	N/A
4513	PIECE2	DCSS-M	N/A	00680	006FF	SR	A	00001	N/A	N/A
3614	PIECE3	DCSS-M	N/A	00700	0077F	SR	A	00001	N/A	N/A

Response 5 (QUERY NSS NAME MAP)

QUERY NSS NAME output with the MAP option is as follows when the named system data file is a member saved segment:

FILE	FILENAME	FILETYPE	MINSIZE	BEGPAG	ENDPAG	TYPE	CL	#USERS	PARMREGS	VMGROUP
6513	ITEM7	DCSS-M	N/A	00680	006FF	SR	A	00002	N/A	N/A
9835	BIG	DCSS-S	N/A	00600	008FF	--	A	00000	N/A	N/A
3435	BIGGEST	DCSS-S	N/A	00300	007FF	--	A	00001	N/A	N/A
7545	BIGGER	DCSS-S	N/A	00600	009FF	--	A	00000	N/A	N/A

Note that the number of users indicates only users associated with that filename.

Response 6 (QUERY NSS COUNT)

If the COUNT option is specified on the QUERY NSS command, only the number of named saved systems that match the criteria specified is displayed. The following is the response:

FILES: {nnnn} NSS
 { NO }

where:

nnnn

is the number of named saved systems. If no files are found that match the selection criteria, then NO replaces nnnn.

Response 7 (QUERY NSS USERS SEGSPACE)

If the USERS option is specified on the QUERY NSS command for a segment space with active users, this response follows:

```
FILE FILENAME FILETYPE CLASS
0011 SEGSPACE DCSS-S A
```

```
USER0001 USER0002 USER0003 USER0004 USER0005 USER0006 USER0007
```

```
FILE FILENAME FILETYPE CLASS
0012 MEMSEG0 DCSS-M A
```

```
USER0011 USER0012 USER0013 USER0014 USER0015 USER0016 USER0017
USER0021 USER0022 USER0023 USER0024 USER0025 USER0026 USER0027
```

```
FILE FILENAME FILETYPE CLASS
0013 MEMSEG1 DCSS-M A
```

NONE

where:

FILE

is the spool ID of the file.

FILENAME

indicates the name of the NSS or DCSS.

FILETYPE

indicates the type of system data file. NSS is for a named saved system, DCSS is for a discontinuous saved segment, CPNSS is for CP NSS, and CPDCSS is for a CP DCSS, DCSS-S is for a segment space, and DCSS-M is for a member segment.

CLASS

indicates the file class. For a list of the class descriptions that apply to this command, see page 371.

Response 8 (QUERY NSS USERS MEMSEG2)

If the USERS option is specified on the QUERY NSS command for a member segment with active users, this response follows:

```
FILE FILENAME FILETYPE CLASS
1111 MEMSEG2 DCSS-M A
```

```
USER1001 USER0222 USER2203 USER2224 USER6605 USER7006 USER9097
USER2001 USER2222 USER0223 USER0224 USER6005 USER0706 USER0907
USER3001 USER0022 USER0023 USER0024 USER0065 USER0076 USER9097
```

```
FILE FILENAME FILETYPE CLASS
0011 SEGSPACE DCSS-S A
```

```
USER0001 USER0002 USER0003 USER0004 USER0005 USER0006 USER0007
```

Response 9 (QUERY NSS USERS PDCSS)

If the USERS option is specified on the QUERY NSS command for a DCSS with active users, this response follows:

```
FILE FILENAME FILETYPE CLASS
2222 PDCSS DCSS A
```

```
USER1001 USER0222 USER2203 USER2224 USER6605 USER7006 USER9097
USER2001 USER2222
```

QUERY NSS

Response 10 (QUERY NSS USERS CMSNSS)

If the USERS option is specified on the QUERY NSS command for an NSS with active users, this response follows:

```
FILE FILENAME FILETYPE CLASS
3333 CMSNSS   NSS       A
```

```
USER1001 USER0222 USER2203 USER2224 USER6605 USER7006 USER9097
USER2001 USER2222
```

Response 11 (QUERY NSS USERS XYZABC)

If the USERS option is specified on the QUERY NSS command for a nonexisting NSS/DCSS or the NSS/DCSS is not active, this response follows:

```
XYZABC NOT IN USE OR AN INVALID NSS/DCSS NAME
```

Migration Notes

VM/SP HPO: This function is not provided by VM/SP HPO.

QUERY PASSWORD

Privilege Class: G

Use **QUERY PASSWORD** to determine whether you can include a password on the command line of the **LINK**, **LOGON**, **AUTOLOG**, or **XAUTOLOG** commands.

Query	PASSWORD	$\left. \begin{array}{l} \text{LINK} \\ \text{LOGON} \\ \text{XAUTOLOG} \\ \text{AUTOLOG} \end{array} \right\}$
-------	----------	---

where:

PASSWORD

requests a display to show whether you can include a password on a command line for a command that requires a password.

LINK

displays whether you can include a password on a command line for the **LINK** command.

LOGON

displays whether you can include a password on a command line for the **LOGON** command.

AUTOLOG

XAUTOLOG

displays whether you can include a password on a command line for the **AUTOLOG** or the **XAUTOLOG** command.

Usage Notes

1. See the **SET PASSWORD** command for an explanation of how to set password suppression. The commands affected by **SET PASSWORD** are **LINK**, **LOGON**, **AUTOLOG**, and **XAUTOLOG**.
2. For security reasons, the password suppression facility permits your installation to require that issuers of the **CP LINK** command do not put the **LINK** access mode password in the command when they issue it. Instead, the user is prompted to enter the password, and the input field is masked.
3. If you issue **QUERY PASSWORD** using **DIAGNOSE** code X'08', you receive one of the following return codes in your **RY** register:

RC Meaning

- 0** Indicates that you may enter a password on a command line.
- 4** Indicates that you cannot enter a password on a command line.
- 8** Error—invalid operands following on the command line.

Responses

$\left. \begin{array}{l} \text{LINK} \\ \text{LOGON} \\ \text{XAUTOLOG/AUTOLOG} \end{array} \right\}$ PASSWORD MUST BE ENTERED SEPARATELY

indicates that you cannot enter the access mode password on the command line.

QUERY PASSWORD

{ LINK
LOGON
XAUTOLOG/AUTOLOG } PASSWORD MAY BE INCLUDED ON THE COMMAND LINE

indicates that you can enter the access mode password on the command line.

Migration Notes

VM/SP HPO: This function is not provided in VM/SP or VM/SP HPO.

VM/XA SF: The syntax for this command is NOT upwardly compatible with the previous release. The user must indicate specifically which password suppression facility setting is to be displayed.

QUERY PATHS

Privilege Class: A,B

Use the QUERY PATHS command to display all the paths installed to a specified device and to display their current status, either logically online or offline.

Query	PATHS [T0] rdev
-------	-----------------

where:

rdev

is the real device number whose path allocation information is desired.

Usage Notes

The QUERY PATHS command shows all the paths installed to a specific device. For information about the devices associated with a specific channel path, see the QUERY CHPID command.

Examples:

```
QUERY PATHS TO 192
```

A typical response to the above example command may be:

```
PATH STATUS FOR DEVICE 0192: ONLINE - 1A,30,22,10,60,40,12 OFFLINE - 11
```

Responses

```
PATH STATUS FOR DEVICE rdev: ONLINE - y1,yn OFFLINE - x1,xn
```

is the response for the QUERY PATHS command showing the logical online or offline status for each channel path id to the specified device.

where:

rdev

is the real device number.

y1,yn

are the online channel paths to the device.

x1,xn

are the offline channel paths to the device.

QUERY PER

Privilege Class: G

In VM/XA SP, the QUERY TRACE command displays the current TRACE settings. QUERY PER is recognized as a synonym for QUERY TRACE. For further information, please refer to QUERY TRACE.

Migration Notes

VM/SP HPO

1. QUERY PER is accepted as a synonym for QUERY TRACE.
2. The VM/XA SP response returns the setting of all TRACE options instead of those explicitly entered as HPO.

QUERY PFnn

Privilege Class: G

Use QUERY PF to display the program function key command lines associated with a specified 3270 display device program function key.

Query	PF[nn]
-------	--------

where:

PF[nn]

requests a display of the program function command lines associated with the program function key you designate by the value nn. The value nn is a number from 1 (or 01) to 24. If you do not specify a value for nn, all program function keys and their associated data lines are displayed.

Usage Notes

See the CP SET command for an explanation of how to define and use program function keys.

Response 1

PFnn [SUBSTITU] { DELAY } pfdata
 { IMMED }
 { NODISP }

where:

PFnn

is the program function key number.

SUBSTITU

indicates that tokens from the input are substituted in the pfdata. Otherwise, no substitution takes place.

DELAY

IMMED

NODISP

indicates when the program function is executed. IMMED means that the program function is executed when you press the PF key. DELAY means that the program function is displayed in the input area when you press the PF key. NODISP means that the program function is executed when you press the PF key but the function is not redisplayed on the terminal.

pfdata

is the program function command line associated with the key.

Response 2

If there is no function defined for the program function key, you receive the following response:

PFnn UNDEFINED

Migration Notes

VM/XA SF

1. The VM/XA SP response does not display saved lines under a PF key set to RETRIEVE.
2. If QUERY PFnn is issued when all the PF keys are undefined, VM/XA SF gives a one-line response; VM/XA SP provides one line per PF key.

QUERY PROCESSORS

Privilege Class: A,B,C,E

Use the **QUERY PROCESSORS** command to display a list of all real processors indicating how each real processor is being used on the system.

Query	PROcEssors
-------	------------

where:

PROCESSORS

displays a list of real processors on the system.

Usage Notes

None.

Responses

```
PROC nn MASTER
PROC nn ALTERNATE
PROC nn DEDICATED TO userid CPU mm
```

where:

nn

is the real processor address.

MASTER

indicates the master processor.

ALTERNATE

indicates an alternate processor (a nondedicated processor other than the master).

DEDICATED TO userid CPU mm

indicates the virtual CPU to which real processor nn has been dedicated. The "userid" indicates the user who owns the virtual CPU. "mm" is the virtual address of that CPU.

Migration Notes

VM/SP HPO

1. VM/XA SP allows you to spell out the entire operand **PROCESSORS**.
2. The VM/XA SP response indicates whether a processor is the **MASTER**, an **ALTERNATE**, or a **DEDICATED** processor.

QUERY QUICKDSP

Privilege Class: A,E

Use the QUERY QUICKDSP command to indicate whether a user has the QUICKDSP attribute.

Query	QUICKDSP userid
-------	-----------------

where:

userid

is the identifier of the virtual machine whose QUICKDSP setting is to be displayed.

Usage Notes

None.

Responses

USER userid: QUICKDSP={ON }
 {OFF }

appears to the QUERY QUICKDSP command issuer.

QUERY READER/PRINTER/PUNCH

Privilege Class: D,G

Use QUERY READER/PRINTER/PUNCH to display detailed information about one or more spool files of the specified type. The display includes files that are open on virtual devices, that are closed and on the device queue, or that are being processed on the real printers, punches, or readers. The list of spool files is displayed in the order that they will be processed.

Class G users can obtain information about reader, printer, or punch files that they own or originate.

The class D user can obtain information about all spool files in the system, or spool files belonging to any user on the system.

Query	Reader [SYSTEM][[Class c][FORM form]] [AVAIL]	[HOLD NOHOLD SYSHOLD USERHOLD]	[EXP TBL ALL]	[XFER {userid} {ALL}]
	Printer PUnch			
	Reader [userid][[Class c][FORM form]] [AVAIL]	[HOLD NOHOLD SYSHOLD USERHOLD]	[EXP TBL ALL]	[XFER {userid} {ALL}]
	Printer [*] [spoolid] PUnch			

Note: The options userid and system are permitted for class D users only.

where:

READER
RDR
PRINTER
PRT
PUNCH
PCH

displays the following information about your reader, printer, or punch files:

- Userid (of user who created the file)
- Spool file identification
- Class and originating device type
- Number of logical records in the file
- Number of copies specified for the file
- File HOLD, KEEP, and MSG status
- FORM name.

SYSTEM

allows the class D user to obtain information about all files in the system, regardless of ownership. This is not allowed with the "spoolid" option.

userid

*

is the user identification for the user whose spool files are to be queried. You can specify an asterisk (*) if you are querying about your own files.

The class D user can obtain information about reader, printer, or punch spool files that are owned by any user of the system by specifying that user's userid. If class D users want to obtain a display about their own spool files, they must enter their own userid or an asterisk on the command line.

CLASS c

designates an existing spool file class. "c" is a one-character alphanumeric field from A to Z, or 0 to 9.

FORM form

indicates that you want to obtain spool file information for files with the specified form. "form" is one to eight characters. For the class G user, "form" is the user form name. For the class D operator, "form" is the user form name when you specify "*&". Otherwise, "form" is the operator form number.

AVAIL

displays only available files, not those files which are open or in use.

spoolid

is the system-assigned number of the spool file that is to be queried. This option is not valid with the SYSTEM keyword.

HOLD**NOHOLD****SYSHOLD****USERHOLD**

indicates that you want to query about files that are in a particular hold status.

- Specify hold to query files that are in user or system hold. A file that is in hold status cannot be processed by the spooling device. If you are a class G user, you can change a file's user hold status by using the CP CHANGE command; if you are a class D user, you can change the system hold status also by using the CP CHANGE command.
- Specify NOHOLD to query files that are not in user or system HOLD.
- Specify SYSHOLD to query files that only have system HOLD in effect.
- Specify USERHOLD to query files that only have user HOLD in effect.

EXP**TBL**

displays expanded information for spool files.

The information displayed is:

- User form name and operator form number
- Forms control buffer (FCB)
- Name and number of the copy modification module
- Flash name
- Number of copies to be flashed (flash count)
- Where the 3800 LOAD CCWs are positioned within the spool file
- Name of the character set or sets.

TBL is accepted as a substitute for the EXP option.

ALL

displays additional information for spool files. In addition to the information normally displayed for the reader, printer, or punch files, the following is also displayed:

- Date and time the file was created
- Filename and filetype of file (if any)
- Distribution code of the file.

Note: The form name is not displayed when you use the ALL option.

XFER { **userid** }
 { **ALL** }

allows you to query the number of spool files currently owned by other users that you originated. XFER *userid* displays information about the spool files that you originated, and which are currently owned by the user specified by the *userid*. XFER ALL displays information about all the spool files originated by you.

For both the class G and class D users, issuing the XFER ALL or the *userid* operands in a QUERY will not change the response header, but will change the information listed underneath the header.

Usage Notes

1. If the *userid* option is used with the XFER option, then the *userid* option is treated as the origin id of the files. The result will be information about the files originated by the first user and which are currently owned by the user specified after the XFER option.

For example:

If a class D user issues the command:

QUERY PRT USER1

he will receive information about the files which USER1 currently owns.

If a class D user issues the command:

QUERY PRT XFER USER2

he will receive information about the files which USER2 currently owns but did not originate.

If a class D user issues the command:

QUERY PRT USER1 XFER USER2

he will receive information about the files which USER1 originated, but which USER2 currently owns.

Response 1

The following response is issued when you specify QUERY RDR/PRT/PUN without the ALL option:

For a class G user:

```

ORIGINID FILE CLASS RECORDS CPY HOLD FORM KEEP MSG
userid spid a typ nnnnnnnn nnn NONE form ON ON
                *nnn USER OFF OFF
                SYS
                USYS
    
```

QUERY READER/PRINTER/PUNCH

For a class D user:

OWNERID	FILE	CLASS	RECORDS	CPY	HOLD	USERFORM	OPERFORM	KEEP	MSG
userid	spid	a	typ	nnnnnnnn	nnn	NONE	userform	operform	ON ON
				*nnn	USER			OFF	OFF
					SYS				
					USYS				

Response 2

The following response is issued when you specify QUERY RDR/PRT/PUN with the ALL option:

ORIGINID	FILE	CLASS	RECORDS	CPY	HOLD	DATE	TIME	NAME	TYPE	DIST
userid	spid	c	typ	nnnnnnnn	nnn	NONE	mm/dd hh:mm:ss	filename	filetype	distcode
				*nnn	USER					
					SYS					
					USYS					

Response 3

If you use the EXP option, then you receive the following response:

ORIGINID	FILE	CLASS	RECORDS	FLASH	FCB	MDFY	FLSHC	LOAD	CHARS					
userid	spid	c	typ	nnnnnnnn	name	fcf	mod	n	ccc	NO	name	name	name	name
										BEG				
										ANY				

Response 4

If the file is open (being created or read) on a virtual device, then the OPEN status and the virtual device address (vdev) are displayed in place of the DATE and TIME fields:

ORIGINID	FILE	CLASS	RECORDS	CPY	HOLD	DATE	TIME	NAME	TYPE	DIST
userid	spid	c	typ	nnnnnnnn	nnn	NONE	OPEN- vdev	filename	filetype	distcode
				*nnn	USER					
					SYS					
					USYS					

Response 5

If the file is open (being read or created) for the SPTAPE command, then the OPEN status and SYS are displayed in place of the DATE and TIME fields:

ORIGINID	FILE	CLASS	RECORDS	CPY	HOLD	DATE	TIME	NAME	TYPE	DIST
userid	spid	c	typ	nnnnnnnn	nnn	NONE	OPEN- SYS	filename	filetype	distcode
				*nnn	USER					
					SYS					
					USYS					

Response 6

If a closed spool file is in the process of being printed or punched on a real output device, then the PRT or PUN status and the number of records remaining to be printed in the current copy are displayed in the DATE and TIME fields:

ORIGINID	FILE	CLASS	RECORDS	CPY	HOLD	DATE	TIME	NAME	TYPE	DIST
userid	spid	c	typ	nnnnnnnn	nnn	NONE	PRT- (nnnn)	filename	filetype	distcode
				*nnn	USER	PUN- (nnnK)				
					SYS	(nnnM)				
					USYS					

Note: If the number of records remaining to be printed is greater than 9999, then nnnK will be shown, and if the number of records remaining is greater than 999499 then nnnM is shown.

Response 7

If a spool file is open (being read) on a real card reader, then the OPEN status and the real device number (rdev) are displayed in place of the DATE and TIME fields:

```

ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME  NAME  TYPE  DIST
userid  spid c typ nnnnnnnn nnn NONE OPEN- rdev  filename filetype distcode
                                *nnn USER
                                    SYS
                                        USYS
    
```

Response 8

If no files of the specified type exist, one of the following responses is given, corresponding to the specified file type:

NO RDR FILES

— or —

NO PRT FILES

— or —

NO PUN FILES

where:

ORIGINID

OWNERID

ORIGINID heading the title line for the spool file data is altered to OWNERID when the userid or XFER operand is used, or when a class D user is defaulting to or specifying SYSTEM. In that event, OWNERID represents the owner of the file.

userid

is the identification of the user who owns the file.

spid

is a system-assigned number which is used by the system to identify the file.

c

is the spool file class.

typ

shows the device type that originally produced the spool file. The type can be RDR, PRT, PUN, CON, DMP or SYS.

nnnnnnnn

is the number of logical records contained in the file.

nnn

***nnn**

is the number of copies requested for the spool file. This is shown as a 3-digit number (nnn). The asterisk (*) indicates that the 3800 printer should print each page from the spooled output files nnn times before going on to the next page.

NONE

USER

SYS

USYS

represents the file's HOLD status. NONE indicates there are no holds on the file, USER indicates the file is in user hold status, SYS indicates the file is in system hold, and USYS indicates the file is in both user and system hold.

FORM form

is the name of the paper type as understood by the user (displayed for a class G user).

userform

is the name of the paper type as understood by the user (displayed for a class D user).

operform

is the name of the paper type as understood by the operator (displayed for a class D user).

**KEEP { ON }
 { OFF }**

indicates whether or not the file is retained after being processed. ON indicates that KEEP is in effect and the file will be retained after processing, and OFF indicates that KEEP is not in effect.

**MSG { ON }
 { OFF }**

indicates whether or not messages about the file printing or punching will be displayed. ON indicates these messages will be displayed; OFF indicates they will not be displayed.

mm/dd

is the month and day that the file was created.

hh:mm:ss

is the time when the file was created in hours:minutes:seconds.

filename

is the filename assigned to the file (if any).

filetype

is the filetype assigned to the file (if any).

distcode

is the distribution code of the file.

FLASH name

is the name of the forms overlay frame to be superimposed on the output if printed on a 3800 printer.

FCB name

is the name of the FCB module used in the vertical formatting of a page.

MDFY name

is the name of the copy modification module used to alter output text if printed on a 3800 printer.

n

is the copy modification character set number.

FLSHC ccc

is the number of copies that are printed while the forms overlay frame is in place if printed on a 3800 printer.

LOAD load

is where the 3800 LOAD CCWs are positioned within the spool file – not at all (NO), at the beginning (BEG), or interspersed throughout the spool file (ANY).

CHARS char

is the name of the character set or sets used to generate output if printed on a 3800 printer.

Migration Notes

VM/SP HPO

1. The VM/XA SP response provides information about open spool files.
2. The VM/XA SP response to QUERY READER/PRINTER/PUNCH provides additional information regarding the KEEP and MSG settings.
3. VM/XA SP supports additional options: EXP, XFER, and AVAIL.
4. VM/XA SP does not allow users to specify SYSTEM and spoolid on the same command line. VM/SP HPO (Release 5) users do not see this difference.
5. VM/XA SP indicates page copies by an * (asterisk) in the copy count, rather than with the FLASH name as in HPO.
6. VM/XA SP does not support the DEST and PSF options.

VM/XA SF: VM/XA SP does not allow users to specify SYSTEM and spoolid on the same command line.

QUERY RECORDING

Privilege Class: A, B, C, E, F

Use QUERY RECORDING to determine the status of CP data collection for ACCOUNTING and EREP records, and to determine if record retrieval is in progress.

Query	RECORDING
-------	-----------

where:

RECORDING

requests the status of data collection for ACCOUNTING and EREP records.

Usage Notes

1. If the PENDING indication is ON and the record count has not decreased, then an error has probably occurred and data retrieval is not proceeding normally. The virtual machine retrieving the records should issue a SYSTEM RESET command and reinitiate data retrieval.
2. Class A and B users receive the status of all recording functions. Other users will receive the status of the recording functions for which they are eligible to retrieve data records.

Responses

```

RECORDING      COUNT  LMT  USERID  COMMUNICATION
function {ON }  nnnnn  mmm  userid  {INACTIVE
                {OFF}
                {ACTIVE [PENDING]}
    
```

where:

function

is the name of the recording function. ACCOUNT is the name used for accounting records; EREP is used for error recording records.

ON

OFF

indicates whether records are being generated.

nnnnn

is the number of records pending in storage.

mmm

is the threshold limit used to calculate when the system operator and retrieving virtual machine will be notified that uncollected records are accumulating.

userid

is the user identification of the virtual machine that is eligible to retrieve the associated records. This user can alter selected fields in the CP recording table by using the RECORDING command.

COMMUNICATION

is the status of the virtual machine's communication with CP for record retrieval. **INACTIVE** means that the user must initiate communication with CP to begin record retrieval. **ACTIVE** means that the user has already initiated the communication. **PENDING** means that CP to virtual machine communication has been initiated and that CP has sent a record to the virtual machine for retrieval.

QUERY SCREEN

Privilege Class: G

Use the QUERY SCREEN command to display the extended color and extended highlighting values in effect for the virtual machine console.

Query	SCReen
-------	--------

Usage Notes

1. The extended color and highlighting values are set either by the SCREEN statement in your directory entry or are set when you issue the SCREEN command.
2. If you do not have a SCREEN directory statement in your directory entry and you have not issued the SCREEN command during the current session, the response to the QUERY SCREEN command is DEFAULT for the color value and NONE for the extended highlight value.

Responses

CPOUT	color	exthilight	VMOUT	color	exthilight
INREDISP	color	exthilight	INAREA	color	exthilight
STATAREA	color	exthilight			

where:

color

is the color value currently in effect for your virtual console.

exthilight

is the extended highlight value currently in effect for your virtual console.

QUERY SDF

Privilege Class: A,B,C,D,E,G

Use QUERY SDF to allow a user to obtain a count of all system data files.

Query	SDF
-------	-----

Usage Notes

1. The count of IMG files is displayed only for class A, B, C, D, and E users.
2. The count of NSS files is displayed only for class E users.
3. The count of UCR files is displayed only for class A, B, and C users.
4. The count of NLS files is displayed only for class E users.
5. The count of TRF files is displayed only for class A, C, D, E, and G users.
6. For a class G user, only the number of TRF files that the class G user owns is displayed.
7. QUERY SDF ALL returns the same information as QUERY SDF.

Responses

The response from the QUERY SDF command indicates the count of the different types of system data files. The response is as follows:

FILES: [{ nnnn } IMG] [{ nnnn } NSS] [{ nnnn } UCR] [{ nnnn } TRF] [{ nnnn } NLS]
 [{ NO }] [{ NO }] [{ NO }] [{ NO }] [{ NO }]

where:

nnnn

is the number of system data files of this type.

QUERY SECUSER

Privilege Class: G

Use the QUERY SECUSER command to determine the identity and status of the secondary user specified in your CONSOLE directory statement.

Query	SECuser
-------	---------

where:

SECUSER

requests a display of the identity and status of the secondary user for your userid.

Usage Notes

None.

Response 1

SECONDARY USER userid IS DISCONNECTED

is the response if your directory entry has a secondary userid specified on the CONSOLE statement, and that user is disconnected. If you disconnect from the system, the result is the same as if no secondary userid were designated in your directory entry, unless the secondary user's virtual machine handles console I/O through the message system service, in which case secondary console support is provided to your disconnected virtual machine. If the secondary user reconnects, they can provide secondary console functions for your virtual machine.

Response 2

SECONDARY USER userid IS NOT LOGGED ON

is the response if your directory entry has a secondary userid specified on the CONSOLE statement, and that user is not logged on. If you disconnect from the system, the result is the same as if no secondary userid were designated in your directory entry. If the secondary user logs on, they can provide secondary console functions for your virtual machine.

Response 3

SECONDARY USER userid IS LOGGED ON

is the response if your directory entry has a secondary userid specified on the CONSOLE statement, and that user is capable of providing secondary console support for you. If you, the primary user, disconnect from the system, that secondary user receives your console output until you reconnect or log off. The secondary user may provide console input to your virtual machine or to CP on your behalf by using the SEND command.

Response 4

SECONDARY USER IS UNDEFINED

is the response if your directory entry has no secondary userid specified on the `CONSOLE` statement. If you disconnect from the system, your console output will be lost until you reconnect. If you have spooled your console output, it is retained in your console spool file.

If your virtual machine requests console input while disconnected and you do not have a functional secondary user, CP waits 15 minutes for you to reconnect. If you do not reconnect within that time, CP logs you off the system.

QUERY SET

Privilege Class: G

Use QUERY SET to display the status of many SET command functions.

Query	Set
-------	-----

where:

SET

requests a display of the status of the SET command functions.

Usage Notes

None.

Responses

```

MSG { ON } , WNG { ON } , EMSG { ON } , ACNT OFF , RUN { ON }
  { OFF }   { OFF }   { CODE }
  { IUCV }  { IUCV }   { TEXT }
                               { OFF }
                               { IUCV }

LINEDIT { ON } , TIMER { ON } , ISAM OFF , ECMODE ON
  { OFF }   { OFF }
                { REAL }

ASSIST OFF , PAGEX { ON } , AUTOPOLL { ON }
                { OFF }   { OFF }

IMSG { ON } , SMSG { ON } , AFFINITY NONE , NOTRAN { ON }
  { OFF }   { OFF }
  { IUCV }  { IUCV }
                               { OFF }

VMSAVE OFF , 370E OFF

STBYPASS OFF , STMULTI OFF 00/000

MIH { ON } , VMCONIO { OFF } , CPCONIO { OFF } , SVCACCL OFF , CONCEAL { ON }
  { OFF }   { IUCV }   { IUCV }
                                     { OFF }

MACHINE { 370 } , SVC76 { CP } , NOPDATA { ON } , IOASSIST { ON }
  { XA }   { VM }   { OFF }
                                     { OFF }

CCWTRAN { ON }
  { OFF }
  
```

where:

MSG { **ON** }
 { **OFF** }
 { **IUCV** }

indicates whether messages sent from other users by means of the CP MESSAGE command will be displayed. ON specifies that messages will be displayed. OFF specifies that messages will not be displayed. IUCV specifies that messages are passed to the virtual machine via IUCV if a connection to the message system service exists. If no IUCV connection exists, the message will be handled as if MSG were set ON.

WNG { **ON** }
 { **OFF** }
 { **IUCV** }

indicates whether warning messages sent from an authorized user using the CP WARNING command are received at your display. ON specifies that warning messages will be displayed. If OFF is specified, no warning messages will be displayed. IUCV specifies that warning messages are passed to the virtual machine via IUCV if a connection to the message system service exists. If no IUCV connection exists, the warning message will be handled as if WNG was set ON.

EMSG { **ON** }
 { **OFF** }
 { **CODE** }
 { **TEXT** }
 { **IUCV** }

indicates how error messages will be edited and displayed. ON indicates that both the error code and the message text will be displayed. OFF indicates that no error messages will be displayed. CODE indicates that only the error code will be displayed. TEXT indicates that only the error message text will be displayed. IUCV indicates that both the error code and error message text are passed to the virtual machine via IUCV if a connection to the message system service exists. If no IUCV connection exists, the error message will be handled as if EMSG was set ON.

ACNT OFF

appears for compatibility purposes.

RUN { **ON** }
 { **OFF** }

indicates whether your virtual machine will begin to run after you enter something from your terminal, and also indicates whether it runs when a CP READ status is showing. ON indicates that your virtual machine runs when CP READ status is showing and will switch to RUNNING following the entry. OFF specifies that your virtual machine does not run when CP READ status is showing, and will remain in CP READ status following the entry.

LINEDIT { **ON** }
 { **OFF** }

indicates whether line editing functions and symbols are to be used to edit virtual console input. ON indicates that the system line editing functions will be used. OFF indicates that no character or line editing will be used for the virtual machine operating system.

TIMER { **ON** }
 { **OFF** }
 { **REAL** }

indicates whether the virtual interval timer will be updated for System/370 mode virtual machines. **ON** specifies that the virtual interval timer will be updated during virtual processor run time. **OFF** specifies that the virtual interval timer will not be updated. **REAL** specifies that the virtual interval timer will be updated during virtual processor run time and also during virtual wait time.

ISAM OFF

appears for compatibility purposes.

ECMODE ON

appears for compatibility purposes.

ASSIST OFF

appears for compatibility purposes.

PAGEX { **ON** }
 { **OFF** }

indicates whether the pseudo page fault facility is activated for your virtual machine. **ON** specifies the facility is activated. **OFF** specifies the facility is deactivated.

AUTOPOLL { **ON** }
 { **OFF** }

indicates whether CP or your virtual machine detects changes to virtual autopolling channel programs. **ON** specifies that your virtual machine must notify CP that a change was made to a virtual autopolling channel program. **OFF** specifies that CP tests for changes to virtual autopolling channel programs.

IMSG { **ON** }
 { **OFF** }
 { **IUCV** }

indicates whether informational messages associated with certain CP commands will be displayed. **ON** specifies that informational responses will be displayed. **OFF** specifies that the responses will not be displayed. **IUCV** specifies that the informational responses are passed to the virtual machine via IUCV if a connection to the message system service exists. If no IUCV connection exists, the responses will be handled as if **IMSG** were set **ON**.

SMSG { **ON** }
 { **OFF** }
 { **IUCV** }

specifies whether a virtual machine is receiving special messages or not. **OFF** indicates the virtual machine cannot receive special messages. **ON** indicates the virtual machine can receive special messages. **IUCV** indicates the special messages are passed to the virtual machine via IUCV if a connection to the message system service exists. If no IUCV connection exists, the special messages will be handled as if **SMSG** was set **OFF**.

NOTRAN { **ON** }
 { **OFF** }

indicates whether virtual I/O from the issuing virtual machine will bypass CP CCW translation. **OFF** specifies that CP is to translate the CCWs. **ON** specifies that virtual I/O is to bypass CP CCW translation except under the following conditions:

- Tracing I/O to the device

- Tracing CCWs for the device
- CP has sense data pending for the device
- The I/O device is not dedicated to the virtual = real (V = R) machine, unless it is a “full-pack” mini-disk, and the channel program begins with valid SEEK, Set-File-Mask CCWs.

NOTRAN ON may only be in effect for the preferred virtual machine.

VMSAVE OFF

appears for compatibility purposes.

370E OFF

appears for compatibility purposes.

STBYPASS OFF

appears for compatibility purposes.

STMULTI OFF

appears for compatibility purposes.

AFFINITY NONE

appears for compatibility purposes.

**MIH { ON }
{ OFF }**

ON specifies that CP will handle any detected missing interrupt conditions.

OFF specifies that CP will **not** handle any detected missing interrupt conditions.

**VMCONIO { OFF }
{ IUCV }**

controls the destination of any virtual machine generated output. This includes any CMS error messages and any linemode I/O done by the virtual machine to the virtual console. OFF indicates that output is displayed at the terminal. IUCV indicates that output is sent to the virtual machine via IUCV if a connection to the message system service exists. If no IUCV connection exists, data will be handled as if VMCONIO was set to OFF.

**CPCONIO { OFF }
{ IUCV }**

controls the destination of any CP generated output. OFF indicates that the output is displayed at the terminal. IUCV indicates that the output is sent to the virtual machine via IUCV if a connection to the message system service exists. If no IUCV connection exists, data will be handled as if CPCONIO was set to OFF.

SVCACCL OFF

appears for compatibility purposes.

**CONCEAL { ON }
{ OFF }**

indicates whether the virtual machine is running in a protected application environment. ON indicates that certain error conditions and virtual machine state changes detected by CP will result in an automatic re-IPL of the virtual machine. OFF indicates that the virtual machine will stay in whatever environment it was in when the error occurred.

**MACHINE { 370 }
{ XA }**

indicates whether your virtual configuration is operating in System/370 mode or 370-XA mode.

SVC76 { **CP** }
 { **VM** }

indicates how SVC 76 instructions issued by a virtual machine will be processed by CP. CP specifies that the virtual machine's error records are recorded by CP error recording facilities, if the virtual machine is running in supervisor state. VM specifies that no errors are recorded and all SVC 76 operations are processed by the virtual machine.

NOPDATA { **ON** }
 { **OFF** }

controls whether NOP CCWs are simulated as data transfer CCWs for spooled punches and printers of your virtual machine. ON indicates that for more than one byte of data, CP attempts to transfer the data addressed by a NOP CCW to the spool file. OFF indicates the CP simulates the NOP the same way a real device and channel would.

IOASSIST { **ON** }
 { **OFF** }

indicates whether the start interpretive-execution assist is in effect for the preferred virtual machine.

CCWTRAN { **ON** }
 { **OFF** }

indicates whether virtual I/O from the issuing virtual machine will bypass CP CCW translation. OFF specifies that CP is to translate the CCWs. OFF specifies that virtual I/O is to bypass CP CCW translation except under the following conditions:

- When tracing I/O to the device
- When tracing CCWs for the device
- When CP has sense data pending for the device
- When the I/O device is not dedicated to the V=R machine, unless it is a full-pack minidisk and the channel program begins with valid SEEK, Set-File-Mask CCWs.

CCWTRAN OFF may be in effect only for the preferred virtual machine.

Migration Notes

VM/SP HPO: The VM/XA SP response provides XA-only settings at the end.

QUERY SHARE

Privilege Class: A,E

Use the QUERY SHARE command to display the type and intensity of a user's share of the system.

Query	SHARE userid
-------	--------------

where:

userid

is the userid of the virtual machine whose share is to be displayed.

Usage Notes

None.

Responses

USER userid: ABSOLUTE SHARE=nnn% [; m DEDICATED CPU[S]]

or

USER userid: RELATIVE SHARE=nnnn [; m DEDICATED CPU[S]]

appears to the command issuer.

where:

ABSOLUTE SHARE = nnn%

appears for users with an ABSOLUTE SHARE. nnn% specifies the percent of system resources this user is allowed access to.

RELATIVE SHARE = nnnnn

appears for users with a RELATIVE SHARE. nnnnn specifies the relative priority of access to system resources, when compared to all other users with RELATIVE shares.

m DEDICATED CPU[S]

specifies the number of virtual CPUs this user has which have a real processor dedicated to them. This string does not appear if the user has no dedicated processors (for example, when m = 0).

Migration Notes

VM/XA SF: The "QUERY SRM USER userid" command, which displayed SHARE information, is replaced by this command.

QUERY SRM

Privilege Class: A,E

Use the QUERY SRM (system resource manager) command to display system-wide parameters used by the scheduler to prioritize access to system resources.

Query	SRM	$\left. \begin{array}{l} \text{ALL} \\ \text{IABias} \\ \text{DSPSlice} \\ \text{LDUBuf} \\ \text{STORBuf} \\ \text{DSPBuf} \end{array} \right\}$
-------	-----	---

where:

ALL

causes a summary of the current system-wide settings. This will display the IABIAS, LDUBUF, STORBUF, DSPBUF, and DSPSLICE settings.

IABIAS

specifies that the current settings for the interactive bias “intensity” and “duration” should be displayed. For a description of the interactive bias facility, see the SET SRM IABIAS command.

LDUBUF

displays the current LDUBUF (loading user buffer) percentages used by the scheduler to partition access to the paging configuration based on transaction classes. See the SET SRM LDUBUF command for a description of the percentage values.

STORBUF

displays the current STORBUF (storage buffer) percentages used by the scheduler to partition storage based on transaction classes. See the SET SRM STORBUF command for a description of the percentage values and the transaction classes represented.

DSPBUF

displays the current DSPBUF (dispatch buffer) values used by the scheduler to limit the number of users allowed in the dispatch list for each transaction class. See the SET SRM DSPBUF command for a description of the values.

DSPSLICE

displays the current setting of the dispatching timeslice in milliseconds.

Usage Notes

For more detailed information about the allocation of storage resources to users, refer to the SET SRM command. Also refer to *VM/XA SP Real System Operation*.

Response 1

The response to QUERY SRM ALL is as follows:

```
IABIAS : INTENSITY=mmm%; DURATION=nnn
LDUBUF : Q1=ddd% Q2=eee% Q3=fff%
STORBUF: Q1=aaa% Q2=bbb% Q3=ccc%
DSPBUF : Q1=iiii Q2=jjjj Q3=kkkk
DISPATCHING MINOR TIMESLICE = xxx MS
```

Response 2

The response to QUERY SRM IABIAS is:
IABIAS : INTENSITY=mmm%; DURATION=nnn

Response 3

The response to QUERY SRM LDUBUF is:
LDUBUF : Q1=ddd% Q2=eee% Q3=fff%

Response 4

The response to QUERY SRM STORBUF is:
STORBUF: Q1=aaa% Q2=bbb% Q3=ccc%

Response 5

The response to QUERY SRM DSPBUF is:
DSPBUF : Q1=iiii Q2=jjjj Q3=kkkk

Response 6

The response to QUERY SRM DSPSLICE is:
DISPATCHING MINOR TIMESLICE = xxx MS

Migration Notes

VM/SP HPO: VM/XA SP provides different responses because it has a different scheduler architecture and does not support page migration.

VM/XA SF: In VM/XA SP you must use the QUERY SHARE command to see a summary of scheduler settings for an individual userid.

QUERY (real) STORAGE

Privilege Class: B

Use QUERY (real) STORAGE to display the size of real storage.

Query	STORage
-------	---------

where:

STORAGE

displays the size of real storage.

Usage Notes

The size of real storage is calculated as the minimum of storage at system generation time plus configured real storage.

Responses

$$\text{STORAGE} = \left\{ \begin{array}{l} \text{nnnnnnnK} \\ \text{nnnM} \end{array} \right\}$$

where:

nnnnnnnK

is the number of kilobytes of real storage.

nnnnM

is the number of megabytes of real storage.

Note: Output will be in megabytes (M) only if on a full megabyte boundary; otherwise, the output will be given in kilobytes (K).

Migration Notes

VM/XA SF: The VM/XA SP response provides only the amount of real storage. You must use the QUERY FRAMES command to receive other information regarding real storage usage.

valid

is the volume label.

Response 2

userid vdev mode[, ... up to 3 responses per line]

You get this response when you specify a real device, where:

userid

identifies the users who have links to minidisks on the real device.

vdev

is the virtual address by which the user refers to the minidisk.

mode

is the type of access the user has, either R/O or R/W, or nnnn for the number of cylinders of temporary disk space allocated.

Response 3

DASD MDISKS NOT FOUND

indicates that no active minidisks have been defined on this DASD volume.

Response 4

DASD TDSKS NOT FOUND

designates that no temporary disk space was defined on the indicated volume.

Migration Notes

VM/SP HPO: VM/XA SP supports additional options: ALL and TDISK.

VM/XA SF

1. The VM/XA SP does not provide what used to be the first line of the VM/XA SF response.
2. The VM/XA SP response uses nnnn for TDISK cylinder count instead of Tnnnn as in VM/XA SF.

QUERY TAG

Response 2

tagtext

- or -

(TAG BLANK)

displays the tag information associated with a specified spool file.

where:

tagtext

is the tag information associated with the file. This tag information may be up to 136 characters long.

(TAG BLANK)

is displayed when the tag is all blanks.

Response 3

type {vdev } TAG NOT SET

is the response if:

- You have not specified tag information for a device
- You have explicitly set the tag field to blanks.

where:

type

is PRT, PUN, or CONS.

vdev

is the virtual device number.

Migration Notes

VM/XA SF: The VM/XA SP response provides the tag text or TAG BLANK on the second line.

QUERY TDISK

Privilege Class: B

Use QUERY TDISK to display all currently defined temporary disk space (TDSK) assigned to virtual machine users from all available system-owned volumes.

Query	{TDisk} {TDsk}
-------	-------------------

where:

TDISK

TDSK

requests the display of all currently defined temporary disk space (TDSK) assigned to virtual machine users from all available system-owned volumes.

Usage Notes

None.

Response 1

userid vdev nnnn [, ...] up to 4 responses per line

where:

userid

identifies the users who have links to minidisks on rdev.

vdev

is the virtual address by which the user refers to the minidisk.

nnnn

is the number of cylinders of temporary disk space allocated.

Response 2

DASD TDSKS NOT FOUND

indicates that no active temporary DASD have been defined.

CHARDEL { **char** }
 { **OFF** }

displays the logical character delete symbol in effect for your virtual machine (char), or indicates that no logical character delete symbol is allowed (OFF).

ESCAPE { **char** }
 { **OFF** }

displays the logical escape character in effect for your virtual machine (char), or indicates that no logical escape character is allowed (OFF).

TABCHAR { **ON** }
 { **OFF** }
 { **char** }

ON means that the tab character used by the system is the default logical tab character. OFF means that you have no logical tab character defined. char is the logical tab character currently defined.

LINESIZE { **nnn** }
 { **OFF** }

specifies the maximum allowable line length for terminal output that is in effect for your virtual machine. OFF specifies that the terminal output line length is not adjusted by CP.

ATTN { **ON** }
 { **OFF** }

indicates whether the attention signal is to be reflected on keyboard or printer terminals. If ON, the printer reflects it as an exclamation point (!) followed by a carriage return.

APL { **ON** }
 { **OFF** }

indicates whether CP is to use the APL terminal translation tables or the normal translation tables. If APL is ON, CP uses the APL translation tables.

TEXT { **ON** }
 { **OFF** }

indicates whether CP is to use the TEXT terminal translation tables or the normal terminal translation tables. If TEXT is ON, CP uses the TEXT translation tables.

MODE { **CP** }
 { **VM** }

indicates whether one or more attentions force your virtual machine into the CP environment, or whether one or more attentions are reflected to your virtual machine. If MODE is CP, one attention forces your virtual machine into the CP environment.

HILIGHT { **ON** }
 { **OFF** }

ON means CP highlights the command redisplay line. OFF means CP displays the command redisplay line at normal intensity.

CONMODE { **3215** }
 { **3270** }
 { **NONE** }

indicates the console mode of your virtual console. 3215 indicates that VM/XA SP provides 3215 CCW processing, and 3270 indicates that VM/XA SP provides 3270 CCW processing. NONE indicates that your virtual machine does not currently have a console defined.

BREAKIN { **IMMED** }
 { **GUESTCTL** }

allows you to control break-in by CP for full-screen guests. The **GUESTCTL** option specifies that, while in full-screen mode, messages from CP are displayed only when you hit the **BRKKEY** key. If you specify **GUESTCTL**, CP break-in still occurs if you request a CP function or if a priority message is pending for your virtual machine. If you specify **IMMED**, which is the default, break-in by CP takes place on any full-screen **READ** and full-screen **WRITE** that has an inactive keyboard. Break-in also takes place when you hit the **BRKKEY**.

BRKKEY (**PA1**)
 (**PA2**)
 (**PFnn**)
 (**CLEAR**)
 (**NONE**)

displays the key that controls break-in by CP for full-screen guests (to allow you to return to CP mode). **NONE** indicates that there is no **BRKKEY** defined or it is disabled.

SCRNSAVE { **OFF** }

is included for compatibility reasons with its default setting.

TYPE { **3101** }
 { **TTY** }

3101 identifies the terminal as a display device, and **3101** functions are enabled. **TTY** identifies the terminal as a typewriter terminal, and **3101** functions are not in effect.

PROMPT { **VM** }
 { **TTY** }

VM indicates that **VM** enables the line feed prompting sequence. **TTY** indicates that **VM** enables the normal **TTY** prompting sequence of a period in column 1 and the cursor in column 2.

SCROLL { **nnn** }
 { **CONT** }

nnn specifies the number of lines displayed before ******MORE****** status is displayed. **CONT** specifies continuous scrolling without interruption.

CNTL { **SYS** }
 { **USR** }

SYS specifies that CP inserts **TTY** line control characters for output directed to a **TTY** terminal. **USR** specifies that the application program inserts **TTY** line control characters for output directed to a **TTY** terminal.

ASCITBL { **VM1** }
 { **VM2** }

VM1 specifies the use of the translation table based on the **TTY ANSI X3.4 1977** standard. **VM2** specifies the use of the translation table based on the **TTY ANSI X3.26 1980** standard.

Migration Notes

VM/SP HPO: The **VM/XA SP** response provides **370-XA** only settings at the end.

VM/XA SF: **VM/XA SP** uses the **VM/SP HPO** format with **370-XA** only settings at the end.

QUERY TIME

Privilege Class: G

Use **QUERY TIME** to display the current time, time zone, weekday, date, connect, and processor time for the current session.

Query	Time
-------	------

where:

TIME

requests a display of the current time.

Usage Notes

1. Vector time is a subset of CPU time; that is, using a second of vector CPU time results in incrementing both the CPU time and the Vector Facility time counts. This applies to both virtual (guest) and total time. The reason for this is that using the VF implies using the CPU; all VF usage arises from executing instructions on that CPU.
2. If the wait-state interpretation capability is available on the real processor, it is used for dedicated virtual CPUs, unless the guest is in System/370 mode with **SET TIMER ON**. When the wait-state interpretation capability is used, the guest virtual machine remains dispatched even when it enters a PSW enabled-wait state. Time spent dispatched in the wait state is included in the virtual and total CPU time reported for the guest.

Responses

TIME IS hh:mm:ss zone weekday mm/dd/yy

The current real clock time in hours:minutes:seconds, the time zone (for example, EST), the day of the week, and the calendar date (month/day/year) are displayed.

CONNECT= hh:mm:ss VIRTCPU= mmm:ss.hs TOTCPU= mmm:ss.hs

The time spent in the current session is displayed. If you disconnect, the time does not revert to zero, you will see the actual clock time spent since logon.

where:

CONNECT = hh:mm:ss

is the actual clock time spent in the current session in hours:minutes:seconds.

VIRTCPU = mmm:ss.hs

is the virtual processor time used in the current session in minutes:seconds.hundredths-of-seconds.

TOTCPU = mmm:ss.hs

is the total processor time (virtual and overhead) used in the current session in minutes:seconds.hundredths-of-seconds.

QUERY TRACE

Privilege Class: G

Use QUERY TRACE to display the current TRACE settings.

Query	TRace	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> ACTIVE ALL name SETS RETURNS </div>
-------	-------	---

where:

TRACE

indicates that you want to display the current TRACE settings. QUERY PER will be accepted as a synonym for QUERY TRACE.

ACTIVE

is the default option and displays attributes of all traps within the active trace set.

ALL

displays attributes of all traps within all trace sets.

name

displays attributes of all traps within the trace set specified by the 1- to 8-character name.

SETS

displays all trace set names. No trap attributes are displayed.

RETURNS

displays TRACE CALL return path. No trap attributes are displayed.

Response 1

TRACE COUNT IS nnnnnnnn

precedes the display of trace traps if TRACE COUNT is active.

Response 2

QUERY TRACE SETS Response Format

SETS: name1, name2, ...

where:

name1, name2,

are names of trace sets.

Response 3

QUERY TRACE RETURNS Response Format

RETURNS: name1, name2, ...

where:

name1, name2,

are names of trace sets in the trace return path.

Response 4

**QUERY TRACE ALL, QUERY TRACE ACTIVE, and QUERY TRACE name
Response Format**

The current status of all defined trace sets, of trace traps within the active trace set, or a specifically identified trace set is displayed by QUERY TRACE. Each trace trap has a specific format as shown below where “ident” is the trace trap identifier for each trap within the trace set.

A common header exists for these responses:

```
NAME setname [ACTIVE
              [ACTIVE + SUSPEND]
```

When the ACTIVE operand is specified, it appears at the top of the list of the active trace set.

Instruction Trap Format (DATA option not specified)

```
ident INSTR PSWA hexloc[-hexloc]
      {TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER ]
      {NOTERM} {NOPRINT} {NORUN} [NODAT] [PROB ]
      SKIP n PASS n STOP n STEP n
      CMD ...CP command...
```

Instruction Trap Format (DATA option specified)

```
ident INSTR PSWA hexloc[-hexloc]
      DATA data-string
      {TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER ]
      {NOTERM} {NOPRINT} {NORUN} [NODAT] [PROB ]
      SKIP n PASS n STOP n STEP n
      CMD ...CP command...
```

Store Trap Format

```
ident STORE FROM hexloc[-hexloc] INTO hexloc[-hexloc]
      DATA hexstring
      MASK hexstring
      {TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER ]
      {NOTERM} {NOPRINT} {NORUN} [NODAT] [PROB ]
      SKIP n PASS n STOP n STEP n
      CMD ...CP command...
```

Branch Trap Format

```
ident BRANCH FROM hexloc[-hexloc] INTO hexloc[-hexloc]
      {TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER ]
      {NOTERM} {NOPRINT} {NORUN} [NODAT] [PROB ]
      SKIP n PASS n STOP n STEP n
      CMD ...CP command...
```

Register Alteration Trap Format

```
ident GREG FROM hexloc[-hexloc] DATA hexloc[-hexloc ]
      REG s n1 n2... MASK hexstring
      {TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER ]
      {NOTERM} {NOPRINT} {NORUN} [NODAT] [PROB ]
      SKIP n PASS n STOP n STEP n
      CMD ...CP command...
```

QUERY TRACE

I/O Trap Format

```
ident  typename FROM hexloc[-hexloc]
      HEX n[-m]
      {TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER] [INST]
      {NOTERM} {NOPRINT} {NORUN} {NOSIM} [NODAT] [PROB] [INT]
      SKIP n PASS n STOP n STEP n
      CMD ...CP command...
```

Mnemonic Trap Format

```
ident  typename FROM hexloc[-hexloc]
      {TERM } {PRINT } {RUN } {SIM } [DAT ] [SUPER]
      {NOTERM} {NOPRINT} {NORUN} {NOSIM} [NODAT] [PROB]
      SKIP n PASS n STOP n STEP n
      CMD ...CP command...
```

TRACE MCH Format

```
ident  typename FROM hexloc[-hexloc]
      {TERM } {PRINT } {RUN } {SIM }
      {NOTERM} {NOPRINT} {NORUN} {NOSIM}
      SKIP n PASS n STOP n STEP n
      CMD ...CP command
```

External/Program

```
ident  typename FROM hexloc[-hexloc]
      {EXT }
      {PROG }
      {MC }
      {DIAG }
      {SVC }
      {TERM } {PRINT } {RUN } {SIM }
      {NOTERM} {NOPRINT} {NORUN} {NOSIM}
      SKIP n PASS n STOP n STEP n
      CMD ...CP command
```

Response 6

No trace sets defined

is the response for any form of the QUERY TRACE command if no trace sets have been defined.

Response 7

No trace calls active

is the response for QUERY TRACE RETURNS if no trace calls are active; therefore, there are no return paths.

Migration Notes

VM/SP HPO

1. QUERY PER is accepted as a synonym for QUERY TRACE.
2. The VM/XA SP response returns the setting of all TRACE options instead of those explicitly entered as HPO.

QUERY TRFILES

Privilege Class: A,C,D,E,G

Use the QUERY TRFILES command to display detailed information about one or more system trace files or to display the count of system trace files belonging to a user or to all users. System trace files are system data files created by the TRSAVE and TRSOURCE commands

Query	TRFiles [SYSTEM] [ALL NAME fn] [COUNT]
	[userid] [ALL NAME fn spoolid] [COUNT]
	[*]

where:

TRFILES

specifies that information should be displayed about system trace files.

SYSTEM

specifies that the files should be queried regardless of who owns or originated them. The SYSTEM option is not allowed with the *spoolid* option. SYSTEM is the default for class A, C, D, and E users.

userid

*

specifies the owner or originator of the files to be queried. The file owner is specified by TO *userid* on the TRSAVE command. When TO is not specified, the owner is the command issuer. For trace IDs specified by the TRSOURCE ID *xxx* TYPE GT FOR *userid* command, the file originator is specified by FOR *userid*. Otherwise, it is SYSTEM.

If you are a class G user, you may query only the files that you own or originate. If you, as a class G user, use the *userid* operand to specify a *userid* other than your own *userid*, you receive information on all spool files that you originated and that are owned by the specified *userid*. If you use *, or you default to *, you receive information on the files you currently own.

ALL

queries all system trace files. This is the default.

NAME fn

specifies the filename of the files to be queried. This is the filename specified on the command TRSAVE..ON DASD NAME *fn*.. when the trace was started. More than one file can exist with the same filename for a given owner.

spoolid

specifies the spool file identification number of the file to be queried. This is invalid with the SYSTEM option.

COUNT

displays only the number of system trace files that match the criteria specified on the command.

Usage Notes

None.

Response 1

The following is an example of a response to QUERY TRFILES SYSTEM:

OWNERID	FILE	TYPE	CL	RECS	DATE	TIME	FILENAME	FILETYPE	ORIGINID
MAINT	2240	TRF	A	0200	08/28	11:21:10	CPTRACE	CP	SYSTEM
MAINT	2248	TRF	A	0010	08/28	11:22:02	CPTRACE	CP	SYSTEM
MAINT	8721	TRF	A	1197	09/04	07:05:30	LOSTLINE	VM	RSCS
SYSPGMR	6800	TRF	W	0300	09/05	10:33:04	GTRACE	VMG	SYSTEM
SYSPGMR	6805	TRF	W	0300	09/05	10:43:50	GTRACE	VMG	SYSTEM
SYSPGMR	6820	TRF	W	0170	OPEN-TRSAVE		GTRACE	VMG	SYSTEM
OPERATOR	3234	TRF	A	0300	09/05	10:43:06	DEV200	IO	SYSTEM
OPERATOR	3235	TRF	A	0170	09/05	10:45:70	MODTSA	DATA	SYSTEM

where:

OWNERID

identifies the file owner. The file owner is specified by TO *userid* when the TRSAVE command is issued. If TO is not specified, the owner is the TRSAVE command issuer.

FILE

is the spool file identification number.

TYPE

indicates the type of system data file. This is TRF for system trace files.

CL

is the file class. If the class of the file is A, the file can be processed. If the class of the file is W, it is in the set of files to which an active trace is writing.

RECS

is the number of 4Kb pages in the file.

DATE

is the month and day that the file was closed. If the file is open for reading or writing, or the file is being processed by the TRACERED command, or the file has been opened by DIAGNOSE Code X'E0', OPEN-TRSAVE is displayed instead of a date and time.

TIME

is the time the file was closed.

FILENAME

is the file name assigned by the issuer of the TRSAVE command.

FILETYPE

is the file type assigned to the file. It indicates the type of system trace file as follows:

CP	CP system trace tables
VM	Trace data for a userid
VMG	Trace data for a virtual machine group
IO	Trace data from IO trace
DATA	Trace data from DATA trace.

ORIGINID

is the originator of the trace file. For traces defined by TRSOURCE ... TYPE GT FOR USER ... commands, the origin ID is the userid specified by FOR USER. Otherwise, the origin ID is SYSTEM.

Response 2

If no files as specified by the command line exist, the following response is given:

FILES: NO TRF

Response 3

If the COUNT option is specified on the QUERY TRFILES command, only the number of system trace files that match the criteria specified is displayed. The following is the response:

FILES: $\left. \begin{array}{l} \text{nnnn} \\ \text{NO} \end{array} \right\}$ TRF

where:

nnnn

is the number of system trace files. If no files are found that match the selection criteria, then NO replaces nnnn.

QUERY TRSAVE

Privilege Class: A,C,E,G

Use the QUERY TRSAVE command to display the destination of traces defined by TRSOURCE or to display the current status of recording for CP system tracing (traces controlled by the SET CPTRACE command).

Query	TRSAve [FOR] [CP [ID] traceid ALL]
-------	--

where:

CP

displays the status of TRSAVE activity for CP system tracing. This option is allowed only for class A, C, and E users.

[ID] traceid

displays the status of TRSAVE activity for the given trace ID.

ALL

displays the status of all active TRSAVE functions, or indicates that there are no active functions.

Usage Notes

1. For class G users, QUERY TRSAVE displays the destination of data derived from the trace or traces for that user's virtual machine.
2. For class A, C, or E users, QUERY TRSAVE displays the status of any TRSAVE activity.

Response 1

If tracing is done to DASD:

Source	Filename	Receiver	Keep	Size
CPTRACE	filename	userid	n	mmm
traceid	filename	userid	n	mmm

where:

Source

indicates the source of this trace. CPTRACE identifies tracing activity resulting from the TRSAVE FOR CP ON command. Traceid identifies tracing activity resulting from a TRSOURCE ID *traceid* command.

Filename

is the filename specified or defaulted to on the TRSAVE command.

Receiver

is the userid who was specified, or defaulted to, in "TO userid" on the TRSAVE command. This userid is the only user authorized to read this system trace file.

Keep

indicates the maximum number of files that CP keeps each time this trace is active.

Size

indicates the maximum size in number of 4Kb pages for each of the files created when this trace is active.

When ALL is specified, the above information is displayed for all existing trace IDs and for CPTRACE when it is active.

Response 2

Traceid *traceid* is not defined

is displayed when a class A, C, or E user issues QUERY TRSAVE ID *traceid* for a trace ID that has not been defined.

Response 3

Traceid *traceid* is not tracing for your virtual machine

is displayed when a class G user issues QUERY TRSAVE ID *traceid* for a trace ID that has not been defined or when the user is not authorized to view information for that particular trace ID.

Response 4

No traceids defined

is displayed when the user issues QUERY TRSAVE ALL but no trace IDs have been defined.

Response 5

No traceids defined for your virtual machine

is displayed when a class G user issues QUERY TRSAVE ALL but no trace IDs have been defined for which the user is authorized to view information.

Response 6

If CP system trace data is being written to tape, the system responds with:

TRSAVE for CP on drive rdev1 [rdev2] mode nnnn $\left\{ \begin{array}{l} \text{rewind} \\ \text{run} \end{array} \right\}$

where:

rdev1 [rdev2]

are the real device addresses of the tape drives being used for the TRSAVE function. If only one tape drive is being used, rdev2 is not shown.

nnnn

is the density. Mode values can be 800, 1600, 6250, or 38K (for 3480 tape drives only).

REWIND

RUN

indicates that the tape is to be rewound (REWIND) or both rewound and unloaded (RUN) when the tape fills.

QUERY TRSAVE

Response 7

TRSAVE inactive for CP

is displayed if TRSAVE FOR CP is not active.

Response 8

TRSAVE pending active for CP

indicates that TRSAVE FOR CP is neither active nor inactive, but is in a transition state between the two.

QUERY TRSOURCE

Privilege Class: A,C,E,G

Use the QUERY TRSOURCE command to display the current status of the various traces that have been defined using the TRSOURCE command.

Query	TRSource	<pre> ALL ID traceid SET traceset TYPE { IO } { GT } { DATA } { NONE } USER { userid } { * } </pre>
-------	----------	---

where:

ALL

requests the status of all defined traces. ALL is the default.

ID traceid

requests the status of the specified trace ID.

SET traceset

requests the status of all defined traces of the specified set.

TYPE { IO }
 { GT }
 { DATA }
 { NONE }

specifies the type of trace for which information is displayed.

IO

requests the status of all traces of type IO.

GT

requests the status of all traces of type GT.

DATA

requests the status of all traces of type DATA.

NONE

requests the status of all traces of unspecified type.

USER { userid }
 { * }

requests the status of all traces of type GT that are defined for the specified user's virtual machine. You can use * to specify your own virtual machine.

Usage Notes

1. A trace of type NONE occurs when a trace has been defined by the TRSOURCE or TRSAVE command, but the TYPE xxx parameter of the TRSOURCE command is not specified.
2. The class G user may not specify any parameters. The response to the class G user is the same as the response to QUERY TRSOURCE USER *.

QUERY TRSOURCE

- An IO trace may be defined for a nonexistent device (that is, a device that has no RDEVICE entry in module HCPRIO). If a trace with *only* such devices is enabled, message HCPTXI477I is issued:

Traceid (TRACE ID) not enabled; incomplete definition

Note, however, that a trace with a nonexistent device and a device defined in HCPRIO (whether the latter be online or offline) *can* be enabled, and the response to QUERY TRSOURCE will indicate that a trace for the nonexistent device is activated.

Responses for Undefined Traces

Response 1

No traces are defined

is the response to QUERY TRSOURCE ALL if no traces are defined.

Response 2

Traceid *traceid* is not defined

is the response to QUERY TRSOURCE ID *traceid* if the specified trace ID is not defined.

Response 3

Traceset *traceset* is not defined

is the response to QUERY TRSOURCE SET *traceset* if no traces are defined for the specified trace set.

Response 4

No GT traces are defined for user *userid*

is the response to QUERY TRSOURCE USER *userid* if no GT traces are defined for the specified user's virtual machine.

Response 5

No trace was defined for type *type*

is the response to QUERY TRSOURCE TYPE *tracetype* if no traces are defined for the specified trace type.

Responses for Defined Traces

The format of the responses depends upon the trace types. The grouping of the responses for QUERY TRSOURCE ALL is by trace set.

Response 6

SET *setid*

is the response that heads each group of traces included in a set. *setid* is the name for the set. When no set name has been defined, the set name is NULL.

Response 7

ID	TYPE	SET	STATUS	DEVICES	USER	IODATA
traceid	IO	setname	{ENABLED } {DISABLED }	rdev(s)	userid	nnnn

is displayed for traces of type IO.

where:

ID

indicates the traceid that is the name of the trace, as specified on the ID *xxx* parameter of the TRSOURCE ..TYPE IO .. command.

TYPE

IO indicates the type of trace.

SET

indicates the name used to group traces into a logically related set, as specified on SET *xxx* parameter of the TRSOURCE ID *traceid* .. command.

STATUS

is either ENABLED or DISABLED. ENABLED indicates the trace is active. DISABLED indicates the defined trace is inactive.

DEVICES

indicates the device number (rdev1) or range (rdev1-rdev2) of device numbers whose I/O activity should be traced. It is the value specified on the DEVICE *xxx* parameter of the TRSOURCE ID *traceid* .. command.

USER

indicates which user's I/O should be traced. It is specified on the USER *xxx* parameter of the TRSOURCE ID *traceid* .. command. ALL indicates that all I/O to the specified devices is traced for all users.

IODATA

indicates how many bytes of the transferred data are being traced for the defined IO trace.

Response 8

ID	TYPE	SET	STATUS	ORIGINID	INCLUDED
traceid	GT	setname	{ENABLED } {DISABLED }	sysname userid useridb	{userida } {useridb }

ID	TYPE	SET	STATUS	ORIGINID	EXCLUDED
traceid	GT	setname	{ENABLED } {DISABLED }	sysname userid useridb	userid useridy

ID	TYPE	SET	STATUS	ORIGINID
traceid	GT	setname	{ENABLED } {DISABLED }	userid

is displayed for traces of type GT.

QUERY TRSOURCE

where:

ID

indicates the traceid as specified on the ID *xxx* parameter of the TRSOURCE ..TYPE GT .. command.

TYPE

indicates the type of the trace as GT.

SET

is the name used to group traces into a logically related set, as specified on the SET *xxx* parameter of the TRSOURCE ID *traceid* .. command.

STATUS

is either ENABLED or DISABLED. ENABLED indicates the trace is active. DISABLED indicates the defined trace is inactive.

ORIGINID

is the name of a virtual machine group if TRSOURCE .. FOR VMGROUP *sysname* was entered to define the GT trace ID. When tracing is active for a virtual machine group, the userids of the group members for which tracing is active are displayed as shown. When TYPE GT tracing is being done for a single virtual machine (the command TRSOURCE .. FOR USER *userid* was entered), the origin ID is the name of the virtual machine.

INCLUDED

displays a list of the virtual machines that are included in the virtual machine group trace. These virtual machines may or may not be logged on.

ALL indicates that every virtual machine that connects to the virtual machine group is included in the group trace.

NONE indicates that no virtual machines are included in the group trace.

EXCLUDED

displays a list of the virtual machines that are excluded from the virtual machine group trace. These virtual machines may or may not be logged on.

Either an included list or an excluded list is displayed for a virtual machine group.

Response 9

ID	TYPE	SET	STATUS	LOC	INSTRUCTION
traceid	DATA	setname	{ENABLED } {DISABLED }	{ nnnnnn } { modname+disp }	string...

DATALINKS

DATALINK 001
DATALINK 002
DATALINK XXX

- or -

DATALINKS

NO DATALINKS DEFINED

is displayed for traces of type DATA.

where:

ID

is the name of the trace, as specified on the ID *xxx* parameter of the TRSOURCE ..TYPE DATA .. command.

TYPE

DATA indicates the type of trace.

SET

is the name used to group traces into a logically related set, as specified on SET *xxx* parameter of the TRSOURCE ID *traceid* .. command.

STATUS

is either ENABLED or DISABLED. ENABLED indicates the trace is active. DISABLED indicates the defined trace is inactive.

LOC

displays nnnnnn to indicate the virtual address of the instruction which, when executed, triggers tracing information. modname + disp indicates the module name and displacement of the instruction which, when executed, triggers tracing information.

INSTRUCTION

is the instruction string located at the indicated location.

DATALINKS

defines what information is collected when the trace springs. If no data links were specified for this trace ID, NO DATALINKS DEFINED is displayed under the DATALINKS header line.

Response 10

ID	TYPE	SET	STATUS
traceid	NONE	setname	DISABLED

is displayed for traces of unspecified type, or type NONE.

where:

ID

is the name of the trace, as specified on the ID *xxx* parameter of the TRSOURCE command.

TYPE

displays NONE to indicate that the TYPE *xxx* parameter was not specified on the TRSOURCE ID *traceid* command.

SET

is the name used to group traces into a logically related set, as specified on the SET *xxx* parameter of the TRSOURCE ID *traceid* .. command.

STATUS

displays DISABLED to indicate that the trace is inactive.

QUERY UCR

Privilege Class: A,B,C

Use the QUERY UCR command to display detailed information about one or more UCR (user class restructure) files. This command is equivalent to the QUERY RDR/PRT/PUN command for spool files.

Query	UCR	[ALL NAME fn spoolid]	[COUNT]
-------	-----	-------------------------------	---------

where:

UCR

specifies that information should be displayed about UCR files.

ALL

allows you to query all UCR files. This is the default.

NAME fn

specifies the name of the file to be queried. This is the filename specified when the file was created. More than one file can exist with the same filename.

spoolid

is the spool file identification number of the file to be queried.

COUNT

displays only the number of UCR files that match the criteria specified on the command. No other information is provided.

Usage Notes

None.

Response 1

```
OWNERID  FILE TYPE CL RECS DATE  TIME      FILENAME FILETYPE ORIGINID
*UCR     1234 UCR  A  0001 mm/dd hh:mm:ss CLASS   UCR      originid
```

where:

OWNERID

indicates that the file is a UCR file owned by the system. *UCR is not a userid; it indicates on the queue on which the file resides.

FILE

is the spool file identification number.

TYPE

indicates that the file is a UCR file.

CL

is either A or I. Class A indicates that the file is active and usable. Class I indicates that this file will become active during the next system IPL.

RECS

is the number of 4Kb pages in the file.

DATE

is the month and day that the file was closed. If the file is being created, "OPEN-UCR" is displayed instead of a date and time.

TIME

is the time the file was closed in hours:minutes:seconds.

FILENAME

is the filename.

FILETYPE

is the filetype.

ORIGINID

is the userid of the originator of the file.

Response 2

If no files as specified by the command line exist, the following response is given:

FILES: NO UCR

Response 3

If the COUNT option is specified on the QUERY UCR command, only the number of user class restructure files that match the criteria specified is displayed. The following is the response:

FILES: { nnn } UCR
 { NO }

where:

nnnn

is the number of UCR files. If no files match the selection criteria, then NO replaces nnnn.

QUERY USERID

Privilege Class: G

Use QUERY USERID to display the userid and the system identifier of a user's display.

Query	{ USERID } { userid }
-------	--------------------------

where:

USERID

requests a display of the userid and system identifier (if applicable).

userid

requests the device number of the specified user's display.

Usage Notes

None.

Response 1

userid [AT sysid]

where:

USERID

is the userid of the current virtual machine.

AT sysid

is a 1- to 8-character name that identifies the VM system. The name is assigned by the system programmer. If no name has been assigned to the system being used, this is omitted.

Response 2

userid - { rdev }
 { DSC }
 { ldev }
 { luname }

where:

userid

is the userid you specified in the QUERY userid command line.

rdev

is the real device number of the display where the user is logged on.

DSC

is displayed if the user is disconnected from the system.

ldev

is the logical device number through which VM/XA SP accesses the display where the user is logged on.

luname

identifies an SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

QUERY USERS

Privilege Class: Any

Use QUERY USERS to display the number of logged-on users, the number of users dialed to other virtual machines, and, when a userid is specified, the device number of that user.

Query	Users [userid]
-------	----------------

where:

USERS

displays the number of logged-on and dialed users.

USERS userid

displays the userid and the device number of the user's display.

Usage Notes

1. A dialed display device is not available to CP because it is logically attached to a logged-on user and is a part of that user's virtual machine configuration. For more information, refer to the CP DIAL command.
2. QUERY NAMES and QUERY USERS can return different responses for the number of users logged on. The number of users in the process of logging on and logging off accounts for this difference.

Response 1

The response to QUERY USERS is:

nnnnn USERS, mmmmm DIALED, xxxxx NET

where:

nnnnn

is the total number of logged on users.

mmmmm

is the total number of users logically attached to virtual machines using the DIAL command.

xxxxx

is the total number of users logged on through a VTAM service machine in an SNA network. This number includes LOGON and DIAL connections.

Response 2

The response to QUERY USERS userid is:

$$\text{userid} - \left(\begin{array}{l} \text{rdev} \\ \text{DSC} \\ \text{ldev} \\ \text{luname} \end{array} \right)$$

where:

userid

is the userid you specified in the QUERY USERS command line.

QUERY USERS

rdev

is the real device number of the display where the user is logged on.

DSC

is displayed if the user is disconnected from the system.

ldev

is the logical device number through which VM/XA SP accesses the display where the user is logged on.

luname

identifies an SNA/CCS terminal controlled by a VTAM service machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

Migration Notes

VM/SP HPO: The VM/XA SP response has five digits for the number of users.

QUERY VECTOR

Privilege Class: A, B, C, E

Use the QUERY VECTOR command to display the status of installed Vector Facilities or the userids of Vector Facility users. A virtual machine is considered a Vector Facility user when it has used its virtual Vector Facility or Facilities at least once since the last LOGON or SYSTEM CLEAR command.

Query	VEctor [USERS]
-------	----------------

Usage Notes

A Vector Facility is permanently associated with a single CPU and can be used only by the associated CPU. The address of the Vector Facility is the same as that of the associated CPU.

Response 1

Response to QUERY VECTOR command:

```
VECTOR FACILITY nn {ONLINE }
                   {OFFLINE}
```

is presented for each real Vector Facility installed in the configuration.

where:

nn
is the real Vector Facility address.

Response 2

Response to QUERY VECTOR command:

```
NO VECTOR FACILITIES INSTALLED
```

is the response if no Vector Facilities are installed in the configuration.

Response 3

Response to QUERY VECTOR command:

```
NO VECTOR FACILITIES ONLINE
```

is received if there are Vector Facilities installed on the configuration but none of them are online.

Response 4

Response to QUERY VECTOR USERS command:

```
nnnnn VECTOR USERS

userid1 userid2 userid3 userid4 userid5
```

where:

nnnnn
represents the number of current real Vector Facility users.

QUERY VECTOR

useridn

represents the various real Vector Facility at that time.

Response 5

Response to QUERY VECTOR USERS command:

NO VECTOR USERS

is the response if there are no Vector Facility users.

Migration Notes

VM/SP HPO: The VM/XA SP response provides only userids and number of users; for more information, use the INDICATE VECTOR command.

QUERY VIRTUAL ALL

Privilege Class: G

Use QUERY VIRTUAL ALL to display the size of your virtual storage, the status of all devices in your virtual machine configuration, and the status of your virtual processors.

Query	Virtual [ALL]
-------	---------------

where:

VIRTUAL [ALL]

requests a display of the status of all devices in your virtual machine system, the size of your virtual storage, and the status of your virtual processors.

Usage Notes

The response order for this command is as follows:

- Storage
- Expanded Storage
- CPUs
- Vector Facilities
- Consoles
- Readers
- Punches
- Printers
- DASDs
- Tapes
- Displays
- Lines
- CTCAs
- CTLR
- Unsupported devices.

Responses

You receive responses for all devices that are defined in your virtual machine system. These responses are explained in the QUERY VIRTUAL command that corresponds to each device.

Migration Notes

VM/SP HPO

1. The VM/XA SP response provides additional information regarding expanded storage, CPUs, vector facilities, CTCAs, and CTLRs.
2. VM/XA SP does not provide DEST and CHANNEL information.

QUERY VIRTUAL CONSOLE

Privilege Class: G

Use QUERY VIRTUAL CONSOLE to display the status of your virtual console.

Query	[Virtual] CONsole
-------	-------------------

where:

[VIRTUAL] CONSOLE

displays the status of your virtual console device.

Usage Notes

None.

Responses

```

CONS vdev { ON luname } { TERM } { STOP } [VSM userid]
           { ON GRAF rdev } { NOTERM } { START }
           { ON LINE rdev }
           { DISCONNECTED }

vdev CL c { CONT } { HOLD } { CPY *nnn } { READY }
           { NOCONT } { NOHOLD } { COPY nnn } { NOTREADY } FORM userform

vdev TO userid { RDR } DIST distcode FLASHC ccc
               { PRT }
               { PUN }

vdev FLASH name CHAR char [char[char[char]]] MDYF cmod n FCB fcb

vdev { 3215 } { EOF } { OPEN spoolid } { KEEP } { MSG } { NAME fname ftype }
     { 3270 } { NOEOF } { CLOSED } { NOKEEP } { NOMSG } { NONAME }

[vdev SUBCHANNEL = vsub ]
    
```

where:

vdev

is the virtual device number of the virtual machine console.

luname

identifies an SNA/CCS terminal (controlled by a VTAM service machine) that is the virtual console of this virtual machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

rdev

is the real or logical device number associated with the virtual console if the console is not disconnected.

ON GRAF rdev

ON LINE rdev

is the real or logical device number associated with the virtual console if the console is not disconnected.

DISCONNECTED

indicates that the console is disconnected.

TERM

NOTERM

indicates whether terminal output is or is not produced when console spooling is in effect.

START

STOP

indicates whether console spooling has been started or stopped for the device.

VSM userid

is the userid of the VTAM service machine managing the device.

CL c

is the spool file class to be given to spool files created on this device.

CONT

NOCONT

controls the continuous spooling option.

For output files, CONT causes all CLOSE requests to be ignored unless specified with the EOF override option.

For reader files, CONT specifies that reading continues without end-of-file indications at the end of each virtual spool file until the virtual device is empty. NOCONT specifies that the end of file is signaled after each spool file is read, and then the reader is reset at the next logical file.

HOLD

NOHOLD

controls the HOLD status.

For output devices, HOLD prevents files from being punched or printed on a real device by putting a USER HOLD status on the file. If you specify NOHOLD, you override the spool device HOLD option and prevent the file from being placed in HOLD status.

For input devices, the HOLD/NOHOLD options override the setting for the virtual device or the spool file. The file is not put in HOLD status unless the KEEP option is also specified. If you specify HOLD, the file is not purged after processing and remains eligible for the device. If you specify NOHOLD, the file is purged unless the KEEP option is also specified.

CPY *nnn

COPY nnn

specifies the number of file copies you want printed or punched when the file is produced.

When the optional asterisk (*) is specified, the 3800 printer copies each page from the spooled output files nnn times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (*) is ignored and duplication occurs normally.

READY

NOTREADY

indicates whether the device is in a ready or not-ready status. The status can be changed by the READY and NOTREADY commands.

FORM userform

is the form associated with the device and assigned to the spool files as they are created.

TO userid

is the user identification of the virtual machine that receives spool files created on the device.

RDR**PRT****PUN**

is the spool file queue on which spool files created on this device is placed. Unless changed by the SPOOL or CLOSE command, output would normally go to the PRT queue for a console spool file.

DIST distcode

is the distribution code to be placed on spool files created on this device.

FLASHC ccc

is the flash count for the file.

FLASH name

is the name of the forms overlay frame superimposed on the output pages if the file is printed on the 3800 printer.

CHAR char

is the name of the character set or sets used to generate output if the file is printed on the 3800 printer.

MDFY cmod

is the name of the copy modification module used to alter output text if the file is printed on the 3800 printer.

n

is the copy modification character set number.

FCB fcb

is the name of the forms control buffer used in the vertical formatting of a page if the file is printed on the 3800 printer.

3215**3270**

is the virtual console device type. A 3270 device type is only supported if the user is logged on at a real 3270.

EOF**NOEOF**

EOF closes or NOEOF does not close the spool file on the specified device, regardless of the CONT setting for the device.

OPEN spoolid**CLOSED**

indicates that a spool file (spoolid) has been created on the console and that the file is still open (OPEN), or indicates that no spool file is open on the device (CLOSED).

KEEP**NOKEEP**

controls the KEEP status, which indicates whether the current spool file is to be placed in user HOLD after processing.

For output files, the **KEEP** status is placed on the spool file. After it is transcribed at the real printer or punch, the file is assigned user **HOLD** status, and is not purged. The file's **KEEP** status is retained. Specify **NOKEEP** to override the current **KEEP** setting for the device so that **KEEP** is not set on the spool file.

For input files, **KEEP** overrides the **KEEP** setting on either the virtual reader or the spool file. After processing, the file is placed in **HOLD** status, and the **KEEP** option on the file remains unchanged. Specify **NOKEEP** to override either file or device **KEEP** characteristics. The disposition of a file closed **NOKEEP**—whether it is purged or retained—is controlled by the **HOLD** status of the virtual reader. If it were closed **KEEP**, it would never be purged.

MSG

NOMSG

indicates whether you want to be informed about file printing and punching. If you specify **MSG**, you are informed of file printing and punching. The message is sent to you when the file begins printing or punching on a real device. This message is not sent if you issue **CP SET IMSG OFF**. Specify **NOMSG** to reset the **MSG** option.

NAME fname ftype

NONAME

is the filename and filetype, if any, to be assigned to spool files created on the device.

vsub

is the virtual subchannel number. This value is only displayed for 370-XA mode virtual machines.

Migration Notes

VM/SP HPO

1. The **PRT**, **PUN**, **RDR** indication on the third line exists only in **VM/XA SP**.
2. Virtual device numbers are 4 characters instead of 3.
3. Following **COPY** on the second line, an * (asterisk) indicates that the 3800 printer is copying each page from the spooled output files *nnn* times before going to the next page.
4. The **TO** or **FOR** syntax on the third line of the **VM/SP** response is replaced by **VM/XA SP**.
5. There are either one or two extra lines in the **VM/XA SP** response.

QUERY VIRTUAL CPUS

Privilege Class: G

Use QUERY VIRTUAL CPUS to display a list of virtual processors that are currently defined for your virtual machine.

Query	[Virtual] CPUS
-------	----------------

where:

[VIRTUAL] CPUS

displays a list of your virtual processors.

Usage Notes

None.

Responses

CPU nn ID FFssssssmmmm0000 [(BASE)] [status] [DEDICATED]

is presented for each processor defined in the virtual machine configuration.

where:

nn

is the processor address in hexadecimal, as established by DEFINE CPU.

FFssssssmmmm0000

is the 16-digit processor identification associated with the virtual machine.

FF

is the version code. These two digits are forced to X'FF' to indicate that the virtual machine is running under VM/XA SP.

ssssss

is the serial number of the processor as established by SET CPUID or by a CPU directory entry.

mmmm

is the model number of the processor.

0000

is the machine check extended logout length. This field is forced to X'0000'.

For a virtual processor using the SET command, these values are the same as for the real processor.

(BASE)

indicates that this is the base processor in a virtual MP configuration.

status

indicates the status of the virtual processor. Status can be:

STOPPED indicating that the virtual processor is in hard-stop state.

CHECK-STOPPED indicating that an error occurred on the virtual processor.

If no status is shown, the processor is running or in soft-stop state.

DEDICATED

indicates that the virtual CPU has a real CPU dedicated to it.

QUERY VIRTUAL CTCA

Privilege Class: G

Use QUERY VIRTUAL CTCA to display the status of the virtual channel-to-channel adapters that are accessible from the processor on which you issue the command.

Query	[Virtual] CTCa
-------	----------------

where:

[VIRTUAL] CTCA

requests a display of the status of each CTCA.

Usage Notes

None.

Response 1

```
CTCA vdev1 { NOT READY                } [SUBCHANNEL = vsub]
             { COUPLED TO userid vdev2 }
```

is displayed for each virtual channel-to-channel adapter in your virtual machine system.

Response 2

If your channel-to-channel adapter is a real device that is dedicated to your virtual machine, you receive the following response:

```
CTCA vdev1 ON DEV rdev [SUBCHANNEL = vsub] [IOASSIST ELIGIBLE]
```

where:

vdev1

is the virtual device number of the CTCA.

NOT READY

COUPLED TO

indicates the status of your CTCA.

userid

is the user identification of the virtual machine to which your CTCA is connected.

vdev2

is the virtual device number of the CTCA connected by the virtual machine identified as userid.

ON DEV rdev

is the real device number of your CTCA.

vsub

is the virtual subchannel number. This is displayed only if your virtual machine is operating in 370-XA mode.

IOASSIST ELIGIBLE

means that the device (vdev1) is eligible for the start interpretive execution assist.

Migration Notes

VM/SP HPO: The VM/XA SP response is upward compatible with HPO but adds an extra XA-only portion at the end of the HPO compatible part.

QUERY VIRTUAL DASD

Privilege Class: G

Use QUERY VIRTUAL DASD to display the status of all virtual direct access storage devices that are accessible from the processor on which you issue the command.

Query	[Virtual] DASd [DETAILS]
-------	--------------------------

where:

[VIRTUAL] DASD

requests a display of the status of your virtual direct access storage devices.

DETAILS

displays additional information for DASD as well as the usual response, if applicable.

Usage Notes

The DETAILS subcommand is only meaningful for dedicated DASD or mini-disk DASD on a cache subsystem. The responses for other DASD type devices will remain the same.

Response 1

DASD vdev type {valid } {R/W} nnnn CYL ON DASD rdev [SUBCHANNEL=vsub]
 {(TEMP)} {R/O}

is displayed for each minidisk or temporary disk in your virtual machine system.

Response 2

If the device is a real device dedicated to your virtual machine, you receive the following response:

DASD vdev ON DASD rdev {R/O} valid [SUBCHANNEL = vsub] [IOASSIST ELIGIBLE]
 {R/W}

Response 3

In addition to the normal QUERY VIRTUAL DASD response, this line will be displayed if the DETAILS option is specified:

[SYSCTL | DEVCTL | NOCTL | CACHE | NOCACHE]

is the response displayed if the DETAILS option is specified.

where:

SYSCTL

specifies that CCWs will be accepted which have a direct effect on global subsystem resources and functions and not just those related to the device.

DEVCTL

specifies that CCWs will be accepted which have an effect on resources and functions directly related to the device.

NOCTL

No CCWs are accepted that can exercise control of subsystem resources and functions, regardless of whether they are directly related to the device.

CACHE

means that the mini-disk has access to the control unit cache.

NOCACHE

means that CP will force I/O to the mini-disk to bypass the control unit cache.

Response 4

If the device is a multiple exposure paging subsystem, you receive the following additional response:

```
DASD vdev type volid {R/O} nnnn CYL [SUBCHANNEL = vsub]
      vdev vdev vdev ALTERNATE ADDRESSES ON rdev rdev rdev ...
      (continuation) ... [SUBCHANNEL = vsub vsub vsub vsub]
```

Each field is repeated to indicate the virtual device number, real device number, and virtual subchannel number of each of the three page-mode exposures.

where:

vdev

is the virtual device number.

type

is an IBM direct access device type.

volid

(TEMP)

is the volume label of the real device on which your virtual disk resides. Or if (TEMP), indicates that your virtual DASD has been defined from system temporary disk space.

R/W

R/O

indicates the type of access you have to the disk. R/O is read-only access; R/W is read and write access.

nnnn CYL

is the number of cylinders on the virtual DASD device.

ON DASD rdev

is the real device number of the volume containing the virtual device.

vsub

is the virtual subchannel number. This value is displayed only if your virtual machine is in 370-XA mode.

IOASSIST ELIGIBLE

means that the device (vdev) is eligible for the start interpretive execution assist.

QUERY VIRTUAL DASD

Migration Notes

1. VM/XA SP provides an additional operand, DETAILS.
2. The response is upward compatible by including an extra line for the DETAILS operand.
3. The VM/XA SP response is upward compatible with HPO but adds an extra XA-only portion at the end of the HPO compatible part.

VM/XA SF: VM/XA SP changed valid to TEMP for HPO compatibility.

QUERY VIRTUAL DEVICE

Privilege Class: G

Use QUERY VIRTUAL DEVICE to display the status of virtual devices that are accessible from the processor on which you issue the command.

Query	[Virtual] { vdev } [DETAILS] { vdev-vdev }
-------	---

where:

vdev

is the virtual device number of the device whose status is to be displayed.

vdev-vdev

is a range of virtual device numbers for devices whose status is to be displayed.

DETAILS

displays additional information for the specified virtual device or range of devices.

Usage Notes

The DETAILS option is only meaningful for dedicated DASD or mini-disk DASD on a cache subsystem. The responses for devices other than DASDs, if specified, will remain the same.

Response 1

Any other responses that you receive will depend upon the type of device that is located at the device number you specified. These responses are included in the QUERY VIRTUAL commands for those devices.

If the device is a:

- 3704 or 3705 Communications Controller
- Mass Storage Controller
- A virtual 270X line
- A miscellaneous device (not described in the QUERY VIRTUAL commands that follow),

you receive the following response:

```
type vdev ON DEV rdev [SUBCHANNEL = vsub] [IOASSIST ELIGIBLE]
```

where:

type

is the virtual device type. This can be:

Type	Meaning
CTLR	3704 or 3705 Communications Controller
MSC	Mass Storage Controller
LINE	Virtual 270X line
DEV	Any miscellaneous device

QUERY VIRTUAL DEVICE

vdev

is the virtual device number.

ON DEV rdev

is the real device number.

vsub

is the virtual subchannel number. This will be displayed only if your virtual machine is in 370-XA mode.

IOASSIST ELIGIBLE

means that the device (vdev) is eligible for the start interpretive execution assist.

Response 2

[SYSCTL | DEVCTL | NOCTL | CACHE | NOCACHE]

is the response displayed, in addition to the normal QUERY VIRTUAL DEVICE response, if the DETAILS option is specified.

where:

SYSCTL

specifies that CCWs will be accepted which have a direct effect on global subsystem resources and functions and not just those related to the device.

DEVCTL

specifies that CCWs will be accepted which have an effect on resources and functions directly related to the device.

NOCTL

No CCWs are accepted that can exercise control of subsystem resources and functions, regardless of whether they are directly related to the device.

CACHE

means that the mini-disk has access to the control unit cache.

NOCACHE

means that CP will force I/O to the mini-disk to bypass the control unit cache.

QUERY VIRTUAL GRAF

Privilege Class: G

Use QUERY VIRTUAL GRAF to display the status of all your locally attached virtual display devices.

Query	[Virtual] {GRaf}
-------	------------------

where:

[VIRTUAL] GRAF

requests a display of the status of all your virtual display devices.

Usage Notes

None.

Response 1

You receive the following response for each of your locally attached display devices:

```
GRAF vdev {ON GRAF rdev} [SUBCHANNEL = vsub] [IOASSIST ELIGIBLE]
          {ON GRAF ldev}
          {ON luname  VSM userid}
          {NOT DIALED }
```

where:

vdev

is the virtual device number of the display device.

ON GRAF rdev

ON GRAF ldev

shows the real (rdev) or logical (ldev) device that is dialed in to or attached to this virtual machine.

ON luname

identifies an SNA/CCS terminal (controlled by the VTAM service machine) that is dialed in to this virtual machine. It is the name from the VTAM definition of the terminal as an SNA logical unit.

VSM userid

is the userid of the VTAM service machine managing the device.

NOT DIALED

shows the status of a virtual display device that has not been attached using the CP DIAL command.

vsub

is the virtual subchannel number. This value is only displayed for 370-XA mode virtual machines.

IOASSIST ELIGIBLE

means that the device (vdev) is eligible for the start interpretive execution assist.

Response 2

You receive the following response for each of your locally attached display 328x printers:

```
PRT vdev { ON DEV rdev } [SUBCHANNEL = vsub] [IOASSIST ELIGIBLE]
          { ON DEV ldev }
```

where:

vdev

is the virtual device number of the display 328x printer.

ON DEV rdev

ON DEV ldev

shows the real (rdev) or logical (ldev) device number of the display 328x printer that is DIALED into or ATTACHED to this virtual machine.

vsub

is the virtual subchannel number. This value is only displayed for 370-XA mode virtual machines.

IOASSIST ELIGIBLE

means that the device (vdev) is eligible for the start interpretive execution assist.

Migration Notes

VM/SP HPO: The VM/XA SP response is upwardly compatible with VM/SP HPO but adds an extra 370-XA-only portion at the end of the VM/SP HPO-compatible section.

VM/XA SF: The VM/XA SP response uses GRAF instead of DISP.

QUERY VIRTUAL PRINTER

Privilege Class: G

Use QUERY VIRTUAL PRINTER to display the status of all virtual printers that are accessible from the processor on which you issue the command.

Query	Virtual {Printer} {Prt}
-------	----------------------------

where:

**VIRTUAL {PRINTER}
{PRT}**

displays the status of your virtual printers.

Usage Notes

None.

Response 1

```
PRT vdev CL a { CONT } { HOLD } { COPY nnn } { READY } FORM userform
              { NOCONT } { NOHOLD } { CPY *nnn } { NOTREADY }

vdev TO userid { RDR } DIST distcode FLASHC ccc
               { PRT }
               { PUN }

vdev FLASH name CHAR char [char[char[char]]] MDYF cmod n FCB fcb

[ vdev SIZE width length { 2WCGM } { BTS } { DATCK }
  { 4WCGM } { CFS } { NODATCK } ]

vdev type { EOF } { OPEN spoolid } { KEEP } { MSG } { NAME fname ftype }
          { NOEOF } { CLOSED } { NOKEEP } { NOMSG } { NONAME }

[vdev SUBCHANNEL = vsub]
```

is displayed for each virtual printer defined in your virtual machine configuration.

Note: The bracketed fourth line appears for the 3800 printer only.

where:

vdev

is the virtual device number of the printer.

CL a

is the spool file class to be given to spool files created on this device.

QUERY VIRTUAL PRINTER

CONT

NOCONT

controls the continuous spooling option.

CONT causes all CLOSE requests to be ignored unless specified with the EOF override option.

HOLD

NOHOLD

controls the HOLD status.

HOLD prevents files from being printed on a real device by putting a USER HOLD status on the file.

COPY nnn

CPY *nnn

is the number of copies requested for the spool file. This is shown as a 3-digit number (nnn). The asterisk (*) indicates that the 3800 printer will print each page from the spooled output files nnn times before going on to the next page. If the "*" is not present, the entire file will be printed sequentially.

READY

NOTREADY

indicates whether the device is in a ready or not-ready status. The status can be changed by the READY and NOTREADY commands.

FORM userform

displays the form associated with the device and assigned to the spool files as they are created.

TO userid

is the user identification of the virtual machine that will receive output from your virtual printer.

RDR

PRT

PUN

is the spool file queue that will receive your printer output.

DIST distcode

is the distribution code assigned to each spool file produced from the device.

FLASHC ccc

is the number of copies that are printed while the forms overlay frame is superimposed on the output pages. The default is ccc = 000.

FLASH name

is the name of the forms overlay frame superimposed on the output pages if the file is printed on the 3800 printer.

CHAR char

is the name of the character set or sets used to generate output if the file is printed on the 3800 printer.

MDFY cmod

is the name of the copy modification module used to alter output text if the file is printed on the 3800 printer.

n

is the copy modification character set number. If no copy modification character is set, then n will be 0.

FCB fcb

is the name of the forms control buffer used in the vertical formatting of a page if the file is printed on the 3800 printer.

width

indicates the form width code (in hexadecimal) for a virtual 3800 printer and is a two-character field.

length

is a 2-character field that indicates in decimal notation the length of the paper in half-inches loaded into a 3800 printer.

2WCGM

4WCGM

specifies the number of writable character generation modules (WCGM) for the virtual 3800 printer. A WCGM is a 64-position portion of the 3800's character generation storage that holds the scan elements of one character set.

BTS

CFS

designates the stacker assumed for the virtual 3800 printer. BTS is a burster-trimmer-stacker, and CFS is a continuous forms stacker.

DATCK

NODATCK

specifies the handling of certain virtual 3800 data checks. If DATCK is specified, all 3800 data checks are reflected to the virtual machine (provided the 'BLOCK DATA CHECK' CCW has not been issued). If NODATCK is specified, only data checks that occur due to invalid translate table specifications or unmatched FCB codes are reflected to the virtual machine.

type

is an IBM printer type.

EOF

NOEOF

EOF closes or NOEOF does not close the spool file on the specified device regardless of the CONT setting for the device.

OPEN spoolid

CLOSED

indicates that a spool file has been created on the console and that the file is still open (OPEN), or indicates that no spool file is open on the device (CLOSED).

KEEP

NOKEEP

controls the KEEP status, which indicates whether the current spool file is to be placed in user HOLD after processing.

The KEEP status is placed on the spool file. After it is transcribed at the real printer, the file is assigned user HOLD status, and is not purged. The file's KEEP status is retained.

MSG

NOMSG

indicates whether you want to be informed about file printing and punching. If you specify MSG, you will be informed of file printing and punching. The message is sent to you when the file begins either printing or punching on a real device. This message is not sent if you have issued CP SET IMSG OFF. Specify NOMSG to reset the MSG option.

QUERY VIRTUAL PRINTER

NAME fname ftype

NONAME

is the filename and filetype, if any, to be assigned to spool files created on the device.

vsub

is the virtual subchannel number. This value is displayed only for a 370-XA mode virtual machine.

Response 2

If the printer is a real device that is dedicated to your virtual machine, you receive the following response:

```
PRT vdev ON PRT rdev [SUBCHANNEL = vsub] [IOASSIST ELIGIBLE]
```

where:

vdev

is the virtual device number.

rdev

is the real device number.

vsub

is the virtual subchannel number. This value is displayed only for a 370-XA mode virtual machine.

IOASSIST ELIGIBLE

means that the device (vdev) is eligible for the start interpretive execution assist.

Usage Notes

The response line beginning *vdev SIZE* ... is included in the response for a 3800 printer only.

QUERY VIRTUAL PUNCH

Privilege Class: G

Use QUERY VIRTUAL PUNCH to display the status of all virtual punches that are accessible from the processor on which you issue the command.

Query	Virtual {PUnch} {PCH}
-------	--------------------------

where:

VIRTUAL {PUNCH}
{ PCH }

requests a display of the status of your virtual punches.

Usage Notes

None.

Response 1

PUN vdev CL a { CONT } { HOLD } { COPY nnn } { READY } FORM userform
 { NOCONT } { NOHOLD } { CPY *nnn } { NOTREADY }

vdev TO userid { RDR } DIST distcode
 { PRT }
 { PUN }

vdev FLASH name ccc CHAR char [char[char[char]]] MDFY cmod n FCB fcb

vdev type { EOF } { OPEN spid } { KEEP } { MSG } { NAME fname ftype }
 { NOEOF } { CLOSED } { NOKEEP } { NOMSG } { NONAME }

[vdev SUBCHANNEL = vsub]

is displayed for each of your virtual punches.

Response 2

If the punch is a real device that is dedicated to your virtual machine, the following response is displayed:

PUN vdev ON PUN rdev [SUBCHANNEL = vsub] [IOASSIST ELIGIBLE]

where:

vdev
is the virtual device number.

CL a
is the spool file class to be assigned to spool files created on this punch.

CONT
NOCONT

controls the continuous spooling option.

CONT causes all CLOSE requests to be ignored unless specified with the EOF override option.

HOLD
NOHOLD

controls the HOLD status.

HOLD prevents files from being punched on a real device by putting a USER HOLD status on the file. If you specify NOHOLD, you override the spool device HOLD option and prevent the file from being placed in HOLD status.

COPY nnn
CPY *nnn

is the number of copies requested for the spool file. This is shown as a 3-digit number (nnn). The asterisk (*) indicates that the 3800 printer will print each page from the spooled output files nnn times before going on to the next page. If the "*" is not present, the entire file will be printed sequentially.

READY
NOTREADY

indicates whether the device is in a ready or not-ready status. You can change the status with the CP READY and CP NOTREADY commands.

FORM userform

displays the form associated with the device and assigned to the spool files as they are created.

TO userid

is the user identification of the virtual machine that will receive output from your virtual punch.

RDR
PRT
PUN

is the spool file queue that will receive your punch output.

DIST distcode

is the spool file distribution code.

FLASH name ccc

FLASH name is the name of the forms overlay frame superimposed on the output pages if the file is printed on a 3800 printer.

ccc is the number of copies that are printed while the forms overlay frame is superimposed on the output pages. The default is ccc=000.

CHAR char

is the name of the character sets used to generate output if the file is printed on a 3800 printer.

MDFY cmod

is the name of the copy modification module used to alter output text if the file is printed on a 3800 printer.

n

is the copy modification character set number. If no copy modification character is set, then n will be 0.

FCB fcb

is the name of the forms control buffer used in the vertical formatting of a page if the file is printed on a 3800 printer.

type

is an IBM punch type.

EOF

NOEOF

EOF closes or NOEOF does not close the spool file on the specified device regardless of the CONT setting for the device.

OPEN spid

CLOSED

indicates that a spool file (spid) has been created on the console and that the file is still open (OPEN), or indicates that no spool file is open on the device (CLOSED).

KEEP

NOKEEP

controls the KEEP status, which indicates whether the current spool file is to be placed in user HOLD after processing.

The KEEP status is placed on the spool file. After it is transcribed at the real punch, the file is assigned user HOLD status, and is not purged. The file's KEEP status is retained. Specify NOKEEP to override the current KEEP setting for the device so that KEEP is not set on the spool file.

MSG

NOMSG

indicates whether you want to be informed about file printing and punching. If you specify MSG, you will be informed of file printing and punching. The message is sent to you when the file begins either printing or punching on a real device. This message is not sent if you have issued CP SET IMSG OFF. Specify NOMSG to reset the MSG option.

NAME fname ftype

NONAME

indicates whether a file name and file type are to be assigned to the spool file. If a name is to be assigned, it is also displayed.

vsub

is the virtual subchannel number. This value is displayed only if your virtual machine is in 370-XA mode.

IOASSIST ELIGIBLE

means that the device (vdev) is eligible for the start interpretive execution assist.

Migration Notes

VM/SP HPO: The VM/XA SP response is upwardly compatible with VM/SP HPO but adds an extra 370-XA-only portion at the end of the VM/SP HPO compatible part.

QUERY VIRTUAL READER

Privilege Class: G

Use QUERY VIRTUAL READER to display the status of all virtual readers that are accessible from the processor on which you issue the command.

Query	Virtual {Reader} {Rdr }
-------	----------------------------

where:

**VIRTUAL {READER}
 {RDR }**

requests a display of the status of your virtual readers.

Usage Notes

None.

Response 1

```
RDR vdev CL a { CONT } { HOLD } { EOF } { READY }
               { NOCONT } { NOHOLD } { NOEOF } { NOTREADY }

vdev type      { OPEN spid } { KEEP } [SUBCHANNEL = vsub]
               { CLOSED }  { NOKEEP }
```

is displayed for each of your virtual readers.

Response 2

If the reader is a real device that is dedicated to your virtual machine, you receive the following response:

```
RDR vdev ON RDR rdev [SUBCHANNEL = vsub] [IOASSIST ELIGIBLE]
```

where:

vdev

is the virtual device number.

CL a

is the spool file class that the reader services. Your reader can only read files of a matching class. A class of * indicates that your reader can read files of any class.

type

is an IBM reader device type.

CONT

NOCONT

controls the continuous spooling option.

CONT specifies that reading continues without end-of-file indications at the end of each virtual spool file until the virtual device is empty. NOCONT specifies

that the end-of-file is signaled after each spool file is read, and then the reader is reset at the next logical file.

HOLD

NOHOLD

controls the HOLD status.

The HOLD/NOHOLD options override the setting for the virtual device or the spool file. The file is not put in HOLD status unless the KEEP option is also specified. If you specify HOLD, the file is not purged after processing and remains eligible for the device. If you specify NOHOLD, the file is purged unless the KEEP option is also specified.

EOF

NOEOF

EOF closes or NOEOF does not close the spool file on the specified device regardless of the CONT setting for the device.

READY

NOTREADY

indicates whether the device is in a ready or not-ready status. You can change the status using the CP READY and CP NOTREADY commands.

OPEN spid

CLOSED

indicates the status of the spool file. If it is OPEN, the system spool file identifier is also displayed.

KEEP

NOKEEP

controls the KEEP status, which indicates whether the current spool file is to be placed in user HOLD after processing.

KEEP overrides the KEEP setting on either the virtual reader or the spool file. After processing, the file is placed in HOLD status, and the KEEP option on the file remains unchanged. Specify NOKEEP to override either file or device KEEP characteristics. The disposition of a file closed with NOKEEP (whether it is purged or retained) is controlled by the HOLD status of the virtual reader. If it were closed KEEP, it would never be purged.

vsub

is the virtual subchannel number. This value is displayed only if your virtual machine is in 370-XA mode.

rdev

is the real device number of the reader.

IOASSIST ELIGIBLE

means that the device (vdev) is eligible for the start interpretive execution assist.

Migration Notes

VM/SP HPO: The VM/XA SP response is upwardly compatible with HPO but adds an extra VM/XA SP-only portion at the end of the VM/SP HPO compatible section.

QUERY VIRTUAL STORAGE

Privilege Class: G

Use QUERY VIRTUAL STORAGE to display the size of your virtual storage.

Query	[Virtual] STORage
-------	-------------------

where:

[VIRTUAL] STORAGE

requests a display of the size of your virtual storage.

Usage Notes

None.

Responses

$$\text{STORAGE} = \left\{ \begin{array}{l} \text{nnnnnnnK} \\ \text{nnnM} \end{array} \right\}$$

where:

nnnnnnnK

nnnM

is the size of the virtual machine storage. nnnnnnnK is the storage size in multiples of 1,024 (one kilobyte) bytes; nnnM is the storage size in multiples of 1,048,576 (one megabyte) bytes.

Migration Notes

VM/SP HPO:

1. The VM/XA SP response provides storage sizes in megabyte increments;
2. The VM/XA SP response may provide storage sizes larger than sixteen megabytes.

QUERY VIRTUAL TAPES

Privilege Class: G

Use QUERY VIRTUAL TAPES to display the status of all the virtual tape devices that are accessible from the processor on which you issue the command.

Query	[Virtual] Tapes
-------	-----------------

where:

[VIRTUAL] TAPES

displays the status of your virtual tape devices.

Usage Notes

None.

Responses

```
TAPE vdev ON DEV rdev {R/O} [SUBCHANNEL = vsub] [IOASSIST ELIGIBLE]
                       {R/W}
```

is displayed for each tape device attached to your virtual machine system

where:

vdev

is the virtual device number.

rdev

is the real device number.

R/O

R/W

is the type of access you have to the device. R/O indicates that you have read only access; R/W indicates that you have both read and write access.

vsub

is the virtual subchannel number. This value is displayed only if your virtual machine is in 370-XA mode.

IOASSIST ELIGIBLE

means that the device (vdev) is eligible for the start interpretive execution assist.

Migration Notes

VM/SP HPO: The VM/XA SP response is upwardly compatible with VM/SP HPO but adds an extra VM/XA SP-only portion at the end of the VM/SP HPO compatible part.

QUERY VIRTUAL UR

Privilege Class: G

Use QUERY VIRTUAL UR to display the status of all unit record devices (printers, punches, and readers) that are accessible from the processor on which you issue the command.

Query	[Virtual] UR
-------	--------------

where:

[VIRTUAL] UR

requests a display of the status of all your unit record devices.

Usage Notes

None.

Responses

See the commands QUERY VIRTUAL PRINTER, QUERY VIRTUAL PUNCH, and QUERY VIRTUAL READER.

Migration Notes

VM/SP HPO

1. The VM/XA SP response does not include DEST information.
2. VM/XA SP does not support logical printers.
3. The VM/XA SP response is upwardly compatible with VM/SP HPO but adds an extra VM/XA SP-only portion at the end of the VM/SP HPO compatible part.

QUERY VIRTUAL VECTOR

Privilege Class: G

Use QUERY VIRTUAL VECTOR to display the status of all your virtual Vector Facilities.

Query	[Virtual] VECtor
-------	------------------

where:

[VIRTUAL] VECTOR

displays the status of your virtual Vector Facilities.

Usage Notes

If no real Vector Facility is available to service a given virtual Vector Facility (for example, because it was taken offline or because it is on a dedicated processor), a response is issued.

Response 1

VEC nn

This response is displayed for each virtual vector facility that is defined and may be accompanied by a warning message (see “Usage Notes”).

where:

nn

is the address of the virtual Vector Facility.

Response 2

NO VECTOR FACILITIES DEFINED

This response is displayed when no real Vector Facility is available.

QUERY VIRTUAL XSTORE

Privilege Class: G

Use QUERY VIRTUAL XSTORE to display the size of the Expanded Storage capability attached to your virtual machine.

Query	[Virtual] { XStore } { XStorage }
-------	--------------------------------------

where:

**[VIRTUAL] { XSTORE }
{ XSTORAGE }**

displays the size of the Expanded Storage attached to your virtual configuration.

Usage Notes

None.

Responses

$XSTORE = \left\{ \begin{array}{l} \text{nnnnnnnnM} \\ \text{NONE} \end{array} \right\}$

where:

nnnnnnnnM

displays in megabytes the actual size of Expanded Storage attached to your virtual configuration.

NONE

is the response when the Expanded Storage is not attached to the virtual configuration.

QUERY V = R

Privilege Class: B

Use the QUERY V = R command to display information about real storage allocated to V = R and V = F guests from the V = R area.

Query	V=R
-------	-----

where:

V = R

requests the display of information about real storage allocation of V = R and V = F guests in the V = R region.

Usage Notes

None.

Responses

```

USER      STORAGE  STORAGE  XSTORE  XSTORE
          ORIGIN  SIZE     ORIGIN  SIZE
userid   nnnnM    nnnnM    nnnnnnnM nnnnnnnM
userid   nnnnM    nnnnM    nnnnnnnM nnnnnnnM
.
.
userid   nnnnM    nnnnM    nnnnnnnM nnnnnnnM

SIZE OF THE V=R REGION                nnnnM

MAXIMUM NUMBER OF PREFERRED USERS ALLOWED  nnn
    
```

where:

USER

is the userid of the user assigned the corresponding host real storage

STORAGE ORIGIN

is the main storage origin of the block of contiguous storage assigned to this user.

STORAGE SIZE

is the size in megabytes of the host real storage assigned.

XSTORE ORIGIN

is the Expanded Storage origin assigned to this user.

XSTORE SIZE

is the size in megabytes of the Expanded Storage assigned. This is the configured size.

SIZE OF THE V = R REGION

is the size in megabytes of the V = R region. This is the same value as was specified in the sysgen via the VRSIZE operand of the SYSSTORE macro.

QUERY V=R

MAXIMUM NUMBER OF PREFERRED USERS ALLOWED

is the maximum users that can be logged on as either V=R or V=F. This number includes those already logged on and shown in the detail of this response.

QUERY XSTORE

Privilege Class: B

Use QUERY XSTORE to display the current assignment of the real Expanded Storage facility.

Query	{ XSTORE } [USER [userid]] { XSTORAGE } [MAP]
-------	--

where:

XSTORE

XSTORAGE

requests a display of the size and percent of the Expanded Storage in use by CP for paging and the size of each virtual machine partition.

USER userid

indicates that the Expanded Storage partition for this specific userid is to be displayed. If the "userid" operand is left off, all guest partitions are displayed.

MAP

indicates that a map of Expanded Storage is to be displayed. This map shows how Expanded Storage is currently allocated.

Usage Notes

1. The responses to QUERY XSTORE may display different partition sizes than you requested when you issued the ATTACH XSTORE nnnnnnnnM or RETAIN XSTORE nnnnnnnnM commands.

If you tried to attach a partition that was too large, CP assigns the maximum size partition available to the virtual machine.

If you retained a portion of Expanded Storage for CP paging, CP has access to the retained size and can also use any portion of Expanded Storage that was not attached to the virtual machine.

Example

If you divide Expanded Storage between CP and a virtual machine (MVS) as follows:

```
Configured and Installed Expanded Storage = 256M
RETAIN XSTORE 64M
ATTACH XSTORE 128M to MVS
```

a possible response to QUERY XSTORE is:

```
XSTORE= 256M,  ONLINE= 256M
XSTORE= 128M,  USERID= SYSTEM  USAGE= 72%  RETAINED= 64M  PENDING= 0M
XSTORE= 128M,  USERID= MVS
XSTORE= 64M,   USERID= (NONE)  MAX ATTACH= 64M
```

Note: CP has access to the remaining 64M not assigned to MVS and not specified by the RETAIN XSTORE command.

Response 1

The response to QUERY XSTORE is:

XSTORE= nnnnnnnnM, ONLINE= nnnnnnnnM

XSTORE= nnnnnnnnM, USERID= SYSTEM USAGE= $\left\{ \begin{array}{l} \text{INIT} \\ \text{nnn\%} \\ \text{NONE} \end{array} \right\}$ RETAINED= nnnnnnnnM PENDING= nnnnnnnnM

XSTORE MDC min=nnnnnnnnM, max=nnnnnnnnM, usage=ppp%

XSTORE= nnnnnnnnM, USERID= userid

XSTORE= nnnnnnnnM, USERID= (NONE) MAX ATTACH= nnnnnnnnM

Line 1 lists the installed and configured Expanded Storage.

Line 2 lists usable Expanded Storage, the percentage of CP-usable Expanded Storage that CP is using, retained Expanded Storage, and pending retained Expanded Storage.

Line 3 lists the minimum and maximum size possible for the minidisk cache and the size of the minidisk cache as a fraction of the Expanded Storage CP partition.

Line 4 lists the size of a guest's partition. It is repeated for each guest partition, if any.

Line 5 lists the total amount of Expanded Storage that could be attached to virtual machines and the largest amount of Expanded Storage that could be attached to a single virtual machine. It is displayed after all guest partitions have been displayed.

where:

nnnnnnnnM

is the actual size of Expanded Storage that is available to the processor environment. nnnnnnnnM is the size in megabytes.

userid

is the userid of the user to whom the Expanded Storage is attached.

USAGE = $\left\{ \begin{array}{l} \text{INIT} \\ \text{nnn\%} \\ \text{NONE} \end{array} \right\}$

INIT indicates that Expanded Storage is initializing. nnn% indicates the percentage of CP-usable Expanded Storage being used by CP. NONE indicates there is no Expanded Storage for CP usage.

PENDING = nnnnnnnnM

is the amount of Expanded Storage that has not yet been made retained. The amount of Expanded Storage specified on the last RETAIN XSTORE command could not be fully satisfied. When this amount is detached from virtual machines, it will become retained Expanded Storage.

MAX ATTACH = nnnnnnnnM

indicates the maximum amount of Expanded Storage that could be attached to a single virtual machine. If this does not match the XSTORE value on this line, then the Expanded Storage is fragmented.

Response 2

The response to QUERY XSTORE USER:

XSTORE= nnnnnnnM, USERID= userid

XSTORE= nnnnnnnM, USERID= (NONE) MAX ATTACH= nnnnnnnM

Line 1 lists the size of a guest's partition. It is repeated for each guest partition, if any.

Line 2 lists the total amount of Expanded Storage that could be attached to virtual machines and the largest amount of Expanded Storage that could be attached to a single virtual machine. It is displayed after each guest partition has been displayed.

Response 3

The response to QUERY XSTORE USER *userid*:

XSTORE= nnnnnnnM, USERID= userid

lists the size of the specified guest's partition.

Response 4

The response from QUERY XSTORE MAP is:

START	SIZE	STATUS	%-IN-USE	%-UNUSABLE
nnnnnnnnM	nnnnnnnnM	CP	nnn%	nnn%
		CP(RETAINED)		
		useridnn		
		useridnn(N/A)		
		useridnn(MIG)		
		N/A		

where:

START

indicates the megabyte where the partition starts.

SIZE

indicates the size of this partition in megabytes.

STATUS

indicates the status or owner of the partition as follows:

CP

indicates CP is using the partition for CP paging. It is available for attaching to a virtual machine.

CP(RETAINED)

indicates CP has exclusive use of the partition for CP paging. It is not available for attachment to a virtual machine.

useridnn

The specified user has this partition attached to its virtual machine.

useridnn(N/A)

The specified user's virtual machine has this partition attached. This particular area is not available for guest use. One or more other partitions exist with this same userid above or below this partition. They are treated as one single guest partition, even though it appears that the guest has discontinuous Expanded Storage.

userid(MIG)

This partition is in the process of being attached to the virtual machine specified. CP is migrating data from this area prior to completing the attachment.

N/A

indicates this area of Expanded Storage is not available for use by CP or a guest.

%-IN-USE

indicates the percentage of this partition that contains a page of guest storage. It is displayed only for CP and CP(RETAINED) partitions.

%-UNUSABLE

indicates the percentage of the partition that CP cannot use for paging due to hardware conditions. This appears for each partition except for those marked "N/A" or "userid(N/A)."

READY

Privilege Class: G

Use the **READY** command to set a device-end interruption pending for a specified virtual device.

READY	vdev
-------	------

where:

vdev

is a virtual device number.

Usage Notes

1. Use the **READY** command to clear a previous “not ready” condition for a device.
2. Do not use the **READY** command for a virtual channel-to-channel adapter (CTCA). An invalid device type error will result.

Responses

None.

RECORDING

Privilege Class: A, B, C, E and F

Use the RECORDING command to alter the processing parameters for CP recording facilities.

RECORDing	$\left. \begin{array}{l} \text{ACCOUNT} \\ \text{EREP} \\ \text{ALL} \end{array} \right\} \left[\begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right] [\text{LIMIT nnn}] [\text{PURGE}] [\text{userid}]^1$
-----------	--

¹The userid option is valid only for class A and class B users.

where:

ACCOUNT

EREP

ALL

specify the record type(s) for which the parameter change is being requested.

ON

requests that recording of the specified record type(s) be started.

OFF

requests that recording of the specified record type(s) be stopped.

LIMIT nnn

specifies a decimal value, from 1 to 255, which is used as a threshold indicating when to notify the system operator and the receiving virtual machine that uncollected records are accumulating.

Uncollected records are records that have been created but not received by the recording virtual machine. Whenever the number of uncollected records of the specified type equals an integral multiple of the LIMIT, the system operator and the virtual machine user who normally receives the records are notified.

For example, suppose you specify LIMIT as 4. The system operator and the virtual machine user who normally receives the records will be notified when four uncollected records accumulate. Notifications will be issued when the number of uncollected records reaches 8, 12, 16, and so on.

The default limit for ACCOUNT records is 20 and for EREP records is 2. Any LIMIT setting that you indicate lasts for the duration of the IPL.

PURGE

requests the purging of any records in storage that have been collected but not retrieved.

userid

specifies the user eligible to retrieve the records of the type specified. If you are specifying your own userid, you can specify userid as an asterisk (*).

As part of the VM/XA SP system generation process, your system programmer indicated which virtual machine would retrieve records of a designated type. If the userid that you specify on the RECORDING command line is different from the userid that your system programmer specified at system generation time, the change will be valid only for your current system IPL session.

This operand is only valid for class A and B users.

Usage Notes

1. If a virtual machine is in the process of retrieving records when you issue the **RECORDING** command to purge records or change parameters, and CP cannot terminate the retrieval, the recording table will not be modified. You will receive a message informing you of the condition.
2. If you are not a class A or class B user, your userid must match the userid in the recording table entry for the specified record type.
3. If you do not specify any of the optional operands, **ON** is assumed.

Responses

COMMAND COMPLETE

indicates that the command has completed successfully.

REDEFINE

Privilege Class: G

Use REDEFINE to change the virtual device number of any existing virtual device in your configuration.

REDEFine	vdev1 [AS] vdev2
----------	------------------

where:

vdev1

is the virtual device number of an existing virtual device.

vdev2

is the new virtual device number.

Usage Notes

1. If you are changing the virtual device number of a multiple-exposure device, specify the base exposure as the virtual device number in both vdev1 and vdev2. When you change the virtual device number of a multiple-exposure device, all exposures are redefined.

Thus, if 01B0 is the device number of a 2305, the command:

```
redefine 01B0 as 01D0
```

redefines device numbers 01B0 through 01B7 as 01D0 through 01D7, respectively.

2. Virtual paging subsystem devices should not be redefined in order to protect the integrity of the user's virtual paging subsystem.
3. When you use the REDEFINE command to alter the configuration of your virtual machine, the changes are temporary and are only in effect for the current session.
4. If your virtual machine is operating in 370-XA mode, the virtual device numbers you specify can be no higher than FFFF. If your machine is operating in System-370 mode, you can define virtual device numbers through FFFF, but the virtual machine will only be able to use the virtual devices numbered 0 through 1FFF.
5. Informational responses will be sent to your display to confirm that the configuration change you requested has taken place. These responses will not appear if you have issued the SET IMSG OFF commands.

Responses

```
type vdev DEFINED
```

confirms that the device number has been changed.

Type	Meaning
DASD	Direct Access Storage Device
TAPE	Magnetic Tape
RDR	Card Reader
PRT	Line Printer
PUN	Card Punch
GRAF	Display Device
CONS	Console
CTCA	Channel-to-Channel adapter
CTLR	3704 or 3705 Communications Controller
MSC	Mass Storage Controller
DEV	Any other device.

Migration Notes

VM/XA SF: VM/XA SP uses GRAF rather than DISP in its response.

REPEAT

REPEAT

Privilege Class: D

Use the REPEAT command to increase the number of copies of an output file beyond that quantity originally requested by the file's creator.

REPEAT	rdev [<u>1</u>] [nnn]
--------	------------------------------

where:

rdev

is the real device number of the output device (printer or punch) whose current active spool file is to be repeated.

nnn

is the number of extra copies of the current file that are to be made. If you do not specify a value, the number of extra copies defaults to 1; the maximum value that you may specify for nnn is 255.

Usage Notes

This command applies only to files that are currently being written to a real punch or printer.

Responses

```
{PRT} rdev REPEATED userid FILE spoolid RECS { nnn } COPY [*]nnn SEQ sss  
{PUN}                                     { nnnK }  
                                          { nnnM }
```

where:

rdev

is the real device number of the device whose current output is being repeated.

userid

is the identification of the user who is to receive the output.

spid

is the system-assigned spoolid number of the file.

```
RECS { nnn }  
     { nnnK }  
     { nnnM }
```

is the number of records in the spool file remaining to be printed. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

is the remaining number of copies of the file to be created.

When the optional asterisk (*) is specified, the 3800 printer will print each page from the spooled output files nnn times before going on to the next page.

SEQ sss

is the sequence number for the file on the printer or punch.

Migration Notes*VM/SP HPO*

1. VM/XA SP provides a one-line response instead of a two-line response as in VM/SP HPO.
2. VM/XA SP does **not** accept the HOLD option on the REPEAT command as allowed by XM/SP HPO.
3. VM/XA SP does not support logical printers.

REQUEST

Privilege Class: G

Use the REQUEST command to make an attention interruption pending for your virtual console.

REQuest	
---------	--

Usage Notes

1. The REQUEST command performs the same functions as the CP ATTN command; you can use the two interchangeably.
2. The CP BEGIN command is not required after you issue REQUEST.
3. CP ignores the ATTN and REQUEST commands if an I/O operation is in progress at the console, or if other interrupts are pending for your virtual machine. This condition may result if you issue the ATTN or REQUEST command by means of DIAGNOSE X'08'.

Responses

None.

RESET

Privilege Class: B,G

Use the RESET command to clear all pending interrupts from a specified virtual device.

RESET	{ vdev RESERVE [ON] rdev }
-------	-------------------------------

where:

vdev

is the virtual device number of the device to be reset.

rdev

is the real device number of the device to be released.

Usage Notes

1. Use RESET RESERVE with extreme caution. If you *must* release a device that is being reserved by CP for some user, you may use the CP RESET RESERVE command. You should use this command only when you know that the user for whom CP is holding the reserve is hung or is holding the reserve for too long and preventing access to the device by other users.
2. In addition to clearing all pending interrupts, all virtual sense bytes and error conditions occurring as a result of unit checks are reset.
3. The operator of a virtual system can issue the CP RESET command to return to CP any local displays that were connected to the virtual system using the DIAL command.
4. The reserve status of system DASD may be determined by issuing the class B CP QUERY DASD RESERVE command.
5. When RESET RESERVE is issued to a device, that device immediately becomes available to the other sharing system.

Response 1

DEVICE RESET - type vdev

is the response you receive upon the successful completion of the RESET vdev command.

where:

type

indicates the type of device that has been reset (CON, CTCA, CTLR, DASD, GRAF, MSC, PRT, PUN, RDR, TAPE, or UR).

vdev

is the virtual device number.

RESET

Response 2

RESERVE ON type rdev RESET

is the response you receive upon the successful completion of the RESET RESERVE command.

Migration Notes

VM/SP HPO: The VM/XA SP response provides the type of device which has been reset.

VM/XA SF

1. VM/XA SP uses GRAF rather than DISP in its response.
2. The format of the VM/XA SP response has been rearranged.

RETAIN XSTORE

Privilege Class: B

Use the RETAIN XSTORE command to:

- Retain some or all of Expanded Storage for CP's exclusive use. This limits the maximum size of Expanded Storage that can be assigned to guest virtual machines to the maximum size of Expanded Storage, less the amount of Expanded Storage retained for CP use.
- Specify the maximum and the minimum amount of Expanded Storage that can be used by a minidisk cache. The actual size of the minidisk cache varies between those two values and is determined dynamically according to the activity of paging and minidisk I/O.

RETAIN	$\left\{ \begin{array}{l} \text{XStore} \\ \text{XStorage} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{nnnnnnnM} \\ \text{ALL} \\ \text{NONE} \end{array} \right\}$
	$\left\{ \begin{array}{l} \text{MDCache} \\ \left[\begin{array}{l} \text{ON} \\ \text{OFF} \\ \text{ALL} \\ \left[\begin{array}{l} \text{minM} \\ \text{0M} \end{array} \right] \left\{ \begin{array}{l} \text{maxM} \\ \text{MAX} \end{array} \right\} \end{array} \right] \end{array} \right\}$	

where:

XSTORE

XSTORAGE

specifies you are retaining Expanded Storage for CP use.

nnnnnnnM

specifies the size expressed in megabytes of real Expanded Storage that will be retained for CP use.

ALL

specifies that all of Expanded Storage configured on the system is retained for CP use.

NONE

specifies that none of the Expanded Storage configured on the system is retained for CP use.

MDCACHE

indicates that minidisk cache parameters are being changed.

ON

indicates that CP is to use the amount of minidisk cache that best balances I/O and paging performance. This is the default. The minidisk cache size can range from 0M to the size of the CP partition.

OFF

indicates that CP is not to use minidisk cache. The minidisk cache size is set to 0M.

ALL

indicates that CP is to use the CP partition of Expanded Storage exclusively for minidisk cache. CP will do no paging to Expanded Storage.

RETAIN XSTORE

minM **0M**

indicates the minimum size of the minidisk cache in megabytes. *min* can be from 0 to 16 777 216. If *minM* is specified, *maxM* must also be specified. If *minM* is not specified, a value of 0M is assumed.

maxM **MAX**

indicates the maximum size of the minidisk cache in megabytes. *max* can be from 0 to 16 777 216. If *minM* is specified, *max* must be greater than or equal to *min*. If MAX is specified for the maximum size of the minidisk cache, the upper limit of the minidisk cache is set to the size of the CP partition.

Note: If the value specified as *max* is 0, the minidisk cache is turned off.

When *min* equals *max*, the size of the minidisk cache is fixed. It does not vary dynamically.

Usage Notes

1. If the amount specified is greater than the amount available, the amount requested is rounded down to the amount available.
2. Issuing the RETAIN XSTORE command at initialization suspends virtual machine execution until Expanded Storage initialization completes. To avoid this, you can (a) wait until XSTORE initialization completes to issue RETAIN XSTORE or (b) issue RETAIN XSTORE from a disconnected virtual machine that has been automatically logged on.
3. In the initial state, all the Expanded Storage is assigned to the CP partition, and none is retained.
4. The RETAIN XSTORE command allows the class B operator to control the amount of Expanded Storage that can be assigned to guest virtual machines. A value of ALL inhibits the use of Expanded Storage by guest virtual machines regardless of the user directory Expanded Storage statement and size value. The user directory Expanded Storage size is the guest partition size established, if the directory size is less than or equal to the maximum guest partition size at logon. If the user directory size is greater than the maximum guest partition, a guest partition equal to the maximum guest partition is established.
5. If the RETAIN command is issued after Expanded Storage has been given to the user, the response to the command may indicate that the amount retained is less than the amount requested. In this case, pending retain is created.

To recover the Expanded Storage from a guest, use the following procedure:

- a. Guest operator issues VARY XSTORE OFFLINE.
- b. Guest operator issues DETACH XSTORE.
- c. CP will now acquire the Expanded Storage.
- d. Guest operator or the CP operator issues the ATTACH XSTORE command. (Note that ATTACH is a class B command.)
- e. Guest operator issues the VARY XSTORE ONLINE command.

Note: The guest operator must be able to issue a VARY XSTORE command to use this procedure. If a guest operator is not available, the guest operating system must be quiesced before being DETACHED.

6. The amount of Expanded Storage retained for exclusive use for CP paging may occupy more than one extent or partition of Expanded Storage. The amount may be specified in 1 megabyte increments.
7. If you are running a processor-constrained environment, you should consider issuing `RETAIN XSTORE MDCACHE OFF`.
8. If `RETAIN XSTORE MDCACHE maxM` is issued when *max* is greater than the current size of the minidisk cache, a gradual reduction in the use of Expanded Storage for CP paging will occur until the minidisk cache attains the new size. This will not cause a rapid migration of paging data to auxiliary storage.

Responses

Response 1

```
XSTORE RETAINED SIZE={nnnnnnnnM} PENDING={nnnnnnnnM}
                   {NONE}                {NONE}
```

where:

SIZE = nnnnnnnnM

is the amount of Expanded Storage, expressed in megabytes, that is retained for CP use. If `NONE` appears after `SIZE`, then no Expanded Storage was retained.

PENDING = nnnnnnnnM

is the amount of Expanded Storage that is waiting for retention. This means the amount of Expanded Storage specified could not be fully satisfied. When this amount is detached from virtual machines it will become retained Expanded Storage.

Be careful of fragmentation caused by pending retained Expanded Storage. To avoid fragmentation, it is advisable to issue the `RETAIN XSTORE` command again with the amount to be retained equal to the amount given for `SIZE` in this response.

Response 2

```
XSTORE RETAINED MDCACHE minM maxM
```

where:

minM

indicates the minimum value for the size of minidisk cache. The value of *min* can be from 0 to 16 777 216.

maxM

indicates the maximum value for the size of minidisk cache. The value of *max* can be from 0 to 16 777 216.

REWIND

Privilege Class: G

Use the REWIND command to rewind (but not unload) a tape on a real tape drive.

REWind	vdev
--------	------

where:

vdev

is the virtual device number of the tape drive on which the tape is mounted.

Usage Notes

None.

Response 1

REWIND NOT PERFORMED

is the response you receive when the rewind operation is pending.

Response 2

REWIND COMPLETE

is the response you receive when the rewind operation is started.

Migration Notes

VM/XA SF: The VM/XA SP responses have been changed: REWIND COMPLETE and REWIND NOT PERFORMED instead of TAPE vdev REWIND STARTED and TAPE vdev NOT READY as in VM/XA SF.

SAVESEG

Privilege Class: E

The SAVESEG command is provided as the final step of a saved segment build process. This command allows a system administrator or an installation procedure to save the page range areas previously specified by a DEFSEG command. The areas contain the appropriate objects, such as code or data, that were loaded into these defined areas by an install procedure.

Use the SAVESEG command to save the pages associated with the name of the saved segment into a previously defined class S (skeleton) system data file (SDF). Upon successful completion of the SAVESEG command the class S SDF becomes an active class A SDF.

If the saved segment was defined as a member of a saved segment space, then the saved segment space class S SDF would have its member saved segment entry in the saved segment space's descriptor page updated to saved status. When all members of a saved segment space indicate saved status, then the class S saved segment space SDF becomes an active class A SDF, or class R.

The saved segment may have been defined as a member of several saved segment spaces. In this case, each saved segment space having this member would have its descriptor page updated.

SAVESEG	dcssname
---------	----------

where:

dcssname

is the name (a 1-8 character alphanumeric string) of the segment to be saved. This is the filename of a class S (skeleton) SDF previously defined with the DEFSEG command. When a DEFSEG command using the SPACE operand is issued, the "dcssname" specified in the SAVESEG command is interpreted as the member name specified in the DEFSEG command.

Usage Notes

1. After the execution of this command, the system data file has a class of A or R. Class A indicates that it is in the unrestricted available state and may be accessed by any user. Class R indicates that the DCSS or saved segment space is restricted; to load it, the user must have a NAMESAVE statement for the saved segment in the directory entry.
2. Only one available (class A or R) saved segment with a particular name may exist at a given time. If any class A or R saved segment already exists with the same name as the one being saved, and there are no virtual machine users associated with the saved segment, then the old copy of the saved segment is purged and the new one is made available.

If there are users associated with the old copy and neither the old copy nor the new copy of the saved segment is a CPDCSS (that is, no page ranges were defined with the "SC" page descriptor code), the old copy is marked as pending purge, and the new copy is saved. Any users that were associated with the old copy continue to reference the old (now pending purge) saved segment. Subsequent users are associated with the new copy.

SAVESEG

If there are users associated with the old copy and either the old copy or the new copy of the saved segment is a CPDCSS (that is, one or more page ranges were defined with the "SC" page descriptor code), the SAVESEG command is not accepted.

3. If there is a copy of the saved segment that is class P, pending purge (the PURGE NSS command was used to purge it while there were users associated with it). The same rules apply as described in Usage Note 2:
 - If neither the old copy nor the new copy of the saved segment is a CPDCSS, then the new copy *may* be saved
 - If either the old or the new copy is a CPDCSS, the new copy *cannot* be saved.
4. Saved segments are loaded via DIAGNOSE code X'64'. Named saved systems (NSS) are loaded via the IPL command. A member saved segment cannot be loaded until at least one of the saved segment spaces of which it is a member is class A or R.
5. The contents of and storage keys for pages defined as "no data saved" on the DEFSEG command (or defaulted to "no data saved") are not saved in the saved segment system data file when the SAVESEG command is executed. When referenced by a user who loads the saved segment, storage contents and keys are zero.
6. The SAVESEG command may not be issued for a "spacename" (see the DEFSEG command).
7. The saved segment space SDF is changed from class S to class A when all members within the saved segment space have been saved.
8. The SAVESEG command cannot be issued by a V=R or V=F guest.

Responses

None.

Migration Notes

VM/SP HPO: In HPO, this function is done with the SAVESYS command.

VM/XA SF: The addition of CP saved segments adds restrictions on the saving of saved segments (both CP and non-CP) with the same name. See Usage Notes 2 and 3 for details.

SAVESYS

Privilege Class: E

Use the SAVESYS command to save a named saved system (NSS) into a previously defined system data file (SDF).

SAVESYS	name
---------	------

where:

name

is the name of the NSS to be saved. This is the filename of a class S (skeleton) system data file previously defined with the DEFSYS command.

Usage Notes

1. After the execution of this command, the system data file has a class of A or R. Class A indicates that it is in the unrestricted available state, and may be accessed by any user. Class R indicates that the NSS is restricted and the users must have a NAMESAVE statement for the NSS in their directory entries in order to load it.
2. Only one available NSS (class A or R) with a particular name may exist at a given time. If any class A or R NSS already exists with the same name as the one being saved, and there are no virtual machine users associated with the NSS, then the old copy of the NSS is purged and the new one is made available.

If there are users associated with the old copy and neither the old copy nor the new copy of the NSS is a CP system service NSS (that is, no page ranges were defined with the "SC" page descriptor code), the old copy is marked as pending purge, and the new copy is saved. Any users that were associated with the old copy continue to reference the old (now pending purge) NSS. Subsequent users are associated with the new copy.

If there are users associated with the old copy and either the old copy or the new copy of the NSS is a CP system service NSS (that is, one or more page ranges were defined with the "SC" page descriptor code), the SAVESYS command is not accepted.

3. If there is a CP NSS with the specified name that is class P, pending purge (the PURGE NSS command was used to purge it while there were users associated with it), then the same rules as described in Usage Note 2 apply:
 - If neither the old copy nor the new copy is a system service NSS, the new copy *may* be saved
 - If either the old or the new copy is a CP system service NSS, the new copy *cannot* be saved.
4. If a skeleton system data file exists for an NSS with the specified system name, then the data is saved into it. If no skeleton NSS file exists, but there is a DCSS skeleton file with the same name, the data and storage keys (but not the virtual machine environment) is saved into the DCSS skeleton file, exactly as if the SAVESEG command had been issued instead.

SAVESYS

5. The contents of and storage keys for pages defined as or defaulted to “no data saved” on the DEFSYS command are not saved in the NSS system data file when the SAVESYS command is executed. When referenced by a user who IPLs the NSS, storage contents and keys are zero.
6. A named saved system (NSS) is loaded via the IPL command. A saved segment is loaded via DIAGNOSE code X'64'.
7. The SAVESYS command cannot be issued by a V=R or V=F guest.

Responses

None.

Migration Notes

VM/SP HPO

1. VM/XA SP provides a message instead of a response.
2. In VM/XA SP systems are saved to system data files instead of being saved to reserved DASD space as in VM/SP HPO.

VM/XA SF: The addition of CP system service NSSs adds restrictions on the saving of NSSs (both CP and non-CP) with the same name. See Usage Notes 2 and 3 for details.

SCREEN**Privilege Class: G**

Use the SCREEN command to change either the color definition, the highlighting definition, or both definitions for a virtual machine console.

SCREen	$\left(\begin{array}{l} \text{area} \left(\begin{array}{l} \text{extcolor} \left[\text{exthilight} \right] \\ \text{DEFault} \left[\text{NONE} \right] \end{array} \right) \\ \left(\begin{array}{l} \text{exthilight} \left[\text{extcolor} \right] \\ \text{NONE} \left[\text{DEFault} \right] \end{array} \right) \end{array} \right) \dots$
--------	--

where:

area

specifies the area of the screen. The area can be:

<i>Area</i>	<i>Definition</i>
ALL	the entire screen. If you specify this operand, you cannot specify any other area on the same command.
INArea	the input area.
STAtarea	the system status area.
OUTArea	the output area. If you specify OUTAREA, you cannot specify CPOUT, VMOUT, or INREDISP on the same command line.
CPOut	is output from the VM/XA Control Program. You may not specify CPOUT on the same command as OUTAREA.
VMOut	is output from CMS or the virtual machine operating system running in your virtual machine. You may not specify VMOUT on the same command line as OUTAREA.
INRedisp	is your input redisplay. Messages are displayed with the same characteristics specified for the input redisplay area. You may not specify INREDISP on the same command line as OUTAREA.

extcolor

specifies the color you want to assign to an area of your screen. The color can be:

BLUe	TURquois
RED	PINK
GREen	WHItE
YELlow	DEFault (green and white)

exthilight

specifies the extended highlight value you want to assign to an area of your screen. The exthilight value can be:

BLInk ----	blinking
REVvideo -	reverse video
UNDErlin -	underlining
NONE -----	no extended highlighting

SCREEN

Usage Notes

1. If you issue the SCREEN command from a display that does not support extended color or extended highlighting, CP still sets the values you specify for the virtual machine console. If you disconnect from that display and logon at a display that does support extended color or extended highlighting, the values you set are in effect for your virtual machine after the reconnection. These values remain in effect for the duration of the current session unless you alter them with another SCREEN command.
2. You cannot specify a particular 'area' operand more than once on the same SCREEN command line.
3. Since the command is not device dependent you may issue the SCREEN command from any IBM-supported terminal or from a PROFILE EXEC. However, the terminal controller must have the extended color feature installed for the SCREEN command to have an effect on your display.
4. If you issue the SCREEN command for a particular area and you specify only the "extcolor" or default option, the "exthiligh" will remain unchanged from its previous setting. Likewise, if only the "exthiligh" or NONE option is specified, the "extcolor" will remain unchanged from its previous setting.

Example:

If you enter:

```
SCREEN OUTAREA RED NONE INAREA BLINK GREEN STATAREA PINK UNDERLIN
```

CP sets the following values for your terminal:

Area	Color	Exthiligh
cpout }	red	none
vmout } outarea	red	none
inredisp }	red	none
inarea	green	blinking
statarea	pink	underlining

Responses

```
COMMAND COMPLETE
```

indicates that the command has been completed.

SEND**Privilege Class: G**

Use the **SEND** command to enter console input to a disconnected virtual machine for which you provide secondary console services. You may enter CP commands, virtual machine commands, responses to virtual machine or CP messages, or other virtual machine data as required.

SEND	[CP] userid [text]
------	----------------------

where:

CP

indicates that CP should execute the input text on behalf of the receiving virtual machine. If the first four characters of the text you want to send are your logical escape character, the receiver's logical line end character, and "CP" then you need not specify the CP operand to send a CP command. The same holds true if the receiver's virtual machine is already in CP READ. If you specify the CP operand with no text, the receiving virtual machine is placed in console function mode, just as if a "#CP" command was entered.

userid

represents the disconnected virtual machine that has a specified secondary user in its directory entry.

text

can be any command allowed in the virtual machine command environment or a reply to a prompt issued by the disconnected virtual machine. CP passes the text to the disconnected virtual machine as console input.

To send multiple lines in *text*, you must use the logical line end character immediately preceded by the logical escape character; in addition you must have line editing enabled for your virtual machine. Also, note that the logical escape and line end characters must be those line editing characters defined for your virtual machine (the SEND issuer), rather than those defined for the receiving virtual machine.

Usage Notes

1. Use a "SEND CP userid text" command to reply to message HCPQSC0150A USER userid HAS ISSUED A CP READ.
2. Use a "SEND CP userid" command to force a CP READ when no CP READ exists.
3. Use a "SEND userid text" command to reply to message HCPQSC0150A USER userid HAS ISSUED A VM READ.
4. The receiving virtual machine must be disconnected at the time you issue the SEND command.
5. You must be authorized as the secondary user on the CONSOLE statement in the directory entry of the receiving virtual machine.
6. If both a CP and a VM read are pending the CP read is satisfied first regardless of the order in which they were posted.

SEND

Examples

1. Issuing the command "SEND CP USER1" places the disconnected user, USER1, into console function read.
2. Issuing the command "SEND USER3 MSG USER2 HI" would cause USER2 to receive the message MSG FROM USER3 HI.
3. If the secondary user received the message USER5 CHANGE TOD CLOCK (YES|NO) and responded by issuing the command "SEND USER5 yes" the receiver's virtual machine would accept this input as if it was entered from his or her own terminal.

Responses

None.

SET

Use the SET command described in this section to control various functions associated with your virtual machine or with the VM/XA SP system. The format box below shows all of the operands available with the SET command. Complete descriptions of the operands appear on the following pages, in the same sequence as the operands appear here.

SET	ABEND { HARD HARD aaannn... SOFT }	B
	ACCount nnnnnnnn	G
	AUTOPoll [ON] [OFF]	G
	CACHE { DEVICE } { ON } [FOR] { rdev SUBSYSTEM } { OFF } { rdev ... (rdev-rdev)	B
	CCWtran { OFF } { ON }	G
	CMDLIMit [XAUTOLog] [nnn] [AUTOLog] [4]	B
	CONCeal { ON } { OFF }	G
	CPCONIo { IUCV } { OFF }	G
	CPLANGUage langid	G
	CPTRace [[{FOR NONSPECIFIC}] [ON] [ALLcodes] [more] FOR { SPECIFIC } [OFF] [code1 code2 ...] { SYSTEM } ALL userid]] [FOR { SYSTEM } RESET { SPECIFIC } userid]]	A,C
	CPUid ssssss	G
	DUMP { DASD } [IPL] [CP] { rdev } [NOIPL] [ALL] [V=R]	B

SET

SET	EMSG { ON } { CODE } { TEXT } { OFF } { IUCV }	G
	IMSG { ON } { OFF } { IUCV }	G
	IOAssist { ON } [[FOR] ALL] { OFF } [[userid] *]	B,G
	LINEDit { ON } { OFF }	G
	LOGmsg [nn [[*]text]] [NULL] [NULL]	B
	MACHine { 370 } { XA }	G
	MAXUsers { nnnnn } { OFF }	A
	MIH { ON } { OFF }	G
	MITIME { DASD } { TAPE } { GRAF } { UR } { MISC } { rdev } { rdev-rdev } { OFF }	A,B
	[DASD] [TAPE] [GRAF {mm:ss} ...] [UR {mm:ss}] [MISC {OFF}]	

SET	MODE [RETRY] {Quiet } {Record }	A,F
	MONData { ON } { OFF }	C
	MSG { ON } { OFF } { IUCV }	G
	NOPDATA { ON } { OFF }	G
	NOTRANS { ON } { OFF }	G
	PAGEX { ON } { OFF }	G
	PASSWORD { LINK {SEPARATE } {INCLUDE } } { LOGON } { XAUTOLOG } { AUTOLOG }	B
	PFnn [SUBSTITU] [IMMed] [pfdata1#pfdata2#...pfdata n] [DElAy] [NODisp]	G
	PFnn COPY [rdev] [ldev] [luname]	G
	PFnn RETrieve {CLEAR}	G
	PFnn TAB n1 [n2 ...]	G
	QUICKDsp userid [ON] [OFF]	A
	RECORD { ON } rdev LIMIT nn BYTE nn BIT n [{AND } BYTE nn BIT n] { OFF } [{OR }]	F

SET

SET	RESERVED {userid} {nn} { * } { OFF }	A
	RUN { ON } { OFF }	G
	SHARE userid { ABSolute nnn% } { RELative nnnn } { INITial }	A
	SHARED { ON } { OFF } [FOR] { rdev } { rdev-rdev }	B
	SMsg { ON } { OFF } { IUCV }	G
	SRM { IABias [nnn[%] mmm] } { <u>90</u> } { <u>2</u> } { INITial }	A
	DSPSlice minslice	
	STORBuf { aaa[%] bbb[%] ccc[%] } { INITial }	
	LDUBuf { ddd[%] eee[%] fff[%] } { INITial }	
	DSPBuf { iiii jjjjj kkkk } { INITial }	
	SVC76 { CP } { VM }	G
	TIMER { ON } { OFF } { REAL }	G
	VMCONIo { IUCV } { OFF }	G
	WNG { ON } { OFF } { IUCV }	G

General Usage Notes for SET

Use the CP QUERY SET command to determine the current settings of the following SET command functions:

SET AUTOPOLL
 SET CCWTRAN
 SET CONCEAL
 | SET CPCONIO
 | SET ECMODE
 SET EMSG
 SET IMSG
 SET IOASSIST
 SET LINEDIT
 SET MACHINE
 SET MIH

SET MSG
 SET NOPDATA
 SET NOTRANS
 SET PAGEX
 SET RUN
 SET SMSG
 SET SVC76
 SET TIMER
 | SET VMCONIO
 SET WNG

Use the CP QUERY command with the proper operand to determine the settings of the following SET command functions:

SET ABEND
 SET ACCOUNT
 SET ACNT
 SET CACHE
 | SET CMDLIMIT
 SET CPTRACE
 SET CPUID
 SET DUMP
 SET LOGMSG
 SET MAXUSERS
 SET MITIME

SET MONDATA
 SET PASSWORD
 SET PF
 SET QUICKDSP
 SET SHARE
 SET SRM (DSPBUF)
 SET SRM (DSPSLICE)
 SET SRM (IABIAS)
 SET SRM (LDUBUF)
 SET SRM (STORBUF)

SET ABEND

Privilege Class: B

Use SET ABEND to force specific soft abend codes, or all soft abend codes, to cause full system termination.

SET	ABEND { HARD HARD aaannn... SOFT }
-----	---

where:

ABEND

indicates that you want to change system processing for soft abends.

HARD

specifies that any soft abend will cause full system termination without attempting system recovery.

HARD aaannn

specifies the soft abend with abend code aaannn will cause full system termination, but other soft abends will continue to be processed normally.

You can specify up to 10 unique soft abend codes with one or more invocations of the SET ABEND command. The codes will be inspected for format validity, that is, three letters followed by three numbers. The nnn portion of the code is required to be less than or equal to decimal 255. The code will not be inspected to determine whether it corresponds to a soft abend that is actually present in the system.

SOFT

cancels all previous specifications of HARD and HARD aaannn. All soft abends operate normally as established during system initialization.

Usage Notes

If any invalid abend code is encountered in SET ABEND HARD aaannn aaannn..., no soft abend settings are changed.

Responses

None.

SET ACCOUNT

Privilege Class: G

Use SET ACCOUNT to change the current session account number.

SET	ACCount nnnnnnnn
-----	------------------

where:

nnnnnnnn

is the 1- to 8-character account number that will replace the current session account number.

Usage Notes

1. The account number you specify on the command line must be the same as one contained in your entry in the system directory.
2. If the account number you specify is valid, an accounting record will be created for your session under the current accounting number.

Responses

None.

SET AUTOPOLL

Privilege Class: G

SET AUTOPOLL specifies how CP detects changes to virtual autopolling channel programs: whether VM/XA SP itself tests for the change or whether your virtual machine notifies VM/XA SP of the change. In general, the guest operating system that notifies CP of changes to virtual autopolling channel programs issues this command without the need for any action or intervention on your part.

SET	AUTOPOLL [ON] [OFF]
-----	----------------------------

where:

AUTOPOLL [ON]
[OFF]

controls whether CP tests for changes to virtual autopolling channel programs. ON indicates that your virtual machine must notify CP when a change is made to a virtual autopolling channel program. OFF indicates that CP tests for changes to virtual autopolling channel programs.

Usage Notes

1. In general, do not issue this command yourself. Guest operating systems that notify CP of changes to virtual autopolling channel programs usually issue this command themselves internally. If you do issue the command while a guest operating system is running, you may cause the system to stop or to terminate abnormally.
2. If SET AUTOPOLL ON is specified for your virtual machine, the operating system running in your virtual machine must use DIAGNOSE code X'28' to notify VM/XA SP whenever a virtual autopolling channel program has been modified. If the operating system does not issue a DIAGNOSE code X'28' instruction after the modification, results are unpredictable.
3. If SET AUTOPOLL OFF is specified and the operating system running in your virtual machine uses the DIAGNOSE code X'28' interface, VM/XA SP and your virtual machine are duplicating a function. In this case, VM/XA SP is testing for virtual autopolling channel program modification and at the same time the virtual machine is notifying VM/XA SP whenever modification occurs. Therefore, IBM recommends the combined use of the SET AUTOPOLL ON command and DIAGNOSE code X'28' to eliminate this unnecessary duplication.

Responses

None.

SET CACHE

Privilege Class: B

Use the SET CACHE command to activate or deactivate caching by device or by subsystem for 3880 Model 13 and 23 subsystems.

SET	CACHE { DEVICE } { ON } [FOR] { rdev { SUBSYSTEM } { OFF } { rdev ... { rdev-rdev }
-----	--

where:

rdev

rdev ...

rdev-rdev

is the real device number, a list of real device numbers, or a range of real device numbers of devices in the subsystem.

**DEVICE { ON }
 { OFF }**

DEVICE ON indicates that caching is to be activated for the specified devices. DEVICE OFF indicates that caching is to be deactivated for the specified devices. Any user to whom the device was attached will bypass the cache.

**SUBSYSTEM { ON }
 { OFF }**

SUBSYSTEM ON indicates that caching is to be made available for the entire subsystem to which the device is attached. SUBSYSTEM OFF indicates that caching is to be made unavailable for the entire subsystem to which the device is attached.

Usage Notes

1. If SET CACHE DEVICE ON is issued for a device, all minidisks that were defined on that device in the user's directory entry with the DASDOPT or MINIOPT statement then have access to the cache. This means that CP allows virtual I/O to these minidisks to pass through the cache.

If SET CACHE DEVICE OFF is issued for a device, all minidisks that were defined on that device in the user's directory entry with the MINIOPT or DASDOPT statement no longer have access to the cache. This means that CP forces virtual I/O to these minidisks to bypass the cache.

Example

To activate device 340 for caching, enter:

```
SET CACHE DEVICE ON 340
```

Response 1

```
rdev CACHE ACTIVATED FOR DEVICE
```

This response will be repeated for each specified device when you issue the SET CACHE DEVICE ON *rdev* command.

SET CACHE

where:

rdev

is the real device number of the device that has been given access to the cache.

Response 2

rdev CACHE DEACTIVATED FOR DEVICE

This response will be repeated for each specified device when you issue the SET CACHE DEVICE OFF *rdev* command.

where:

rdev

is the real device number of the device that has been given access to the cache.

Response 3

rdev CACHE UNAVAILABLE FOR SUBSYSTEM

This response will be repeated for each specified device when you issue the SET CACHE SUBSYSTEM OFF *rdev* command.

where:

rdev

is the real device number of the device whose subsystem's cache has been deactivated.

Response 4

rdev CACHE AVAILABLE FOR SUBSYSTEM

This response will be repeated for each specified device when you issue the SET CACHE SUBSYSTEM ON *rdev* command.

where:

rdev

is the real device number of the device whose subsystem's cache has been activated.

SET CCWTRAN

Privilege Class: G

Use SET CCWTRAN to control CCW translation for CP.

SET	CCWtran { OFF } { ON }
-----	---------------------------

where:

CCWTRAN { OFF }
 { ON }

controls CCW translation for CP. If you specify OFF, all virtual I/O from your virtual machine bypasses CP CCW translation except under the following conditions, which force CCW translation:

- Tracing I/O to a device.
- Tracing CCWs for a device.
- CP has sense data pending for a device.
- The I/O device is not dedicated to the preferred virtual machine unless it is a full-pack minidisk and the channel program begins with valid SEEK, SET-FILE-MASK CCWs.
- CP has delayed response messages pending for the device.
- CP is expecting a global CCW to be issued, which must be translated.

When a preferred virtual machine is logged on, the CCWTRAN option is set OFF to bypass CCW translation. Note, however, that if you run VM/XA SP in a virtual machine, a preferred virtual machine guest who logs on to your second-level system will have the CCWTRAN option set ON.

Usage Notes

You can use this command only if your preferred virtual machine runs in the V=R area.

Responses

None.

SET CMDLIMIT

Privilege Class: B

Use SET CMDLIMIT to define a system limit for the number of incorrect passwords a user can enter on the AUTOLOG and XAUTOLOG commands. If a user exceeds this limit, CP disables the AUTOLOG and XAUTOLOG commands for the remainder of the user's logon session.

SET CMDLIMIT	XAUTOLOG [nnn] AUTOLOG [4]
--------------	-----------------------------------

where:

XAUTOLOG nnn

AUTOLOG nnn

sets the system limit for incorrect passwords entered on the AUTOLOG and XAUTOLOG commands. The maximum limit is 255; the minimum limit is 1. The limit is set to 4 (the initial system setting) if nnn is omitted.

Usage Notes

1. For security reasons, CP limits the number of incorrect passwords allowed on the AUTOLOG and XAUTOLOG commands in each user's logon session. The SET CMDLIMIT lets you raise or lower that limit.
2. Use the QUERY CMDLIMIT command to determine the current system limit value.
3. AUTOLOG is an accepted synonym for XAUTOLOG. The minimum abbreviation is AUTOL.

Responses

None.

Migration Notes

See the XAUTOLOG and AUTOLOG commands.

SET CONCEAL

Privilege Class: G

Use the SET CONCEAL command to activate and to deactivate the protected application environment for a virtual machine, thus protecting a session from unexpectedly entering CP READ.

SET	CONCeal {ON } {OFF }
-----	-------------------------

where:

ON

specifies that the virtual machine is to operate in the protected application environment. This environment is intended for the application end user who does not want to interact with CP. Instead of presenting the user with an unexpected CP READ on occurrence of certain error conditions (for example, paging error, soft abend, disabled wait PSW loaded), the protected application facility forces an automatic re-IPL of the virtual machine without issuing any error messages to the user. The IPL is initiated using the IPL command last used for the virtual machine.

To prevent a re-IPL loop, CP does not force re-IPL if a previous automatic re-IPL has occurred less than a minute before. No more than 10 automatic re-IPLs are attempted between activation and deactivation of the protected application environment. If a re-IPL is not attempted or if the attempt fails, CP issues the same message as when the protected application environment is not active and puts the terminal into CP READ state.

When you SET CONCEAL ON, the terminal break key is disabled. If you subsequently deactivate the protected application environment using the SET CONCEAL OFF command, the break key is reenabled. You may also enable the break key by issuing a TERMINAL BRKKEY command while in the protected application environment.

OFF

specifies that the virtual machine should no longer operate in the protected application environment. Certain error conditions (such as a paging error, a soft abend, or a disabled wait PSW in the virtual processor) cause CP to take control of the virtual machine and display CP READ on the terminal. No re-IPL is initiated. The terminal break key is reenabled when you issue SET CONCEAL OFF.

Usage Notes

1. With your break key disabled, you cannot enter the CP READ state while running in full screen mode unless your guest (operating system or application) provides a way. The guest can use DIAGNOSE code X'08' to pass either the CP command or no command at all to enter CP READ. In addition, the TERMINAL MODE CP command allows entry to CP on a console attention interrupt when in line mode.
2. An application can establish the protected application environment for itself by issuing a SET CONCEAL ON command using DIAGNOSE code X'08'.

SET CONCEAL

3. You may not specify SET CONCEAL ON if you have defined a multiprocessor configuration. The protected application facility only supports virtual uniprocessor systems.
4. If you use an application which runs with the protected application environment active, be aware that an automatic re-IPL will reset your virtual machine. If you IPL a named saved system, storage is also cleared; therefore, data in virtual machine storage is lost.
5. If the protected application environment is already active, the SET CONCEAL ON command has no effect. Thus, if the break key has been reenabled via the TERMINAL BRKKEY command, this second SET CONCEAL ON command does not disable the break key.

Responses

None.

Migration Notes

VM/SP HPO: For SET CONCEAL OFF, VM/XA SP resets the break key to the last key specified on the TERMINAL BRKKEY command, rather than unconditionally resetting it to the PA1 key as in HPO.

SET CPCONIO

Privilege Class: G

Use the SET CPCONIO command to control the destination of all CP output to the virtual machine console. At LOGON, CPCONIO is set OFF.

SET	CPConio {IUCV} {OFF }
-----	--------------------------

where:

IUCV

specifies that responses are to be passed to the virtual machine via IUCV if a connection to the message system service exists. If the console is being spooled, the responses are also placed in the console spool file as well as being sent via IUCV. If no IUCV connection exists, data will be handled as if SET CPCONIO OFF had been issued.

OFF

specifies that responses are to be displayed at the terminal.

Usage Notes

None.

Responses

None.

SET CPLANGUAGE

Privilege Class: G

Use the SET CPLANGUAGE command to change the current language of your virtual machine.

SET	CPLANGUage langid
-----	-------------------

where:

langid

is the language identifier of the language to which the virtual machine is to be set. *langid* may be 1-to-5 characters in length and must consist of CMS file system characters only.

Usage Notes

1. Be aware of the language-related terminal restrictions when you specify a language. For example, you may have your terminal set to display the German character set; if you then set your language to Spanish while using the same terminal, you may get unexpected characters on your screen or, in the worst case, CP may disconnect you. If you are not sure what characters your terminal can display, check with your system administrator.
2. The QUERY CPLANGLIST command displays all the valid language ids that you can set in CP. To display the language that is currently active in your virtual machine, issue the QUERY CPLANGUAGE command. Contact your system administrator if you have any questions about the languages available on your VM/XA SP system.

Responses

None.

Examples

If your virtual machine is set to American English and you want to change your language to French, issue the command:

```
SET CPLANGUage FRANC
```

To change it back to American English, issue the command:

```
SET CPLANGUage AMENG
```

SET CPTRACE

Privilege Class: A, C

Use SET CPTRACE to activate or deactivate CPTRACE entry recording of real machine events in the CP TRACE tables.

SET	<p>CPTRACE $\left[\left[\text{FOR } \left(\begin{array}{l} \text{NONSPECIFIC} \\ \text{SPECIFIC} \\ \text{SYSTEM} \\ \text{ALL} \\ \text{userid} \end{array} \right) \right] \left[\begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right] \left[\begin{array}{l} \text{ALLcodes} \\ \text{code1 code2 ...} \end{array} \right] [\text{more}] \right]$</p> <p>$\left[\text{FOR } \left(\begin{array}{l} \text{SYSTEM} \\ \text{SPECIFIC} \\ \text{userid} \end{array} \right) \text{ RESET} \right]$</p> <p>where [more] is:</p> <p>$\left[\left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\} \left[\begin{array}{l} \text{ALLcodes} \\ \text{code1 code2 ...} \end{array} \right] \right] [\text{more}]$</p>
-----	---

where:

CPTRACE

enables or disables tracing of real machine events in the CP trace tables.

FOR NONSPECIFIC

specifies the target for the subsequent tracing changes.

NONSPECIFIC indicates the specified tracing applies to all virtual machines not specifically set or named by a previous SET CPTRACE command.

NONSPECIFIC is the default. All virtual machines begin in the NONSPECIFIC group. All members of the NONSPECIFIC group have the same codes traced.

FOR $\left(\begin{array}{l} \text{SPECIFIC} \\ \text{SYSTEM} \\ \text{ALL} \\ \text{userid} \end{array} \right)$

specifies the target for the subsequent tracing changes.

SPECIFIC indicates the specified tracing applies only to those virtual machines specifically set or named by a previous SET CPTRACE command. Members of the SPECIFIC group may each have a unique set of codes traced.

SYSTEM indicates the specified tracing applies to system work. SYSTEM becomes a member of the SPECIFIC group, even though the specified tracing may not match the virtual machines already in the group. Therefore, SYSTEM is similar to "userid" in that it names the work for which tracing applies, and SYSTEM can also be a part of either the SPECIFIC or NONSPECIFIC groups but not both.

ALL indicates the specified tracing applies to both the SPECIFIC and NONSPECIFIC groups of virtual machines and to system work.

userid indicates the specified tracing applies only to this specific virtual machine. userid becomes a member of the SPECIFIC group, even though the specified tracing may not match others in the group.

RESET

RESET ends specific tracing for the indicated virtual machine, system work, or the entire SPECIFIC group. The virtual machine or machines (and/or SYSTEM) become members of the NONSPECIFIC group. SET CPTRACE FOR SPECIFIC RESET, therefore, ensures that all virtual machines and SYSTEM are in the NONSPECIFIC group.

ON

OFF

ON, the default, enables tracing of the specified categories. OFF disables tracing of the specified trace categories. ON and OFF may be intermixed in the same trace category list to enable and disable trace categories in the same command. The trace category list is processed from left to right.

ALLCODES

ALLCODES, the default, specifies all trace codes.

code1 code2 ...

is either a single 4-digit trace code or the name of a category of codes. The system displays only the setting of this code or category of codes.

The following list describes the major (second-level) code category names in the system hierarchical order:

- EXT** External interruption activities
- SVC** Supervisor call interruption related activities
- PRG** Program interruption entries
- MCH** Machine check interruption related activities
- I/O** Input/output interruption related activities
- STOR** Real storage maintenance related activities
- SCH** Adding/dropping users to or from a scheduler queue
- IOS** I/O instructions (including real, logical device, and I/O assist)
- IOR** Stacking/unstacking of IORBKs or TRQBKs
- CPX** Stacking/unstacking of CPEBKs and related activity
- RUNU** Virtual machine dispatching activity
- VINT** Simulated I/O interruptions
- VCSW** Simulated I/O instructions
- IUCV** IUCV activities
- CCS** SNA/CCS activities.

The following lists the code category names and the trace events within these categories also in system hierarchical order (for a full numerical list regardless of category, refer to *VM/XA SP Diagnosis Reference*).

ALLCODES

Sets or resets tracing of all events and is the default option.

EXT

Sets or resets tracing of external interruption related activities:

- 0100 External interruption
- PCDIAG Processor controller (P.C.) diagnose
 - 3C00 P.C. diagnose request started
 - 3C55 P.C. diagnose request returned
- PCSERVC Processor controller (P.C.) service call
 - 3C10 P.C. service call request started
 - 3C65 P.C. service call request returned

3CFF Unsolicited processor controller interrupt received

AE00 SIGP instruction

SVC

Sets or resets tracing of supervisor call interruption related activities:

0200 SVC interruption

2200 Add or stack a COMBK

2301 Execute CP command

CALLRET (Trace Codes 2800 and 2C00)

2800 - Call-with-save-area

2C00 - Return-with-save-area

3600 Exit to the dispatcher

ABEND (Trace Codes FFEE and FFFF)

FFEE - Resume trace after soft abend

FFFF - Suspend trace during soft abend

PRG

Sets or resets tracing of program interruption entries:

0300 Program interruption

MCH

Sets or resets tracing of machine check interruption related activities:

0400 Machine check interruption

F400 CPU IS check stopped

F401 Check stop CPU recovery

I/O

Sets or resets tracing of input/output interruption related activities:

0500 I/O interruption

10F0 I/O sense data received

STOR

Sets or resets tracing of storage maintenance related activities:

FRE FREE and FRET

0600 - Obtain free storage (FREE)

0700 - Return free storage (FRET)

PFRE Pageable free storage requests

0610- Obtain pageable free storage

0710 - Return pageable free storage

4000 Page translation results

FRAMEREP Available list replenishment

4001 - Demand scan pass 1

4002 - Demand scan pass 2

VMSTOR

4010 - Virtual machine lock storage

4011 - Virtual machine unlock storage

SET CPTRACE

EXTEND Extended free storage
 F700 - Return free storage page
 F701 - Obtain free storage scan

SCH

Sets or resets tracing of adding and or dropping users to or from a scheduler queue:

3200 Add user to dispatch list
3210 Drop user from dispatch list

IOS

Sets or resets tracing of all I/O instructions:

REALIOS - Real I/O

CSCH Clear subchannel
 1000 - CC 0
 1003 - CC 3

HSCH Halt subchannel
 1010 - CC 0
 1011 - CC 1
 1013 - CC 3

MSCH Modify subchannel
 1020 - CC 0
 1021 - CC 1
 1023 - CC 3

SSCH Start subchannel
 1030 - CC 0
 1031 - CC 1
 1033 - CC 3

SSCHSNS Start subchannel (sense)
 1038 - CC 0
 1039 - CC 1
 103B - CC 3

TSCH Test subchannel
 1050 - CC 0
 1051 - CC 1
 1053 - CC 3

RSCH Resume subchannel
 1080 - CC 0
 1081 - CC 1
 1082 - CC 2
 1083 - CC 3

1C01 Unit check
1C02 Valid code for unit exception
1D01 I/O related machine check

CHLCHEK Channel check
 1D02 - Channel check
 1D03 - Channel check at termination

LDIOS - Logical Device I/O

LDCSCH Clear logical subchannel
 8000 - CC 0
 8003 - CC 3

LDHSCH Halt logical subchannel
 8010 - CC 0
 8011 - CC 1
 8013 - CC 3

LDMSCH Modify logical subchannel
 8020 - CC 0
 8021 - CC 1
 8023 - CC 3

LDSSCH Start logical subchannel
 8030 - CC 0
 8031 - CC 1
 8033 - CC 3

LDSSCHS Start logical subchannel (sense)
 8038 - CC 0
 8039 - CC 1
 803B - CC 3

LDTSCH Test logical subchannel
 8050 - CC 0
 8051 - CC 1
 8053 - CC 3

80F0 Logical I/O sense data received
 8500 Logical I/O interruption

IOAIOS - I/O Assist

IOACSCH I/O Assist clear subchannel
 5000 - CC 0
 5003 - CC 3

IOAHSCH I/O assist halt subchannel
 5010 - CC 0
 5011 - CC 1
 5012 - CC 2
 5013 - CC 3

SET CPTRACE

IOAMSCH	I/O assist modify subchannel
	5020 - CC 0
	5021 - CC 1
	5022 - CC 2
	5023 - CC 3
IOASSCH	I/O assist start subchannel
	5030 - CC 0
	5031 - CC 1
	5032 - CC 2
	5033 - CC 3
IOATSCH	I/O assist test subchannel
	5050 - CC 0
	5051 - CC 1
	5053 - CC 3
IOATPI	I/O assist test pending interrupt
	5060 - CC 0
	5061 - CC 1
IOARSCH	I/O assist resume subchannel
	5080 - CC 0
	5081 - CC 1
	5082 - CC 2
	5083 - CC 3
5500	I/O assist I/O interruption

IOR

Sets or resets tracing of stacking/unstacking of IORBKS or TRQBKS:

3000	Stack IORBK/TRQBK
3010	Unstack IORBK/TRQBK

CPX

Sets or resets tracing of stacking/unstacking of CPEBKS and related activity:

3300	Stack CPEBK
3310	Unstack CPEBK
3700	Stack work bits

RUNU

Sets or resets tracing of virtual machine dispatching activity:

0A00	Run user
0A10	Load/store Vector Facility
VSIE	Virtual SIE
	1A00 - Run user in virtual SIE
	1A11 - Virtual SIE interception
3500	Interruption, not instruction
3504	Instruction/interruption

VINT

Sets or resets tracing of simulated I/O interruptions:

0C00	Virtual 370-XA I/O interruption
0D00	Virtual CSW stored
2500	Guest I/O untranslation

VCSW

Sets or resets tracing of simulated I/O instruction:

0C32	Virtual modify subchannel
0C33	Virtual start subchannel
0C35	Virtual test subchannel
0C36	Virtual test pending interruption
0D90	Virtual start I/O
0D91	Virtual start I/O fast

IUCV

Sets or resets tracing of inter-user communication vehicle activities.

1500	IUCV Query function
1501	IUCV Test message function
1502	IUCV Retrieve buffer function
1503	IUCV Describe function
1504	IUCV Send function
1505	IUCV Receive function
1506	IUCV Reply function
1507	IUCV Test completion function
1508	IUCV Reject function
1509	IUCV Purge function
150A	IUCV Accept function
150B	IUCV Connect function
150C	IUCV Declare buffer function
150D	IUCV Quiesce function
150E	IUCV Resume function
150F	IUCV Sever function
1510	IUCV Set mask function
1511	IUCV Set control mask function

CCS

Sets or resets tracing of SNA/CCS activities.

CCSN	CCS normal events
1600	CCS ACCEPT
1603	CCS PURGE
1604	CCS RECEIVE
1606	CCS REPLY
1608	CCS SEND one way
1609	CCS SEND two ways
160A	CCS SEVER
160B	CCS Logic error in CCS interface CB
160C	CCS REPLY from VSM
160E	CCS CONNECT from VSM
1610	CCS SEVER from VSM
1611	CCS Message complete from VSM, one-way SEND

1612 CCS Connect from VSM for LU
 1613 CCS Logic error in VSM interface CB
 1614 CCS Error in user environment, SEVER user
 1615 CCS SNA control block chain invalid

CCSE CCS error events.

1680 CCS ACCEPT error
 1684 CCS RECEIVE error
 1686 CCS REPLY error
 1688 CCS SEND one-way error
 1689 CCS SEND two-way error.

Usage Notes

1. You may combine several commands into one command as long as the FOR qualifier is the same. For example:

```
SET CPTRACE FOR NONSPECIFIC ON SVC OFF EXT
```

Note: FOR NONSPECIFIC is the default value. The following command would have the same result:

```
SET CPTRACE ON SVC OFF EXT
```

2. If you start a SPECIFIC trace, you must be aware of the initial state of the trace. The NONSPECIFIC trace only modifies a trace that has already been started. To find out the initial trace state, issue the following two commands:

```
QUERY CPTRACE FOR SPECIFIC
```

After checking the response, then issue:

```
QUERY CPTRACE
```

This will give you the NONSPECIFIC state.

3. You cannot disable tracing for entries that can occur at the time of a CP trace table full condition. The two entry conditions that can occur are "PROGRAM INTERRUPTION" (0300) for Trace Table Page Full, program interruption X'16', and "TRACE TABLE PAGE FULL DURING MACHINE CHECK HANDLING" (F40F).
4. The SET CPTRACE command processes categories sequentially from left to right; tracing for the first category begins before the second category is processed. Therefore, it may be possible to have trace settings in effect during command processing that are different from the trace settings before and after command processing.

The NONSPECIFIC trace setting, modified by the command initially creating the SPECIFIC trace is the initial state of a SPECIFIC trace setting.

To get an initial state of ALLCODES enabled, issue:

```
SET CPTRACE FOR user ON ALLCODES
```

Combined with or followed by the desired trace settings.

Examples

To disable the trace for IUCV instructions, issue:

```
SET CPTRACE OFF IUCV
```

As an example, assume that when your system is IPLed, all users are in the NONSPECIFIC category. If you want to put all users in the NONSPECIFIC category and you want to trace all trace codes, issue the command:

```
SET CPTRACE FOR NONSPECIFIC ON ALLCODES
```

Issuing the command with all its defaults and abbreviations specified, the command would look like this:

```
SET CPTR
```

If you want to put some users in the SPECIFIC category, issue the following commands:

```
SET CPTRACE FOR NONSPECIFIC OFF ALL
```

This turns off all trace codes for users in the NONSPECIFIC category.

Now, perhaps you want to set a SPECIFIC trace for users CMS000 and SYSTEM; for CMS000 you want to set ON code categories I/O and IUCV, and for SYSTEM you want to set ON only code categories SVC and IUCV. (See trace code category listings beginning on page 510.) To do this, issue the following commands:

```
SET CPTRACE FOR CMS000 OFF ON I/O
```

```
SET CPTRACE FOR SYSTEM OFF ON SVC
```

```
SET CPTRACE FOR SPECIFIC ON IUCV
```

If you now want to put the user CMS000 back into the NONSPECIFIC group, issue the following command:

```
SET CPTRACE FOR CMS000 RESET
```

The system now would be tracing SYSTEM for SVC and IUCV trace code categories.

Note: RESET can only be used to set users or SYSTEM back into the NONSPECIFIC group. In order to set off codes for a NONSPECIFIC user, issue the command:

```
SET CPTRACE FOR userid OFF {code or code category}
```

If you want to trace ALLCODES for one SPECIFIC user (a userid) only, issue the following commands:

```
SET CPTRACE FOR ALL OFF ALLCODES
```

```
SET CPTRACE FOR userid ON ALLCODES
```

Responses

None.

SET CPUID

Privilege Class: G

Use SET CPUID to change the processor identification for the virtual processor on which the command is issued.

SET	CPUid ssssss
-----	--------------

where:

CPUID ssssss

changes the virtual processor identification. The six hexadecimal digits you specify in ssssss become the new processor identification number or processor serial number.

Usage Notes

The SET CPUID command changes bits 8-31 of the CPU ID that is returned by the STIDP instruction to the value you specify in the SET CPUID command line.

Responses

None.

SET DUMP

Privilege Class: B

Use SET DUMP to designate the unit that is to receive a VM/XA SP system abend dump.

SET	DUMP	[DASD _rdev]	[IPL NOIPL]	[CP ALL V=R]
-----	------	-----------------	----------------	--------------------

where:

DUMP

indicates that you are designating the unit that will receive CP ABEND dumps.

DASD

specifies that the system dump unit is a disk. The DASD dump space will be released when the dump setting is changed. Spooling cylinders will be allocated for the dump on the first CP-owned DASD device that has enough contiguous spooling cylinders to hold a complete dump of real storage. The CP-owned DASD are searched in the following order:

- 3340
- 3330
- 3350
- 3375
- 3380
- 2305

rdev

is the real device number of a real printer, a fully supported tape device, or a CP-owned DASD device.

IPL**NOIPL**

indicates whether the system is to be automatically restarted after a failure. At system initialization, the value is IPL; if you don't want your system automatically restarted, you must explicitly change the setting to NOIPL.

CP**ALL****V=R**

indicates which pages of storage are to be dumped. Specify CP if only storage locations occupied by the control program are to be dumped. Specify ALL if you want all real storage dumped. At system initialization, the value is CP. Specify V=R if you also want to dump V=R storage. V=F storage is part of the V=R region generated at system generation time when the V=R operand is specified.

Usage Notes

1. The basic format of the dump to tape is the same as the DASD format. Multiple volume tapes are allowed as dump tapes.
2. Dumps produced by VM/XA SP have a different format than and are incompatible with dumps produced by VM/SP.

SET DUMP

3. Your installation's system security can be compromised if you use the SET DUMP ALL command, because user pages will be included in your dumps.
4. You can specify the operands in any order in the SET DUMP command.
5. If you do not specify a dump device on the command line, the dump device setting will not be changed.
6. Whenever you issue the SET DUMP command, the CP and IPL defaults are in effect unless you explicitly change them. For example, if the current dump options are ALL and NOIPL, and you issue:

```
SET DUMP 000E
```

the dump options will be set to CP and IPL.
7. When the system is initialized, the dump device is set to DASD if there is enough spooling space on a CP-owned DASD to hold it, and if your VM/XA SP system is not running in a virtual machine. Upon initialization, the dump options will be set to CP and IPL.
8. If you want to preserve a preferred virtual machine's environment if VM/XA SP terminates because of a system incident:
 - DO NOT specify the ALL or NOIPL operand on the SET DUMP command
 - Select either a DASD or a tape drive as your system dump device. If you specify a tape drive you must keep that drive ready at all times.

Responses

$$\left\{ \begin{array}{l} \text{type rdev} \\ \text{PRINTER} \end{array} \right\} \text{ DUMP UNIT } \left\{ \begin{array}{l} \text{CP} \\ \text{ALL} \\ \text{V=R} \end{array} \right\} \left\{ \begin{array}{l} \text{IPL} \\ \text{NOIPL} \end{array} \right\} [\text{CYL nnn}]$$

where:

type rdev

is the device type and the real device number of your dump device. The type can be DASD, TAPE, or PRINTER.

PRINTER

indicates that no dump device has been assigned and that there is insufficient spool space on the assigned DASD device. The dump device defaults to the first printer available in the order in which printers were defined at SYSGEN.

CP

ALL

V=R

indicates whether CP-owned pages and free storage or all pages of real storage will be dumped.

IPL

NOIPL

indicates whether the system will automatically restart when the dump is completed.

CYL nnn

is the number of cylinders of spooling space allocated for system dumps. This will be displayed for DASD dumps only.

Migration Notes*VM/SP HPO*

1. In VM/XA SP, use the DASD operand instead of the AUTO operand as in VM/SP HPO.
2. VM/XA SP provides a response to this command whereas VM/SP HPO does not.
3. Dumps produced by VM/XA SP have a different format than and are incompatible with dumps produced by VM/SP.

SET EMSG

Privilege Class: G

Use SET EMSG to control error message handling.

SET	EMSG { ON CODE TEXT OFF IUCV }
-----	---

where:

EMSG {

ON

CODE

TEXT

OFF
}

controls error message handling. If you specify ON, both error code and error message text are displayed on your display screen. When you log on, EMSG is set to ON.

If you specify CODE, only the error code is displayed.

If you specify TEXT, only the message text is displayed.

If you specify OFF, no error messages are displayed.

EMSG {IUCV}

specifies that both the error code and text are to be passed to the virtual machine via IUCV if a connection to the message system service exists. If no IUCV connection exists, the message is handled as if SET EMSG ON had been issued.

Usage Notes

1. If you specify the TEXT option, the number of the message is not displayed. The lack of a message number makes it difficult to locate the message in the *VM/XA SP System Messages and Codes Reference*. Therefore, use the TEXT option with caution.
2. If your console is being spooled, the full error message (both code and text) appears in the spooled output regardless of the setting.
3. CMS recognizes EMSG settings for all error (E) and information (I) messages, but ignores the EMSG setting and displays the complete message (error code and text) for all other system messages.

Responses

None.

Migration Notes

VM/SP HPO: In VM/XA SP, the default value when you log on is different. In VM/XA SP, the default is ON; in VM/SP HPO the default is TEXT.

SET IMMSG

Privilege Class: G

Use SET IMMSG to control whether certain informational responses issued by the CP CHANGE, CLOSE, DEFINE, DETACH, IPL, ORDER, PURGE, SPOOL and TRANSFER commands are displayed on your display screen. The descriptions of these CP commands tell which responses are affected.

SET	IMSG { ON OFF IUCV }
-----	----------------------------

where:

IMSG { ON
OFF }

controls whether the informational responses are displayed. If you specify ON, the responses will be displayed; if you specify OFF, the responses will not be displayed.

IUCV

specifies that informational messages are to be passed to the virtual machine via IUCV if a connection to the message system service exists. If no IUCV connection exists, the message will be handled as if SET IMMSG ON had been issued.

Usage Notes

The SET IMMSG ON or OFF command has no effect on the handling of error messages set by the SET EMSG command. When you log on, IMMSG is set ON.

Responses

None.

SET IOASSIST

Privilege Class: B,G

Use SET IOASSIST to either allow or prevent CP from using the start interpretive execution assist for a preferred virtual machine guest.

SET	IOAssist { ON } { OFF }	[[[FOR] ALL userid *]]
-----	----------------------------	---------------------------------

where:

IOASSIST { ON }
{ OFF }

turns on or off CP's capability to use the start interpretive-execution assist for a preferred virtual machine guest.

ALL

enables or disables the start interpretive-execution assist eligibility for all V=R and V=F users, independent of the local setting for these individual users. The local setting is not changed.

When ALL is specified by a class B user, access to the start interpretive-execution assist is allowed or disallowed for all V=R and V=F guests. These include any new V=R or V=F guests who may log on. When this global setting is OFF, it overrides any individual userid setting.

userid

*

enables or disables the individual user's local IOASSIST passthrough value.
* is the default and specifies the issuer of this command.

Usage Notes

1. If you set IOASSIST either ON or OFF while the preferred guest is running, you may cause CP to reflect error status to the guest for some of the guest's devices.
2. If IOASSIST is set ON for all users, an individual user can set IOASSIST OFF without changing the setting for other users.

Note: The reverse is not true; the ALL setting of OFF overrides local user settings.

Responses

None.

SET LINEDIT

Privilege Class: G

Use SET LINEDIT to control logical line editing functions.

SET	LINEDIT { ON } { OFF }
-----	---------------------------

where:

LINEDIT { ON }
{ OFF }

controls the logical line editing functions for your virtual machine. If you specify ON, the system line editing functions and symbols will be used to edit virtual console input requests. If you specify OFF, no character or line editing will be used for the virtual machine operating system. When you log on, LINEDIT is set ON.

Usage Notes

None.

Responses

None.

SET LOGMSG

Privilege Class: B

Use SET LOGMSG to set the log message line that you specify to the text you enter on the command line, to delete a specified line from the log message, or to delete the entire existing log message.

SET	LOGmsg	nn	[[*]text NULL]
		NULL	

where:

nn

is the number of the line in the log message to be changed. nn can be a value from 1-50.

[*]text

is the text that you want to have at line nn of the log message. If you want the line to be automatically displayed on each user's console when that user logs on to VM/XA SP, the text must be preceded by an * (asterisk). The maximum number of characters that you can include in a line is 76, including blank spaces and the asterisk, if specified.

If line nn already has text, it is replaced. Otherwise, the text is inserted for line nn.

NULL

indicates that the current text for line nn is to be deleted from the log message. If NULL is not preceded by a line number, the entire log message is deleted.

Usage Notes

1. If you do not specify any options following SET LOGMSG, the system prompts you to enter a line of text. This line of text is inserted into the log message after the last line that already has text. For example, if the highest line number in the log message to have text is 37, the new line of text will be inserted at line 38. If line 50 already has text defined, you receive an error message and the text is not inserted. If you want the text to be displayed to users when they log on, the first character must be an * (asterisk).
2. The maximum number of lines you can include in your log message is 50, including blank lines, and thus the highest line number in the log message is 50.
3. If you specify a line number with nothing following it (for instance, SET LOGMSG nn), the system prompts you to enter a line of text. The text entered is placed in the log message at the line specified in the command. If you press ENTER without entering any text, the line is deleted from the log message. If you want the text to be displayed to users when they log on, the first character must be an * (asterisk).
4. If the text of a line in the log message does not start with an * (asterisk), the user must issue the CP QUERY LOGMSG command to see the text. (The text is not displayed automatically when the user logs on to the system.)
5. If more than 76 characters of text are entered for a line, the text is truncated after the 76th character without any message being issued.

6. The LOGMSG area is checkpointed at system termination.

Responses

LOGMSG:

is the prompt to enter a line of text when either SET LOGMSG or SET LOGMSG nn is entered. Type the desired line of text in response to the prompt and press the ENTER key. If you want the text displayed when users log on, the first character must be an * (asterisk).

Migration Notes

VM/SP HPO: VM/XA SP limits you to 50 lines of log message data.

SET MACHINE
Privilege Class: G

Use SET MACHINE to change the type of virtual processor configuration defined for your virtual machine.

SET	MACHine { 370 } { XA }
-----	---------------------------

where:

MACHINE { 370 }
{ XA }

changes the type of virtual processor configuration for your virtual machine. If you specify 370, your virtual configuration must obey the conventions of System/370 architecture. If you specify XA, your virtual configuration must obey 370-XA mode architecture conventions.

Usage Notes

1. The SET MACHINE command terminates all IUCV connections and releases the IUCV buffer. Of course, if you issue the command but it does not change your machine setting from what is used to be, then the IUCV connections and buffer remain as they are.
2. When you change your machine type, your virtual machine is reset as if you had issued a CP SYSTEM CLEAR command. All virtual storage and all registers except control registers are cleared to binary zeros; control registers are reset to their initial values. Therefore, you must repeat your IPL procedure to continue operation.
3. A default architectural type can be specified on the user directory MACHINE statement.

Response 1

STORAGE CLEARED - SYSTEM RESET

indicates that the virtual machine has been cleared.

Response 2

SYSTEM = XA

indicates that the virtual processor is now simulating 370-XA architecture.

Response 3

SYSTEM = 370

indicates that the virtual processor is now simulating System/370 architecture.

SET MAXUSERS

Privilege Class: A

Use the SET MAXUSERS command to adjust or to remove the limit set on the number of users who can log on.

SET	MAXUsers { nnnnn } { OFF }
-----	-------------------------------

where:

nnnnn

is the maximum number of users allowed to log on. The range is from 1 to 99999.

OFF

indicates that an unlimited number of users can log on.

Usage Notes

If the maximum number of users is set to a number lower than the current number of logged on users, the system will allow the users to log off, to be forced off the system, and to disconnect and reconnect, but it will not allow any users to log on until the current number of logged on users is lower than the number specified by MAXUSERS.

Response 1

MAXUSERS = nnnnn

is the normal response for this command.

Response 2

REQUEST ACCEPTED. MAXIMUM USERS VALUE IS CURRENTLY EXCEEDED

is the response if the number of users already logged on exceeds the limit just set.

Response 3

NO MAXUSERS LIMIT IS SET

is the response when SET MAXUSERS OFF is entered or if no user limit is set.

SET MIH**Privilege Class: G**

Use SET MIH to specify whether or not missing interrupt conditions should be handled for I/O operations to the virtual machine's dedicated devices.

SET	MIH { ON } { OFF }
-----	-----------------------

where:

MIH { ON }
{ OFF }

ON indicates that missing interrupt conditions to the virtual machine's dedicated devices will be handled automatically by CP. OFF indicates that the missing interrupt conditions to the virtual machine's dedicated devices will be detected but will not be handled.

Usage Notes

1. SET MITIME OFF will invalidate any setting of the MIH. Missing interrupts can be handled only if they are detected.
2. The default, which is not to handle missing interrupt conditions detected for the virtual machine's dedicated devices, is in effect if MIH is not specified on the user directory OPTION statement or if the SET command is not issued.

Responses

None.

SET MITIME

Privilege Class: A, B

Use the SET MITIME command to alter the time interval at which the specified device is checked for missing interrupts.

SET	MITime $\left\{ \begin{array}{l} \text{DASD} \\ \text{TAPE} \\ \text{GRAF} \\ \text{UR} \\ \text{MISC} \\ \text{rdev} \\ \text{rdev-rdev} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{mm:ss} \\ \text{OFF} \end{array} \right\}$	$\left[\begin{array}{l} \text{DASD} \\ \text{TAPE} \\ \text{GRAF} \\ \text{UR} \\ \text{MISC} \end{array} \right\} \left\{ \begin{array}{l} \text{mm:ss} \\ \text{OFF} \end{array} \right\} \dots$
	{OFF}		

where:

MITIME

indicates that you are changing the time interval for checking for missing interrupts.

DASD

indicates that all real devices that were generated as DASD during system generation are to be monitored using the new time interval. Devices used by the mass storage subsystem (MSS) will not be included.

TAPE

indicates that all real devices that were generated as TAPE devices during system generation are to be monitored with the new time interval.

GRAF

indicates that all real devices that were generated as display devices during system generation are to be monitored with the new time interval, an exception being device TYP328X.

UR

indicates that all real devices that were generated as spooling devices during system generation are to be monitored with the new time interval, an exception being device TYP3800.

MISC

indicates real devices TYP328X, TYP3800 and any mass storage device are to be monitored with the new time interval.

rdev

rdev-rdev

is a real device number or range of real device numbers that are to be monitored with the new time interval.

OFF

indicates that missing interrupts are not to be detected for any devices within the specified range. If you specify this operand without any device, missing interrupt monitoring for all devices will be stopped.

SET MITIME

mm:ss

indicates the rate in minutes and seconds at which the device is to be examined for missing interrupts. The maximum value for mm is 99 and for ss is 59. The value will be rounded up to the next multiple of five seconds.

Usage Notes

1. If VM/XA SP is running virtually, missing interrupt detection will default to OFF at IPL time. The default value will be SET MITIME OFF.

When sharing a display device between a real and a virtual system, the virtual system should leave missing interrupt detection off for the shared display device. Under certain circumstances, if the missing interrupt detector is active in the virtual machine for the shared display device, the virtual machine operator may be logged off.

2. The system default values at IPL time are:

SET MITIME DASD 00:15

SET MITIME GRAF 00:30

SET MITIME UR 01:00

SET MITIME TAPE 10:00

SET MITIME MISC 12:00

3. If you specify a range of devices, rdev-rdev, it must be in ascending sequence; otherwise, the command is invalid.
4. If you specify a device or range of real devices on the command line, and the command syntax is correct, all of the real device numbers will be processed. Error messages will inform the operator of those devices that are not supported. (The supported devices are DASD, TAPE, UR, and DASD used by MSS). The unsupported devices will not be monitored for missing interrupts.
5. If you specify a value for :ss which is greater than 00:59 (that is, one minute), the command will be considered invalid. If you specify 00:00, missing interrupt detection for the devices will be stopped.
6. It may take twice as long as the interval specified to detect a missing interrupt.
7. The most recent interval specified for a device is the interval currently in effect. Keep in mind that when you issue the SET MITIME command for an entire class of devices (such as all DASD devices or all TAPE devices), the system monitors *all* members of that class at the new time interval.

For example, assume that the valid range of DASD device addresses at your installation is 190-400. If you entered:

```
SET MITIME 250-255 00:59
```

and at some later point in time entered:

```
SET MITIME DASD 00:20
```

then the current interval for devices 190-400 is 20 seconds.

8. Changes can be made to several device classes (class) with a single command. The command is processed left-to-right so the time intervals specified last on the command line for a device class will be the interval currently in effect.

9. Only device classes may be repeated on a single command line. If real device number(s) are detected along with device classes on a single command line, the command will be rejected.

Valid examples of multiple requests:

```
SET MITIME DASD 00:30 TAPE OFF UR 01:00 MISC 1:40
SET MITIME TAPE 01:50 DASD 02:35 MISC 02:30 TAPE 00:50
SET MITIME DASD 00:00 UR 05:00 MISC 1:40 DASD OFF
```

Invalid examples:

```
SET MITIME DASD 00:30 TAPE OFF UR 01:00 190 01:40
SET MITIME 225-260 02:35 TAPE 00:50
SET MITIME DASD 01:00 UR 05:00 255 00:30 DASD OFF
```

Responses

None.

Migration Notes

VM/SP HPO

1. VM/XA SP allows you to alter missing interrupt intervals by specifying device addresses as well as by classes.
2. VM/XA SP returns a blank response.

SET (error recording) MODE

Privilege Class: A, F

Use SET MODE to set the error recording mode for system recovery machine checks (soft errors). When the hardware recovers from an error without software assistance, it issues a system recovery machine check interruption to report the incident (provided, of course, that the software has enabled the system to accept such interruptions).

SET	MODE [RETRY] { Quiet } { Record }
-----	--------------------------------------

where:

MODE

indicates that you are setting the error recording mode for system recovery machine checks.

RETRY

is an optional keyword provided for compatibility with prior systems.

QUIET

turns off error recording of system recovery machine checks. The system recovery machine checks will be masked out on all processors so that this class of machine checks will not cause interruptions.

RECORD

re-enables all processors for system recovery machine checks. The next 12 system recovery machine checks will be recorded. After that, the system will automatically return to QUIET mode and you will receive a message informing you of that condition.

When a warning machine check occurs, a message is issued and further warning machine checks are disabled. The system then runs disabled for further warning machine checks until the SET MODE RECORD command is issued.

Usage Notes

In addition to re-enabling system recovery machine checks, you can use this command to re-enable warning machine checks.

Responses

```
SYSTEM RECOVERY { QUIET } MODE ACCEPTED
                  { RECORD }
```

indicates that the command has been accepted.

Migration Notes

VM/SP HPO

1. VM/XA SP does not support the cpuid operand.
2. VM/XA SP provides a response whereas VM/SP HPO does not.
3. VM/XA SP gives a response to SET MODE RETRY QUIET.

SET MONDATA

Privilege Class: C

Use the SET MONDATA command to determine whether or not display devices input or output data is to be included in the monitor records.

SET	MONData { ON } { OFF }
-----	---------------------------

where:

ON

indicates that input from and output to user display devices is to be included with the terminal event records.

OFF

indicates that input from and output to user display devices is not to be included with the terminal event records.

Usage Notes

1. The MONDATA option is set to OFF when the system is IPLed and is not changed unless a user issues SET MONDATA ON.
2. To obtain the setting of the MONDATA option, issue the QUERY MONDATA command.
3. It is the responsibility of the installation to protect the security of this data if MONDATA is set ON.

Responses

COMMAND COMPLETE

is the response when a valid SET MONDATA command is issued.

SET MSG

Privilege Class: G

Use SET MSG to control whether messages sent by other users using the MESSAGE command are to be received at your display.

SET	MSG { ON } { OFF } { IUCV }
-----	-----------------------------------

where:

MSG { ON }
 { OFF }

controls whether the messages are displayed. If you specify ON, the messages will be displayed. If you specify OFF, no messages are displayed. When you log on, MSG is set ON.

IUCV

specifies that messages are to be passed to the virtual machine via IUCV if a connection to the message system service exists. If the messages are passed via IUCV, they are not placed in the virtual console spool file. If no IUCV connection exists, the message will be handled as if SET MSG ON had been issued.

Usage Notes

None.

Responses

None.

SET NOPDATA

Privilege Class: G

Use the SET NOPDATA command to control the simulation of NOP CCWs in channel programs for your virtual printers and punches.

SET	NOPDATA { ON } { OFF }
-----	---------------------------

where:

NOPDATA { ON }
{ OFF }

controls whether NOP CCWs are simulated as data transfer CCWs for the spooled punches and printers of your virtual machine.

If you specify ON and if you specify more than one byte of data, CP will attempt to transfer the data addressed by a NOP CCW to the spool file. If you specify OFF, CP will simulate the NOP the same way a real device and channel would.

The initial setting for a virtual machine is OFF.

Usage Notes

1. This command allows programs which need to imbed control information in CP spool files (such as RSCS) to operate under VM/XA SP.
2. Do not use data chaining with NOP CCWs.

Responses

None.

SET NOTRANS

Privilege Class: G

Use the SET NOTRANS command to control CCW translation for CP. The SET NOTRANS command is provided for compatibility with VM/SP HPO; it provides the same function as the SET CCWTRANS command. (SET NOTRANS ON is equivalent to SET CCWTRANS OFF. Also, SET NOTRANS OFF is equivalent to SET CCWTRANS ON.)

SET	NOTRans { ON } { OFF }
-----	---------------------------

where:

NOTRANS { ON }
{ OFF }

controls CCW translation for CP. If you specify ON, all virtual I/O from your virtual machine bypasses CP CCW translation except under the following conditions, which force CCW translation:

- Tracing I/O to a device.
- Tracing CCWs for a device.
- CP has sense data pending for a device.
- The I/O device is not dedicated to the preferred virtual machine unless it is a full-pack minidisk and the channel program begins with valid SEEK, SET-FILE-MASK CCWs.
- CP has delayed-response messages pending for the device.
- CP is expecting a global CCW to be issued, which must be translated.

When a preferred virtual machine is logged on, the default is to bypass CCW translation (SET NOTRANS ON). Note, however, that if you run VM/XA SP in a virtual machine, a preferred virtual machine guest who logs on to your second-level system will not have CCW translation bypassed (SET NOTRANS OFF).

Usage Notes

You can use this command only if your preferred virtual machine runs in the V=R area.

Responses

None.

SET PAGEX

Privilege Class: G

SET PAGEX is used to activate or deactivate the pseudo page fault facility for your virtual machine. When the facility is activated, your virtual machine can dispatch a task while waiting for CP to satisfy a page-in request for another task. When the facility is deactivated, your virtual machine must wait until CP has satisfied a pending page-in request before it can dispatch another task. In general, the guest operating system running in your virtual machine issues this command without the need for any action or intervention on your part.

SET	PAGEX { ON } { OFF }
-----	-------------------------

PAGEX { ON }
{ OFF }

controls whether the pseudo page fault facility is activated or deactivated for your virtual machine. ON activates the facility. OFF deactivates it.

Usage Notes

1. In general, do not issue this command yourself. Guest operating systems that use the handshaking facility activate it themselves internally, using DIAGNOSE Code X'08'. If you do issue the command while a guest operating system is running, you may cause the system to be stopped or to terminate abnormally.
2. The pseudo page fault facility is a function of the VM/VS handshaking feature, and should be activated only for those operating systems that use the handshaking feature. For an overview of the feature, see *VM/XA SP General Information*. For specific information on which operating systems use the VM/VS handshaking feature, see *VM/XA SP Planning and Administration*. If the pseudo page fault facility is not activated for an operating system that supports the handshaking feature, you may cause performance degradation. On the other hand, if the pseudo page fault facility is activated for an operating system that does *not* support the VM/VS handshaking feature, results are unpredictable.
3. A preferred virtual machine uses first-level storage (the V = R area), which is not subject to paging by CP. Therefore, the pseudo page fault facility does not apply to an operating system running in a preferred virtual machine.
4. After you issue the LOGON command or the IPL command, the pseudo page fault facility is *not* activated. The guest operating system running in your virtual machine reissues the SET PAGEX ON command to reactivate it. The facility must be reactivated after each IPL or LOGON command.

Responses

None.

SET PASSWORD

Privilege Class: B

Use the SET PASSWORD command to indicate whether users of commands that require passwords must enter the passwords separately from the command, or on the command line.

SET	PASSWORD { LINK { SEPARATE } LOGON { INCLUDE } XAUTOLOG AUTOLOG }
-----	--

where:

**LINK { SEPARATE }
 { INCLUDE }**

sets the system-wide LINK command password suppression facility. Specify SEPARATE to indicate that passwords cannot be entered as part of the command line. Specify INCLUDE to indicate that passwords are permitted to be part of the command line.

**LOGON { SEPARATE }
 { INCLUDE }**

sets the system-wide LOGON command password suppression facility. Specify SEPARATE to indicate that passwords cannot be entered as part of the command line. Specify INCLUDE to indicate that passwords are permitted to be part of the command line.

**XAUTOLOG { SEPARATE }
 { INCLUDE }**

**AUTOLOG { SEPARATE }
 { INCLUDE }**

sets the system-wide AUTOLOG and XAUTOLOG command password suppression facility. Specify SEPARATE to indicate that passwords cannot be entered as part of the command line. For XAUTOLOG, specify INCLUDE to permit passwords to be part of the command line. For AUTOLOG, INCLUDE specifies that passwords *must* be part of the command line.

Usage Notes

1. The initial setting for the password suppression facility is SEPARATE.
2. If SET was issued for just one or two of the three commands, the suppression settings that were not reset do not change from their previous values.
3. For security reasons, the password suppression facility permits your installation to require that issuers of commands that require passwords do not put the password in the command when they issue it. Instead, the user is prompted to enter the password, and the input field is masked.
4. The SET PASSWORD command changes the settings for the password suppression facility for the LINK, LOGON, AUTOLOG, and XAUTOLOG commands. The allowable syntax of each of these commands is affected whenever the SET PASSWORD command is used. Depending on what setting is used, the corresponding command syntax may or may not allow the password to be included on the command line. This may affect EXECs that use these commands.

Responses

None.

Migration Notes*VM/SP HPO*

1. The syntax for this command is NOT upwardly compatible with VM/XA SP Release 1. The user must indicate specifically which password suppression facility is to be set.
2. Users migrating from VM/SP or VM/SP HPO must use this command instead of the SYSJRNL macro during installation. This command allows the system administrator to change or set the password suppression facility when running.
3. The default system setting in VM/SP HPO is INCLUDE unless changed by the SYSJRNL macro.

SET PFnn
Privilege Class: G

Use SET PFnn to define a program function for a program function (PF) key on a 3270 or 3290 display station and to indicate when that function is to be executed.

SET	PFnn [SUBSTITU] [IMMed DElayed NODisp]	[pfddata1#pfddata2#...pfdatan]
-----	--	--------------------------------

where:

PFnn

indicates that you are defining a program function. *nn* is a number from 1 (or 01) to 24 that corresponds to a program function key on the display. The program function is a programming capability you create by defining a series of commands you want executed. This series of commands executes when you press the appropriate program function key.

SUBSTITU

substitutes any &n (for example, &1, &2, &3, ...) in the PF data with a token from the input area when the PF key is pressed. &1 is substituted with the first token, &2 is substituted with the second token, and so on. If SUBSTITU is not specified, no substitution is made.

IMMED**DELAYED****NODISP**

indicates when the program function is executed. If you specify IMMED, the program function is executed when you press the PF key. If you specify DELAY, the PF data is displayed in the input area when you press the PF key. The data may then be modified if necessary and the ENTER key pressed when you want the function to be executed. If you specify NODISP, the program function is executed when you press the PF key but the function is not redisplayed on the terminal. DELAY is the default.

[pfddata1#pfddata2#...#pfdatan]

defines the command or data lines that constitute the program function. If you are entering more than one command line as program function data, use the line end symbol to separate the lines. The default line end symbol is the pound sign (#); you can define your own line end symbol with the TERMINAL LINEND command. If you use the line end symbol to separate commands that you want executed with a designated PF key, you must precede each line end symbol with the logical escape character. The default logical escape character is a double quotation mark symbol (""); you can define your own logical escape character with the TERMINAL ESCAPE command. If you do not enter any command lines, the SET PFnn command resets a PF key to undefined or resets the IMMED or DELAY status of an existing PF key function to the value you have specified.

If SUBSTITU is specified, any &n (for examples, &1, &2, ...) in PF data is substituted with a token from the input area. &1 is substituted with the first token, &2 is substituted with the second token, and so on. &* is substituted with all the data past the highest token referenced in PF data.

If no corresponding token is found for an &n, it is translated as a null. If a token is not referenced in the PF data, it is ignored.

A token can be delimited by blanks or single quotes. If you want single quotes imbedded in the token, double the number of quotes.

For example, assume you issue the following command:

```
SET PF1 SUBSTITU MSG &1 PLEASE ATTACH &2 TO ME AS &3, &4
```

If you then type the following in the input area:

```
OPERATOR 135 246 'AS SOON AS POSSIBLE'
```

and press the PF1 key, then the following appears in your input area:

```
MSG OPERATOR PLEASE ATTACH 135 TO ME AS 246, AS SOON AS POSSIBLE
```

Usage Notes

1. For DELAY, if the length of the saved data is longer than the input area, it is truncated to the input area length.
2. For SUBSTITU, specifying a period after a substitution character concatenates the substitution character with the following data. (The data can be text or another substitution character.) For example, assume you set PF2 as follows:

```
SET PF2 SUBSTITU IMMED DISPLAY LT&1..&2
```

If you type the following information on the command line:

```
1fd000 90
```

and press PF2, the following command is executed immediately after being displayed in the output area:

```
DISPLAY LT1fd000.90
```

3. You can use the line end symbol to separate commands that you want to execute with the designated program function key.

For example, suppose your virtual machine has the line editing function set ON, and is using the pound sign # as the logical line end symbol.

Enter the following command while in CMS mode:

```
set linedit off
```

and press the ENTER key.

Then enter:

```
set pf02 immed q rdr#q pun
```

- or -

```
cp set pf02 immed q rdr#q pun
```

and press the enter key.

CP assigns the two QUERY commands as functions of the PF02 key.

Then enter:

```
set linedit on
```

and press the enter key.

Pressing the PF02 key executes the two QUERY commands.

Since the entire input line is accepted as valid program function data, do not specify further SET command functions on the SET PFnn command line.

SET PFnn

4. On a DELAY PF key, if SUBSTITU is not specified the cursor is positioned at the end of the data. If SUBSTITU is specified and all of the &ns are substituted with tokens, the cursor is also positioned at the end. If some &ns are not substituted, the cursor is positioned over the first &n that was not substituted.
5. If you press a PF key that has not been set, the following message is displayed in the input area:
* PFnn UNDEFINED

where nn is the PF key number that was pressed.
6. To use the DELAY function on a 3101, you must be in block mode.
7. On a 3101 device, if you press a PF key that has been defined using SUBSTITU, the &ns are always substituted with nulls.

Responses

None.

SET PFnn COPY

Privilege Class: G

Use SET PFnn COPY to copy your 3270 or 3290 display screen to a printer.

SET	PFnn COPY	[rdev ldev luname]
-----	-----------	----------------------------

where:

PFnn COPY

indicates that the program function key that you specify performs the COPY function for your 3270 or 3290 display. If your keyboard has 24 PF keys, specify the PF key number, *nn*, as a value from 1 (or 01) to 24. If your keyboard has only 12 PF keys, specify *nn* as a value from 1 (or 01) to 12. When you press the PF key, the COPY function produces a printed output of the entire screen except for the status field, which is replaced with blanks.

rdev

ldev

rdev represents the real device number of the display printer to be used. *ldev* is the logical device number of the logical display printer to be used.

luname

identifies a printer controlled by the same VTAM service machine that controls the display whose screen is being copied. It is the name from the VTAM definition of the printer as an SNA logical unit. If the luname is omitted, it is assumed that the associated control unit supports the local copy function.

Usage Notes

- After you press the PF key, you receive a NOT ACCEPTED message in the status field of your display if any of the following situations occur:
 - The printer is already busy.
 - The printer is turned off.
 - The printer is operational but has not been varied online or enabled by the system operator.
 - The printer is out of paper or is in any other intervention required condition.
 - The designated device is not a 3284/3286/3287/3288/3289 printer.
 - The SET PFnn COPY command format is invalid.

You may include your own identification on the printed output by entering the data into the user input area of the screen before you press the PF key. The identification appears on the last two lines of the printed copy.

- To ensure the definition of a PF key for the COPY function, you must start with the keyword COPY, followed by at least one blank, followed by the device identifier (rdev, ldev, or luname).
- The existence of more than one data field (rdev, ldev, or luname) following the keyword COPY identifies the string as pfd data rather than a hard copy request.

SET PFnn COPY

4. The following conditions cause the string to be interpreted as PF key data rather than as a hard-copy request:
 - No operand after COPY
 - No rdev or ldev defined between X'0'- X'FFFF'
 - The presence of an operand after rdev, ldev, or luname.
5. SET PFnn COPY is an immediate (nondisplaying) command; its function is executed immediately upon pressing the appropriate program function key. If you insert the keyword DELAY after the PFnn operand, it is ignored; the program function still executes immediately when the PF key is pressed.
6. On real 3284 printers and all logical display printers, a maximum of 1920 bytes are printed. (All logical display printers are defined as 3284 printers.)
7. A copy from a larger display screen to a smaller print buffer may produce unpredictable results.
8. The COPY function is not supported on a 3101 device. The 3101 can have an attached printer and the screen can be copied using the PRINT key.
9. If you press a PF key that has not been set, the following message is displayed in the input area:
 - * PFnn UNDEFINEDwhere nn is the PF key number that was pressed.

Responses

None.

Migration Notes

VM/SP HPO

1. In VM/XA SP you must specify the printer address unless the display whose screen is being copied is controlled by a VTAM service machine.

SET PFnn RETRIEVE

Privilege Class: G

Use this command to assign the RETRIEVE function to a PF key. You may then use this PF key to retrieve the last seven input lines from your command line, one at a time.

SET	PFnn RETrieve {CLEAR}
-----	-----------------------

where:

PFnn RETRIEVE

indicates that the program function key that you specify performs the RETRIEVE function for your virtual machine. If your keyboard has 24 PF keys, specify the PF key number, *nn*, as a value from 1 (or 01) to 24. If your keyboard has 12 PF keys, specify *nn* as a value from 1 (or 01) to 12.

CLEAR

clears the RETRIEVE buffer if already in use.

Usage Notes

1. Once you have defined a RETRIEVE program function key, CP starts saving your input lines. When you press the designated PF key, VM/XA SP redisplay the most current line input in the input area of the screen, so that you can modify and re-enter the data.
2. VM/XA SP keeps track of seven input lines. Any input line that is an exact duplicate of the previous input line is not saved. For security reasons, input lines that are not displayed on the display device, such as passwords, are not saved.
3. When a designated RETRIEVE program function key is first pressed, VM/XA SP redisplay the most current input line. If the key is pressed again, VM/XA SP displays the previous input line. Each time the key is pressed, VM/XA SP steps through the input lines, displaying them one at a time. When the oldest line that has been saved is reached, the most current line is displayed once again.
4. Input lines are kept only when you have a program function key defined to the RETRIEVE function.
5. If the saved line is longer than the input area, the line is truncated to the input area length.
6. Do not use any operands before RETRIEVE. If you do so, CP interprets the RETRIEVE as PF data rather than the RETRIEVE function.
7. To use the RETRIEVE function on a 3101, you must be in block mode.
8. If you press a PF key that has not been set, the following message is displayed in the input area:
 - * PFnn UNDEFINED
 where *nn* is the PF key number that was pressed.

SET PFnn RETRIEVE

Responses

None.

Migration Notes

VM/SP HPO: If you disconnect, VM/XA SP saves the RETRIEVE buffer.

SET PFnn TAB

Privilege Class: G

Use SET PFnn TAB to set values for the TAB function.

SET	PFnn TAB n1 [n2 ...]
-----	----------------------

where:

PFnn TAB

indicates that the program function key that you specify will perform the TAB function for your virtual machine. If your keyboard has 24 PF keys, specify the PF key number, *nn*, as a value from 1 (or 01) to 24. If your keyboard has only 12 PF keys, specify *nn* as a value from 1 (or 01) to 12.

n1 [n2 ...]

are the tab settings that you want to associate with the PF key.

Usage Notes

1. SET PFnn TAB is not supported on a 3101 device. The 3101 has a tab function incorporated in its hardware.
2. SET PFnn TAB is an immediate (nondisplaying) command: its function is executed immediately upon pressing the appropriate program function key. If you insert the keyword DELAY after the PFnn operand, it will be ignored; the program function will still be executed immediately.
3. To insure the definition of a PF key for the TAB function, you must start with the keyword TAB, followed by at least one blank, and follow this blank with at least one tab setting that you want to associate with the PF key. The lack of a tab setting identifies the string 'TAB' as pfddata.

Responses

None.

Migration Notes

VM/SP HPO: Virtual machines designated by the SET FAVORED command (with no percentage value specified) should be considered for the QUICKDSP attribute in the VM/SP system.

SET RECORD

Privilege Class: F

Use SET RECORD to set the recording mode for a device.

SET	RECOrd { ON } rdev LIMIT nn BYTE nn BIT n [{ AND } BYTE nn BIT n] { OFF }
-----	--

where:

RECORD

sets the recording mode for a device.

ON

OFF

controls the intensive recording mode for the device that you specify. You can specify intensive recording for only one device at a time.

If you do not enter SET RECORD OFF, intensive recording will be automatically terminated after ten error records have been accumulated for the device, or when you activate intensive recording for a different device.

LIMIT nn

specifies the value at which a recording will be made. While intensive recording is active, an error record will be generated for the first and for every nn-th qualifying I/O error from the specified device.

BYTE nn

BIT n

the BYTE and BIT operands define the test condition in the sense bytes. You can test two bits using an AND or OR relationship. The range for BYTE is 0 to 23; the range for BIT is 0 to 7.

AND

OR

is the logical operation that you want to perform on two bits of sense data.

Usage Notes

1. For more information on error recording, see *VM/XA SP Real System Operation*.

Responses

None.

SET RESERVED

Privilege Class: A

Use **SET RESERVED** to establish the reserved page performance option for a specified virtual machine. The reserved page performance option will entitle that virtual machine to have a specified number of pages resident in real storage at all times.

SET	RESERVED {userid} {nn } * {OFF }
-----	--

where:

RESERVED

establishes the reserved page performance option.

userid

*

is the user identification of the virtual machine for which you want to establish the reserved page performance option. If you are specifying your own virtual machine, you can specify **userid** as an asterisk (*).

nn

OFF

is the number of pages that the user is entitled to have resident in real storage at all times. If the user's virtual machine has more than **nn** pages resident, normal system stealing procedures will be applied across all pages; reserved pages are not distinguished from other pages. If the user's virtual machine has **nn** pages or fewer resident in real storage, system stealing procedures do not apply to the user's pages unless there are no normal stealable pages in the system.

The maximum number of pages that you can reserve is limited by the size of the user's virtual machine.

Specify **OFF** to reset to zero the number of reserved pages for the specified user.

Usage Notes

1. Although you can specify the reserved page performance option for more than one user, you should use the **SET RESERVED** command sparingly, because system performance can diminish.
2. The **SET RESERVED** command does not provide any benefit when issued for a **V=R** or **V=F** user since such a user already has fixed storage.

Responses

None.

SET RUN

Privilege Class: G

Use SET RUN to control whether your virtual machine will begin to run after you enter a command on your display.

SET	RUN { ON } { OFF }
-----	-----------------------

where:

RUN { ON }
{ OFF }

controls whether your virtual machine will begin to run after you enter a command. If you specify ON, your virtual machine will begin to run following the entry. If you specify OFF, your virtual machine will remain in CP READ status after the entry. When you log on, RUN is set OFF for all users except the primary system operator of the VM/XA SP system.

Usage Notes

1. If you enter #CP preceding a command, your virtual machine will begin to run following the entry, regardless of whether RUN is set ON or OFF.
2. When RUN is set ON and your virtual machine is in a disabled wait state, "RUNNING" is displayed in the status area in the lower right-hand corner of your display screen. In this state, you receive messages without having to press the enter key.
3. The SET RUN ON command is rejected if you issued it when a TRACE set is active. Also, RUN is forced OFF when a TRACE set is made active.

Responses

None.

Migration Notes

VM/SP HPO:

1. VM/XA SP forces RUN OFF when you are tracing.
2. VM/XA SP never forces RUN OFF for a disabled virtual machine.
3. In VM/XA SP, a disabled virtual machine with RUN ON displays RUNNING on the screen.

SET SHARE

Privilege Class: A

Use the SET SHARE command to change a user's priority of access to system resources.

SET	SHARE userid { ABSolute nnn% RELative nnnnn INITial }
-----	---

where:

userid

is the userid of the virtual machine whose share you are changing.

ABSOLUTE nnn%

specifies that this user is to receive nnn% of the scheduled system resources which includes CPU, storage, and paging capacity. nnn ranges from 1-100.

RELATIVE nnnnn

specifies that this user is to receive a relative share of nnnnn. The amount of scheduled system resources available to relative share users is the total resources available less the amount allocated to ABSOLUTE share users. The portion that this user receives is nnnnn divided by the sum of the nnnnn's of all relative share users. For example, if one user's relative share is 100 and another user's relative share is 200, then the second user gets twice as much access to system resources as the first. If no share is specified in a user's directory entry, then the share defaults to a relative share of 100. nnnnn ranges from 1-10000.

INITIAL

specifies that this user's share of the system resources is to be reset to that specified by his or her directory entry. In the rare case when that userid no longer appears in the directory, the user's share of the system resources will be set to a relative share of 100.

Usage Notes

1. Be aware that a user's SHARE denotes priority access to the entire set of system resources, not just CPU. Currently, the following resources are scheduled according to SHARE:

- CPU
- Main storage
- Paging capacity.

In actuality, the bottlenecked resource, and a user's requirement for it, determine the access priority. For example, if storage is constrained, the user will receive access to main storage according to its SHARE. Likewise, if paging is a problem, the user will receive paging service according to its SHARE.

2. An ABSOLUTE SHARE user is receiving its correct service if it is able to perform at the inverse of its SHARE percentage of how it would have performed if running alone in the system. For example, if a user is to receive 50% (1/2) of the system, it should take no more than twice as long to complete a unit of work as it would have if running alone. If it were to get 25% (1/4) of the system, it should not take more than 4 times what it would running alone, and so forth. The gauge of RELATIVE SHARE delivery is slightly different. Two RELATIVE SHARE users with similar resource requirements can be compared.

SET SHARE

If the users are limited by the same bottlenecked resource, a **RELATIVE SHARE** user with twice the **SHARE** of another should run twice as fast.

Responses

```
USER userid: ABSOLUTE SHARE=nnn% [; m DEDICATED CPU[S]]
```

- or -

```
USER userid: RELATIVE SHARE=nnnnn [; m DEDICATED CPU[S]]
```

is the response you receive upon the successful completion of the command.

The string “; m DEDICATED CPU[S]” will only appear if the user has real processors dedicated to its virtual CPU or CPUs.

Migration Notes

VM/SP HPO: The SET SHARE command provides similar function as, but is not compatible to, the VM/SP “SET PRIORITY userid nn” and “SET FAVORED nnn” commands.

VM/XA SF: SET SHARE is a new command to VM/XA SP, but was taken from the “SET SRM SHARE userid ...” command in VM/XA SF. Note that the ABSCPU parameter has been removed from this and all other SHARE designations. This is because SHARE now applies to more resources than CPU. An installation which used the ABSCPU form should now use ABSOLUTE SHARE instead.

SET SHARED

Privilege Class: B

Use the SET SHARED command to allow a full pack mini-disk to either be shared or not shared by the users of multiple real and virtual systems.

SET	SHARED	{ ON } { OFF }	[FOR]	{ rdev } { rdev-rdev }
-----	--------	-------------------	-------	---------------------------

where:

SHARED

indicates that the device is shared or not shared between multiple real and virtual systems.

rdev

is the real device number of the device to be shared.

rdev-rdev

is the range of real device numbers of the devices to be shared.

Usage Notes

Warning: Use SET SHARED OFF with extreme caution and only after careful consideration.

1. You should contact the operator(s) of the system(s) that are sharing the device to assure that they mark a device as unusable in their systems' configurations, and physically disable their interfaces to that device, BEFORE you issue the SET SHARED OFF command.
2. You may use this command to tell CP whether to issue real reserve/release CCWs to the device associated with the full pack mini-disk.
3. When you SET SHARED OFF for a full pack mini-disk, that device will NOT be released by CP. If the device is currently reserved by CP for a user, it will remain reserved until that user issues a release to that device.
4. The SHARED status of system DASD may be determined by issuing the class B CP QUERY DASD RESERVE command.

Responses

```
SHARED SET { ON } FOR rdev
           { OFF }
```

is the response you receive upon the successful completion of the command.

SET SMSG

Privilege Class: G

Use the SET SMSG command to control whether your virtual machine will accept special messages.

SET	SMsg { ON OFF IUCV }
-----	----------------------------

where:

SMSG ON

OFF

controls whether your virtual machine is receiving special messages or not. If you specify OFF, the virtual machine cannot receive special messages. If you specify ON, the virtual machine can receive special messages. At logon time, SMSG is set to OFF.

IUCV

specifies that special messages are to be passed to the virtual machine via IUCV if a connection to the message system service exists. If no IUCV connection exists, the message will be handled as if SET SMSG ON had been issued.

Usage Notes

None.

Responses

None.

SET SRM

Privilege Class: A

Use the SET SRM command to change system parameters.

SET	SRM	$\left(\begin{array}{l} \text{IABias} \left[\begin{array}{l} \text{mmm}[\%] \quad \text{nnn} \\ \underline{90} \quad \underline{2} \\ \text{INITial} \end{array} \right] \\ \\ \text{DSPSlice} \quad \text{minslice} \\ \\ \text{STORBuf} \left\{ \begin{array}{l} \text{aaa}[\%] \quad \text{bbb}[\%] \quad \text{ccc}[\%] \\ \text{INITial} \end{array} \right\} \\ \\ \text{LDUBuf} \left\{ \begin{array}{l} \text{ddd}[\%] \quad \text{eee}[\%] \quad \text{fff}[\%] \\ \text{INITial} \end{array} \right\} \\ \\ \text{DSPBuf} \left\{ \begin{array}{l} \text{iiii} \quad \text{jjjj} \quad \text{kkkk} \\ \text{INITial} \end{array} \right\} \end{array} \right)$
-----	-----	--

where:

IABIAS $\left[\begin{array}{l} \text{mmm}[\%] \quad \text{nnn} \\ \underline{90} \quad \underline{2} \end{array} \right]$

specifies the new “interactive bias” to take effect. An interactive bias with a nonzero “mmm” causes users doing trivial transactions to receive better service than their scheduling share would otherwise entitle them to receive. Interactive bias causes a user’s share of the system to be momentarily larger than normal at the start of a new transaction. Then the user’s boosted share fades back to the normal share as the transaction progresses. If the transaction lasts even longer, then the user’s share sinks below what it would normally be until the extra service received earlier can be made up. The effect of this is to give good response time to trivial transactions that only require a small fraction of a second of processing to complete.

The value “mmm” is a percentage of the distance from the user’s normal position to the topmost user. The value “mmm” controls the “intensity” of the interactive bias. It may range from 0 to 100, with the IPL value and the default value being 90. The intensity determines where in the dispatch list the user is inserted for the first minor timeslice at the start of a new transaction. The insert point is somewhere between the normal (unbiased) insert point and the topmost user in the list. A setting of 100 gives a very strong interactive bias, while a setting of 0 eliminates the interactive bias.

The value “nnn” controls the “duration” of the interactive bias. It may range from 1 to 100; the initial value at IPL time and the default value is 2. The duration specifies the number of minor timeslices it takes for a user to fade back to his or her normal position in the dispatch list after the initial interactive boost. For example, a setting of 3 means the boost to the user lasts for 3 minor timeslices. The effect is strongest on the first timeslice and fades away until on the third timeslice the user receives only a small boost in the dispatch list. On the fourth timeslice the user receives no interactive boost at all and on the fifth timeslice the user begins to sag below their normal position in an effort to compensate for the larger-than-normal share the user received on the first few timeslices.

If “mmm” is specified without “nnn,” then the current interactive bias duration is not changed.

IABIAS INITIAL

sets the intensity (mmm) and duration (nnn) back to the initial values, 90 and 2.

DSPSLICE minslice

is the new size, in milliseconds, of the dispatching minor timeslice. This is the amount of CPU time a virtual machine may consume before re-examination by the scheduler. The value “minslice” must be in the range of 1 to 100.

STORBUF {aaa[%] bbb[%] ccc[%]}

partitions the scheduler’s view of main storage when choosing users to move from the eligible to dispatch lists. The result is a buffering mechanism based on the “transaction class” of a user, defined as:

E1, Q1 - users with short-running transactions

E2, Q2 - users with medium-running transactions

E3, Q3 - users with long-running transactions

where Ex is the class assigned to a user in the eligible list, and Qx is the designation after moving to the dispatch list. An E1 user in the eligible list becomes a Q1 user in the dispatch list, and so on. An additional “transaction class” is Q0, which are users who never wait in the eligible list to contend for resources in the dispatch list (users with the QUICKDSP attribute, for example).

The SET SRM STORBUF parameters partition the storage as follows:

aaa[%] - is the fraction of main storage the scheduler should view when considering adding an E1, E2, or E3 user to the dispatch list. If the total working sets of all users currently in the dispatch list plus the Ex user under consideration exceeds the aaa percentage of main storage, the user may not be added immediately and is delayed further.

bbb[%] - is the fraction of main storage the scheduler should view when considering adding an E2 or E3 user to the dispatch list.

ccc[%] - is the fraction of main storage the scheduler should view when considering adding an E3 user to the dispatch list.

In each case, the STORBUF values must be in the range 0-999, and $aaa \geq bbb \geq ccc$.

The result of this partitioning is that storage buffers are carved out for short-running work, which should include interactive work. Only Q0 and Q1 users may occupy the percentage of main storage represented by (aaa - bbb). Similarly, medium-running transactions can have a larger view of storage than long-running ones.

STORBUF INITIAL

sets the percentage parameters back to their initial values, 100%, 85%, and 75%.

LDUBUF {ddd[%] eee[%] fff[%]}

partitions the commitment of the system’s paging resources when choosing users to move from the eligible to dispatch lists. The result is a paging capacity buffering mechanism based on the “transaction class” of a user. (See the description of the STORBUF parameter on page 560 for a definition of transaction classes.)

During system initialization, the number of exposures to paging DASD is determined. Users expected to have high paging rates in the dispatch list are termed "loading users." They are identified by recent paging activity or with the knowledge that their working sets do not reside in main storage. One loading user is expected to keep one paging exposure busy. The LDUBUF (Loading User Buffer) percentages, then, determine the number of loading users allowed in the dispatch list by transaction class.

The SET SRM LDUBUF parameters partition the paging capacity as follows:

- ddd[%]** - is the fraction of paging exposures the scheduler should view when considering adding a loading E1, E2, or E3 user to the dispatch list. If the total number of loading users currently in the dispatch list plus the loading Ex user under consideration exceeds the ddd percentage of paging exposures, the user may not be added immediately and is delayed further.
- eee[%]** - is the fraction of paging exposures the scheduler should view when considering adding a loading E2 or E3 user to the dispatch list.
- fff[%]** - is the fraction of paging exposures the scheduler should view when considering adding a loading E3 user to the dispatch list.

In each case, the LDUBUF values must be in the range 0-999, and $ddd \geq eee \geq fff$.

LDUBUF INITIAL

sets the percentage parameters back to their initial values, 100%, 75%, and 60%.

DSPBUF {iiii jjjj kkkk}

affects the scheduler's policy for admitting users to the dispatch list in cases of users who are not prevented from entering by the more resource-specific barriers established by STORBUF and LDUBUF. DSPBUF stands for "dispatch buffer" and refers to the total number of users (by class) that are allowed in the dispatch list.

iiii refers to the number of openings in the dispatch list made available to users with short-running (E1) transactions. A subset, jjjj, of the iiii openings is also made available to users with medium-length transactions; the E1 and E2 users compete for these slots. Another subset, kkkk, of the jjjj openings is also made available to users with long-running transactions; the E1, E2, and E3 users all compete for these slots. QUICKDSP and other E0 users are not limited by these numbers and may enter the dispatch list without limit.

The valid range of values are 1 to 32767 where:

$iiii \geq jjjj \geq kkkk$

Example:

DSPBUF 35 30 18

Thirty-five openings are available to E1 users, but only five of them ($5 = 35 - 30$) are exclusively available to E1 and are guaranteed not to be already occupied by E2 and E3 users.

Thirty openings are available to E2 users, but only twelve of them ($12 = 30 - 18$) are guaranteed not to be already occupied by E3 users. None of the 30 openings are guaranteed against occupation by E1 users; however, E1 users generally get out of the way quickly.

Eighteen openings are available to E3 users, but none of the openings are guaranteed against occupation by E1 and E2 users. When E1 and E2 users do occupy these openings, you can expect them to get out of the way relatively quickly compared to the time scale of an E3 transaction.

DSPBUF {INITIAL}

sets the parameters back to their initial values which, for practical purposes, are infinity.

Usage Notes

1. In using SET SRM DSPSLICE, the initial value of the dispatching minor timeslice is determined by the model of the real machine that VM/SP is IPLed on. Making the timeslice smaller increases the system overhead, but it also gives better sorting of the dispatch list and more accurate share control by the scheduler. Making the timeslice larger decreases system overhead, but at the expense of the accuracy of the scheduler share controls. If the timeslice is made very large, response times and throughput rates may vary widely across users with similar shares.

Remember that when the dispatch slice gets changed, the IABIAS "duration" no longer represents the same amount of processing as before. If the size of the DSPSLICE is changed significantly, you may want to adjust the IABIAS "duration" to compensate so that it still represents roughly the same amount of processing as before.

2. The scheduler's use of the STORBUF percentages is best described by example. Assume the values are 100%, 85% and 75%. When the scheduler is determining whether or not to put an E1 user in the dispatch list, it first ensures that this user's storage requirement (working set size, or WSS) plus the working set sizes of all users already in the dispatch list is not greater than 100% of the available storage. When determining whether or not to put an E2 user in the dispatch list, the E2 user's WSS plus total WSS's in the dispatch list must not be greater than 85% of available memory, regardless of the mix of Q0, Q1, Q2 and Q3 users already in the dispatch list.

An installation may use the SET SRM STORBUF command to undercommit or overcommit storage. Note that Q0 users are not restricted by their storage requirement. By setting the the first percentage value (aaa) less than 100%, a storage buffer may be reserved for Q0 users. The SET RESERVED command, however, is a better method for a small number of Q0 users with predictable storage needs.

By increasing or decreasing the differences in the three percentage values, an installation can effect the response times and expansion factors for short-running (generally interactive), medium-running, and long-running work.

3. The SET SRM LDUBUF is very similar to SET SRM STORBUF, except that the resource buffered is the paging configuration capacity, rather than storage. An installation may use SET SRM LDUBUF to undercommit or overcommit the access to paging, or to change the emphasis on transaction length. In short, whenever SET SRM STORBUF is used, SET SRM LDUBUF should also be considered.

One difference to keep in mind, however, is that short-running transactions are likely to require more paging activity than longer-running ones. This probably is not true for the relative storage requirement. Short-running work is dormant more often for longer periods, thus have their working sets moved to paging DASD. For this reason, the difference between the first (ddd) and second (eee)

percentage values should perhaps be greater than that of SET SRM STORBUF (aaa and bbb).

Response 1

```
IABIAS : INTENSITY=nnn; DURATION=mmm
```

shows the results of a valid SET SRM IABIAS command.

Response 2

```
DISPATCHING MINOR TIMESLICE = xxx MS  
HCPxxxxI DSPSLICE AFFECTS THE IABIAS "DURATION". SEE "SET SRM IABIAS"
```

shows the results of a valid SET SRM DSPSLICE command.

Response 3

```
STORBUF: Q1=aaa% Q2=bbb% Q3=ccc%
```

shows the results of a valid SET SRM STORBUF command.

Response 4

```
LDUBUF : Q1=ddd% Q2=eee% Q3=fff%
```

shows the results of a valid SET SRM LDUBUF command.

Response 5

```
DSPBUF : Q1=iiii Q2=jjjj Q3=kkkk
```

shows the results of a valid SET SRM DSPBUF command.

Migration Notes

VM/XA SF: The SHARE option of the SET SRM command is no longer accepted. Use the SET SHARE command instead.

SET SVC76

Privilege Class: G

Use SET SVC76 to specify whether the control program (CP) or your virtual machine will handle the SVC76 instruction.

SET	SVC76 {CP} {VM}
-----	--------------------

where:

SVC76 { CP }
{ VM }

controls whether SVC76 instructions issued by your virtual machine are processed by CP for recording of virtual device errors. Specify CP if guest error records are to be recorded by CP when your virtual machine is running in supervisor state. Specify VM if you do not want errors recorded and if you want all guest SVC76 operations to be processed by your virtual machine. At logon, SVC76 is set to CP.

Usage Notes

1. If your virtual machine has SET SVC76 set to CP, and:
 - Issues an SVC76 from problem state, or
 - Uses an invalid SVC76 interface, or
 - Presents an unrecognizable EREP record, or
 - Presents an EREP record that cannot be translated,
 the SVC76 is reflected to the user.
2. Unless you are intentionally simulating hardware errors in your virtual machine (for example, in order to test error recovery procedures), you should let CP record all error records. When CP maintains the hardware error log, your installation has the most accurate information about equipment errors.

Responses

None.

SET TIMER

Privilege Class: G

Use SET TIMER to control the virtual interval timer. This command is valid only for System/370 mode virtual machines.

SET	TIMER { ON OFF REAL}
-----	-------------------------------

where:

TIMER {
ON
OFF
REAL}

controls the virtual interval timer. If you specify ON, the virtual timer is to be updated only when the virtual processor is running. If you specify OFF, the timer is not to be updated. If you specify REAL, the timer is updated during the virtual processor run time and also during virtual wait time. The initial setting when you log on is ON.

Usage Notes

None.

Responses

None.

SET VMCONIO

Privilege Class: G

Use the SET VMCONIO command to control the destination of virtual machine generated responses. This includes any linemode I/O done by the virtual machine to the virtual machine console. When you logon, VMCONIO is set OFF.

SET	VMConio {IUCV} {OFF }
-----	--------------------------

where:

IUCV

specifies that responses are to be passed to the virtual machine via IUCV if a connection to the message system service exists. If no IUCV connection exists, data will be handled as if SET VMCONIO OFF had been issued.

OFF

specifies that responses are to be displayed at the terminal.

Usage Notes

None.

Responses

None.

SET WNG

Privilege Class: G

Use SET WNG to control whether warning messages sent using the WARNING command are to be displayed. The initial setting for WARNING is ON.

SET	WNG { ON OFF IUCV }
-----	---------------------------

where:

ON

OFF

controls whether warning messages are displayed on your display screen. If you specify ON, all messages sent using the WARNING command will be displayed. If you specify OFF, no warning messages will be received. When you log on, WNG is set ON.

IUCV

specifies that warnings are to be passed to the virtual machine via IUCV if a connection to the message system service exists. If no IUCV connection exists, the message will be handled as if SET WNG ON had been issued.

Usage Notes

None.

Responses

None.

SHUTDOWN

Privilege Class: A

Use the SHUTDOWN command to systematically end all system functions and to checkpoint the system for an eventual warm start. This is the normal method that you will use to shut down your system.

SHUTDOWN	
----------	--

Usage Notes

1. Because the SHUTDOWN command stops all virtual machine operations, you should warn all users of an impending shutdown as early as possible so they may quiesce their systems in an orderly manner.

You can use the SET LOGMSG, MESSAGE, and WARNING commands to inform the affected users.

Response 1

SYSTEM SHUTDOWN STARTED

indicates that the command is accepted and termination processing is in progress. If a class A user other than the primary system operator issues the SHUTDOWN command, this response is sent to both the command issuer and the primary system operator.

Response 2

SYSTEM SHUTDOWN COMPLETE

indicates that the shutdown has been successfully completed.

Migration Notes

VM/SP HPO

1. VM/XA SP does not support the REIPL operand, which is supported in VM/SP HPO.
2. VM/XA SP provides an additional response to indicate that SHUTDOWN has started.

VM/XA SF

1. VM/XA SP allows a class A user other than the primary system operator to issue the SHUTDOWN command.
2. If a class A user other than the primary system operator issues the command, the response SYSTEM SHUTDOWN STARTED is sent to that user as well as to the primary system operator.

SLEEP

Privilege Class: Any

Use the SLEEP command to stop your virtual machine for a specified period of time. When the time period elapses, your virtual machine will automatically resume execution.

Sleep	[nn [SEC MIN HRS] [ATTN]]
-------	-------------------------------------

where:

nn [SEC
MIN
HRS]

indicates the number of seconds, minutes, or hours, measured by the time-of-day clock, that your virtual machine is to remain stopped. The value nn can be any decimal number from 00 through 99. If you specify no time unit, CP assumes that nn indicates minutes.

ATTN

indicates that if the SLEEP command with a time interval was issued via a DIAGNOSE Code X'08' instruction, the return code should indicate whether the SLEEP command ended because the time interval expired (RC=0), the ENTER key pressed (RC=2) or because an attention was signalled (RC=1457).

Usage Notes

1. During the dormant period, the virtual machine does not run but connection time is still counted.
2. Messages continue to be displayed even though your virtual machine is in the dormant state.
3. You can start your virtual machine at any time by pressing the ENTER key. If you specify a sleep interval in the command line, your virtual machine awakens automatically when the specified interval has elapsed.
4. If you do not specify an interval, your virtual machine remains dormant until you press the ENTER key.
5. If you issue the SLEEP command from a CP environment, you will return to the CP environment when the time interval has elapsed, or when you press the ENTER key.
6. If you issue the SLEEP command while in virtual machine environment, the end of the time interval or pressing ENTER returns your display to virtual machine mode without entering the CP environment.
7. The SLEEP command, with the time interval, is a convenient way to delay or schedule the execution of certain jobs that could be run more efficiently at a later time (for example, second shift).
8. If the SLEEP command with a time interval was entered using a DIAGNOSE instruction, the return code indicates whether the interval has expired (RC=0); whether the ENTER key was pressed (RC=2), or an ATTENTION was signaled (RC=1457).

SLEEP

Responses

None.

Migration Notes

VM/SP HPO

1. VM/XA SP does not terminate SLEEP when you reconnect.
2. VM/XA SP supports the ATTN operand which allows you to be notified if attention terminates the SLEEP.

VM/XA SF: Unless you specify ATTN, VM/XA SP always provides return code 0 from a valid SLEEP command even if attention terminates the SLEEP.

SMSG

Privilege Class: G

Use the SMSG command to send a special message to a virtual machine programmed to accept and process the message.

SMsg	userid msgtext
------	----------------

where:

userid

is the identification of the virtual machine to whom the special message is to be sent. If you specify userid as '*', it indicates that the message is to be sent to yourself.

msgtext

is the text of the message that you are sending.

Usage Notes

1. Special messages do not appear on the receiving user's console because the messages are expected to be instructions or commands that are processed by the receiving virtual machine.
2. Virtual machines can only receive special messages under the following conditions:
 - The receiving virtual machine has issued the SET SMSG ON command, or set the VMCP SMSG flag at AUTHORIZE time
 - The user wishing to receive special messages has issued an AUTHORIZE with the Virtual Machine Communication Facility (VMCF)
 - You can also receive special messages via IUCV by issuing the SET SMSG IUCV command.

Responses

None.

SPACE**Privilege Class: D**

Use the SPACE command to force the current output file of a real printer to be single spaced, regardless of the carriage control commands contained in the actual file.

SPAcE	rdev
-------	------

where:

rdev

is the real device number of the printer whose output file is to be single spaced.

Usage Notes

None.

Responses

```
PRT rdev SPACING   userid   FILE spoolid RECS { nnnn } COPY [*]nnn SEQ sss
                   { nnnK }
                   { nnnM }
```

where:

rdev

is the real device number of the printer.

userid

is the identification of the user whose file was active and single spaced.

FILE spoolid

spoolid is a file currently being processed.

RECS { nnnn }
 { nnnK }
 { nnnM }

is the number of logical records in the spool file. If the number is greater than 9999, then the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

is the remaining number of copies of the file to be created. The asterisk (*) indicates that the 3800 printer is to copy each page nnn times before going on to the next page.

SEQ sss

is the sequence number for the file on the printer.

Migration Notes

VM/SP HPO: VM/XA SP provides a response concerning the file being spaced as opposed to no response as in VM/SP HPO.

SPOOL

Use the SPOOL command described in this section to set the spooling control options for one or more of your virtual spooling devices. The options that you select modify operational functions associated with your virtual reader, printer, punch, or console and control the disposition of files after they have been processed.

General Usage Notes for SPOOL

1. A spool file is created when the first record for a printer or punch is processed by the virtual unit record simulation routines in CP. CP associates this spool file with the virtual unit record device and assigns it a number called the spoolid (spool file identification). You can use the spoolid to identify the file that you have produced, and to manipulate the file using the VM/XA System Product CP spooling commands, such as ORDER, CHANGE, and PURGE.

When you print or punch a file, CP displays its spoolid. You can use the spoolid to find out various kinds of information about the file using the CP QUERY command.

2. It is important that you remember that the SPOOL command controls both operational characteristics of your virtual spooling devices and the disposition of files after they have been processed.

For example, you use the SPOOL command to select which files you want to process on your virtual readers. The SPOOL READER command does not change any of the characteristics associated with the files in your reader queue, but simply indicates whether a file will be purged or retained after it has been read, and if it is to be retained, whether the file remains eligible for immediate reprocessing.

The files in your reader queue are controlled using a combination of file characteristics and the options in effect for the virtual reader on which they are to be processed. Input files do not derive their characteristics from the reader, except for HOLD status, which is set as a function of the KEEP option. Figure 1 on page 575 summarizes the disposition of a reader file once it has been processed, depending on the HOLD/KEEP settings specified for the reader, and the KEEP setting of the file.

For output devices (virtual printers, punches, and consoles) use the SPOOL command to define the characteristics of the spool files that you are creating. You can use the SPOOL command to indicate whether files will be held from further processing, and whether they will be retained after they have been processed. This information is associated with the spool file itself, regardless of the characteristics of the device on which the file is to be processed. In addition, you can use the SPOOL command to direct the output files that you are creating to unit record devices (readers, printers, and punches) associated with other virtual machines.

File Status	Device Options	Disposition
NOKEEP	NOHOLD/NOKEEP	The file is purged.
NOKEEP	NOHOLD/KEEP	The file is retained in user HOLD status. KEEP is not set on the file.
NOKEEP	HOLD/NOKEEP	The file is retained and remains eligible for processing on your virtual reader.
NOKEEP	HOLD/KEEP	The file is retained in user HOLD status. KEEP is not set on the file.
KEEP	Any setting	The file is retained in user HOLD status. KEEP status remains on the file.

Figure 1. Disposition of Reader Files Depending Upon HOLD/KEEP Settings

All of your output files derive their characteristics from the virtual device on which they were created. If the output device is spooled with the KEEP and NOHOLD options, the file will be printed or punched, and then be placed in HOLD status, retaining its KEEP characteristics. If you change the file status to NOHOLD, the file is processed at the real device one more time and once again is placed in HOLD status. If you do not want to retain the file, you must change both its NOHOLD status and NOKEEP characteristic, or purge it using the CP PURGE command.

3. Unless you reset these values, the following options are the default values for your spool files:

Device	Default Options
Reader	NOHOLD NOKEEP NOCONT EOF CLASS A ¹
Punch	NOHOLD NOKEEP NOCONT CLASS A ¹ OFF COPY 001 NOMSG NONAME NOEOF
Printer	NOHOLD NOKEEP NOCONT CLASS A ¹ OFF COPY 001 NOMSG NONAME NOEOF
Console	NOHOLD NOKEEP NOCONT CLASS T ¹ OFF COPY 001 TERM TO * PRINTER NOMSG STOP NONAME NOEOF

¹These are default classes only if a specific class is not part of your entry in the system directory.

4. You can transfer a spool file to another VM/XA SP user by issuing the CP SPOOL, CHANGE, CLOSE, or TRANSFER commands. Use the CHANGE and TRANSFER commands to transfer already existing files. Use the SPOOL or CLOSE commands to transfer files which you are in the process of creating, or files which you will create in the future. For example, if you want to transfer the printer or punch files that you are creating to another user's virtual reader, issue the SPOOL command with the TO option:

```
SPOOL vdev TO userid RDR
```

where vdev is the virtual device number or the virtual device "type" of your virtual machine output device and userid is the user identification of the other user's virtual machine. Designating either a reader, printer, or punch is

SPOOL

optional; the default is to the reader. The command defines the user's queue to which you are directing the file.

If you want to print or punch a file for someone else, you can also issue the SPOOL command with the FOR option:

```
SP00L vdev FOR userid
```

where vdev is the virtual device number or the virtual device type of your virtual machine output device and userid is the user identification of the other user's virtual machine.

5. When a file changes ownership, as when it is changed to or from another user, the spool file ID number also changes. The new number is obtained from the pool of available ids of the recipient of the file.

Thus, if user SAM changes his file #3 to user HARRY, he would get the response:

```
RDR 0003 SENT TO HARRY RDR AS 0018
```

Harry would now have a file #18.

SPOOL READER/PRINTER/PUNCH/CONSOLE

Privilege Class: G

Use SPOOL READER/PRINTER/PUNCH/CONSOLE to modify the spooling control options in effect for your virtual machine.

SPool	{Reader } { option A } ¹ {Printer } { [option A] [option B] } ¹ {PUnch } {CONsole} { [option A] [option B] [option C] } ¹ {vdev }
-------	--

¹You must specify at least one option of the indicated options. These are described in "SPOOLING: Options" on page 579.

where:

READER
RDR
PRINTER
PRT
PUNCH
PCH
CONSOLE

modifies the options for all of your virtual readers, printers, punches, or console spooling devices, including the STARTing or STOPping of spooling operations.

vdev

specifies the virtual device number of the virtual spooling device whose options you want to modify.

options...

see "SPOOLING: Options" on page 579.

Migration Notes

VM/SP HPO

1. If a reader file is in KEEP status, VM/XA SP changes the file to HOLD after it has been read.
2. In VM/XA SP a reader file will not be put into user HOLD if the virtual reader is spooled CONT and HOLD.
3. VM/XA SP truncates a name identification for MVS data set names to 16 characters (if longer than 16).
4. VM/XA SP allows you to specify additional operands: KEEP/NOKEEP and MSG/NOMSG.
5. VM/XA SP does not support the DEST option.
6. VM/XA SP sets DIST to what is in the directory for DIST *.
7. VM/XA SP has a minimum truncation of NOT for NOTERM.

SPOOL READER/PRINTER/PUNCH/CONSOLE

8. VM/XA SP accepts SPOOL PRT CLOSE PURGE and then closes and purges the file.
9. VM/XA SP always gives responses whenever a file is closed.
10. If the DIST option is specified for a device, VM/XA SP sets the distribution code for all files created on the device to what was specified.
11. The VM/XA SP response includes the number of records and KEEP setting.

SPOOLING: Options

option A:	[HOld] [KEep] [CONt] [EOF] [Class c] [NOHold] [NOKeep] [NOCont] [NOEof]
option B:	<p>[NOMsg] [COpy [*]nnn] [Dist {distcode}] [MSG] { OFF * }</p> <p>[To] {userid} [Reader] [CLOSE] * {SYSTEM} Printer [PURge] PUnch</p> <p>For {userid} * {SYSTEM}</p> <p>OFF</p> <p>[FORM {form }] {OFF }</p> <p>[Flash {name [ALL] name [nnn] }] {OFF }</p> <p>[MOfify {name [n] }] {OFF }</p> <p>[FCB {name }] { 6 8 10 12 OFF }</p> <p>[CHars {name0 [name1 [name2 [name3]]] }] { OFF }</p> <p>[CHars name0 [CHars name1 [CHars name2 [CHars name3]]]</p> <p>[NAmE {fname [ftype]}] {dsname } NOName</p>
option C:	[STArt] [NOTerm] [STOp] [TErm]

where:

HOLD

NOHOLD

indicates whether you want to HOLD files.

If you specify HOLD for an output device, all the files created by the device are placed in USER HOLD status. When a spool file is in USER HOLD status, it is not processed at its destination device (the real printer, punch, or virtual reader) until you remove the HOLD status by issuing the CP CHANGE command with the NOHOLD option.

If you specify HOLD for an input device, files are retained after they have been read. Unless you specify the KEEP operand for the file or for the device, the file remains in the reader queue and is the next file eligible for processing. This is in contrast to the HOLD status on a spool file, which makes the file ineligible for processing.

Specify NOHOLD to allow files to be purged after processing.

KEEP

NOKEEP

indicates whether you want to retain a spool file after it has been processed.

Specify KEEP if you want to retain files after they have been processed.

For a virtual printer or punch directed to a real device, after the file has been printed or punched, it is placed in USER HOLD status, while retaining its KEEP status.

For a reader, KEEP also places your spool file in user HOLD status after it has been read, but KEEP status is not set on the file. Note that the reader DEVICE hold option does not affect FILE hold status.

Specify NOKEEP to reset the KEEP status so that files are not retained in the system once processing has been completed.

CONT

NOCONT

controls the continuous spooling option.

For output devices, CONT causes all CLOSE requests to be ignored unless specified with EOF override option. NOCONT resets the option.

For readers, CONT specifies that reading will continue without end-of-file indications at the end of each virtual spool file, until the virtual device is empty. NOCONT specifies that end-of-file is signaled after each spool file is read, and then the reader resets at the next logical file.

EOF

NOEOF

For a printer or punch, EOF causes the device to be automatically closed when 50,000 records are placed into the active spool file. Punches are closed immediately at 50,000 records. If page controls are found in a print file within the next 100 records, the file is closed at the end of the page. If no page controls are found by then, it is presumed there are none, and the file is closed after 50,100 records.

For a reader, the EOF/NOEOF option specifies the reflection given when end-of-file is reached (as defined by the CONT/NOCONT option). EOF specifies that a unit exception is reflected on the first read past the end of the file. Further reads result in a normal read of the next file, if any, or with a unit

check if the device is empty. NOEOF specifies that unit check is reflected on the first read past end-of-file, or any subsequent read to an empty reader.

CLASS c

specifies the spool class of the device. c is a 1-character alphanumeric field whose values can be A through Z, 0 through 9, or * (asterisk). The CLASS option determines which destination devices can process the spool file. If you print or punch a file from one of your virtual devices, the file is assigned the class that is associated with the device. The file is then printed or punched on a real device which also has this class associated with it. The operator sets this class.

The class associated with a virtual reader determines which files it recognizes for processing. If you specify a specific class, the reader processes only files of the same class. If you specify *, the reader processes all files regardless of class.

MSG**NOMSG**

indicates whether you want to be informed about file printing and punching. If you specify MSG, you are informed of file printing and punching if the file is printed or punched on the same system as your user ID. The message is sent to you when the file begins printing or punching on a real device. This message is not sent if you have issued CP SET IMSG OFF. Specify NOMSG if you do not want messages sent to you when the file begins printing or punching on a real device.

COPY [*]nnn

indicates the number of copies that are to be printed or punched when the file is spooled to the real unit record device. Note that regardless of what number you specify for COPY, only one spool file is actually created. However, VM/XA SP continues to associate the COPY number with the spool file until the spool file is processed by a real printer or punch. At that time, VM/XA SP produces the specified number of real printed or punched copies. If the real printer is a 3800 device, an asterisk (*) indicates that the printer is to print each page nnn times before going on to the next page.

[TO] { userid
*
SYSTEM }

sends the output of the virtual device to the virtual reader, virtual printer, or virtual punch of the specified userid. The distribution code on the spool files becomes the distribution code specified for userid in the system directory.

When you specify TO, you retain control of the spool file. For example, you can use the TRANSFER command to retrieve the file from the specified userid.

TO SYSTEM is equivalent to specifying OFF and resets the transferred spool option.

READER**PRINTER****PUNCH**

indicates the spool file queue to which your file is directed. If you do not specify a queue, the file is transferred to the other user's reader queue.

FOR { **userid**
*
SYSTEM }

except for console output, transfers the spool file from the virtual device to the corresponding virtual device of the specified userid. VM/SP transfers console output to the receiver's virtual printer. The distribution code on the spool files becomes the distribution code specified for userid in the system directory.

You can specify FOR * to change the spool file ownership and destination code to those of your own virtual machine. Specifying FOR * is equivalent to specifying the OFF operand for the device.

FOR SYSTEM is equivalent to specifying OFF and resets the transferred spool option.

If you specify FOR userid, you cannot specify either the TO or the OFF operand.

OFF

resets the TO or FOR option from the previous spooling command.

FORM form

specifies the form name for the resulting spool file(s). "form" is 1 to 8 characters.

FORM OFF

resets the form to the system default as specified by the installation.

FLASH name [*ALL*]
[**nnn**]

signifies that a form overlay contained in the 3800 printer is to be superimposed (or flashed) onto the specified number of copies of the output. The name, to be 1 to 4 characters in length, identifies the forms overlay to be used in the projection. The number, nnn, must be in the range from 0 to 255. The number specifies that the first nnn copies of output are to be flashed.

For example, if you specify COPY 4 and FLASH name 2, the first two copies of your file are flashed, while the last two copies remain unflashed. If "ALL" is specified, the flash count always matches the copy count. "ALL" is the default.

FLASH OFF

resets the flash parameters. Blanks are inserted into the flash name field of the specified spool file or files, and the count is set to 0.

MODIFY name [n]

assigns a copy modification module to be used when printing the spool file or files on the 3800. This function makes it possible to alter text by either preventing printing of certain information or adding labels to output. The name must be 1 to 4 characters.

"n" selects one of the character sets specified by the CHARS operand to be used for the copy modification text. The values 0, 1, 2, and 3 correspond to the order in which the table names have been specified. If no "n" is specified, the first character set ("0") is assumed.

MODIFY OFF

resets the modify parameters. Blanks are inserted into the modify name field of the spool file or files and, when printed, no modification will occur. The modify number ("n") is set to 0.

FCB { **name** }
 { 6 }
 { 8 }
 { 10 }
 { 12 }

specifies the forms control buffer to be used when printing the spool file or files on the 3800 printer. name is the 1- to 4-character name of an FCB image, or the number 6, 8, 10, or 12. This number tells the 3800 to print the entire spool file at 6, 8, 10, or 12 lines per inch, regardless of the size of the paper that is currently loaded. An FCB specified at 10 will print only on a 3800 Model 3 printer.

FCB OFF

resets the FCB parameter. Blanks are inserted into the FCB name field of the spool file or files and, when printed, a default FCB will control the vertical spacing of the output text.

CHARS name0 [name1 [name2 [name3]]]

CHARS name0 [CHARS name1 [CHARS name2 [CHARS name3]]]

specifies the name of the character set or sets used when printing the spool file or files on the 3800 printer. Each name must be from 1- to 4-characters with a maximum of four names specified. The multiple use of "CHARS" need only be used if "nameN" would conflict with an option name (for example, "FORM") or with a userid. If more than one name is specified, along with modification, the order in which the names appear determines which table will be used for copy modification text.

CHARS OFF

resets the "CHARS" parameters. Blanks are inserted into the name fields of character sets to be used for output generation. The resulting spool file or files, therefore, will use a default character set when actually printed on a 3800 printer.

DIST { **distcode** }
 { **OFF** }
 { * }

specifies the distribution code to be assigned to spool files created on this virtual device. 'distcode' is 1 to 8 characters. You can override this distcode using the DIST option on the CP CLOSE or CP CHANGE commands.

"DIST OFF" and "DIST *" reset the distribution code of the spool file(s) to the default specified in your entry in the system directory.

NAME { **fname [ftype]** }
 { **dsname** }

specifies the file name and optional filetype, or the data set name that will be given to each spool file the virtual device creates. The fname and ftype must be 8 characters or less. The data set name may be up to 24 characters; however, if the name has more than 16 characters, it is truncated to 16 characters.

NONAME

resets the NAME option. Blanks are inserted into the name field for spool files the virtual device creates.

CLOSE

closes the spool file on the specified device regardless of the CONT setting for the device. CLOSE is equivalent to issuing CLOSE with the EOF option. If PURGE is specified, CLOSE is not necessary.

SPOOLING: Options

PURGE

closes and purges the spool file on the specified device regardless of the CONT setting for the device. If PURGE is specified, the CLOSE option is not necessary. PURGE is equivalent to issuing CLOSE with the PURGE option.

START

commences placing input and output from your display station into a spool file. Full-screen data output operations are not placed in the spool file. After the console is closed, the file becomes a printer spool file. START applies only to consoles.

STOP

terminates the spooling of display station input and output. STOP does not CLOSE the console spool file. STOP applies only to consoles.

TERM

displays the display station input and output on your display screen in addition to placing it in a spool file. The TERM operand has no effect unless the START function is also in effect. TERM applies only to consoles.

NOTERM

suppresses the display of input and output on your display screen while placing it in a spool file. Full screen output is not suppressed and is not placed in the spool file. The display of console input and output is not suppressed if:

- It is the result of CP commands entered from CP mode
- The output line immediately precedes a virtual machine read to the console.

The NOTERM operand has no effect unless the START function is in effect. NOTERM applies only to consoles.

Usage Notes

Refer to "General Usage Notes for SPOOL" on page 574.

When specifying "[TO] userid," if you omit the keyword TO, the userid cannot be "T" or "TO." If you are transferring your output to your own virtual machine, you may specify userid as an asterisk (*). If you specify TO userid, you cannot specify either the OFF or FOR operand.

Responses

If the SPOOL command is used to close an output device, the message:

```
{RDR} FILE spd1 SENT {TO } userid {RDR} AS spd2 RECS {nnn} COPY [*]nnn c {HOLD } {KEEP }  
{PRT} {FROM} {PRT} {nnnK}  
{PUN} {PUN} {nnnM}  
{CON}
```

is the response you receive unless you have issued the CP SET IMSG OFF command.

where:

RDR
PRT
PUN
CON

indicates the spool file queue. If you are transferring a file to another user's virtual machine, this field indicates the queue from which the file originated. If you are receiving a file from another user, this field indicates the file queue in which the file has been added.

spd1

is the spool identification number of the file at the time the command is issued.

TO userid

indicates to whom the file has been sent.

FROM userid

indicates from whom the file came.

RDR
PRT
PUN

indicates the spool file queue. If you are transferring a file to another user's virtual machine, this field indicates the queue in which the file has been added. If you are receiving a file from another user, this field indicates the file queue in which the file originated.

spd2

is the spool identification number of the file after the command is complete.

RECS $\left\{ \begin{array}{l} \text{nnnn} \\ \text{nnnK} \\ \text{nnnM} \end{array} \right\}$

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

specifies the number of copies of the file you want printed or punched when the file is produced.

The asterisk (*) indicates that the 3800 printer is to copy each page nnn times before going on to the next page.

If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (*) is ignored and duplication occurs normally.

c

is the spool file class assigned to the file.

HOLD
NOHOLD

is the user HOLD status associated with the file.

KEEP
NOKEEP

is the KEEP status associated with the file.

SPTAPE

The format box below shows all the operands available with the SPTAPE command. Complete descriptions of the operands appear on the following pages in the same sequence as the operands appear here.

SPTape	<p>CANCEL rdev</p> <p>DUMP rdev [userid] { { Printer } { spoolid1 [spoolid2] } } { PUnch } { Class { abcdefgh } } { Reader } { FORM { form } } { } { ALL } }</p> <p>{ NSS } { spoolid1 [spoolid2] } { IMG } { Class { abcdefgh } } { UCR } { } { NLS } { } { TRFiles } { ALL }</p> <p>{ LEAVE } { REWind } { RUN } }</p> <p>{ MODE } { 800 } { 1600 } { 6250 } { 38K }</p> <p>[[SYSHold] [USERHold] [NOHold]] [PURge]</p> <p>LOAD rdev [userid] { { Printer } { spoolid1 [spoolid2] } } { PUnch } { Class { abcdefgh } } { Reader } { FORM { form } } { } { ALL } }</p> <p>{ NSS } { spoolid1 [spoolid2] } { IMG } { Class { abcdefgh } } { UCR } { } { NLS } { } { TRFiles } { ALL }</p> <p>{ LEAVE } [[SYSHold] [USERHold] [NOHold]] { REWind } { RUN }</p> <p>SCAN rdev [userid] { LEAVE } { REWind } { RUN }</p> <p>STOP rdev</p>
--------	--

Notes:

1. Form and class may be specified on the same command line.
2. If you have only class E command privileges, you cannot process reader, printer, or punch files.

The SPTAPE command supports the migrating of spool files between different VM systems. However, this migration is supported between the following systems only:

VM/SP	Release 5
VM/HPO	Release 5
VM/HPO	Release 4.2
VM/HPO	Release 3.8
VM/HPO	Release 3.6
VM/XA SF	Release 1
VM/XA SF	Release 2
VM/XA SP	

Any references to VM/SP in the following SPTAPE pages refer to these releases only. For more information, refer to When to Use the SPTAPE Command on page 589.

This function allows spool files and system data files to be saved on tape. Also, it allows the files to be loaded from the tape to the spool file system.

General Usage Notes for the SPTAPE Command

1. You may issue as many SPTAPE commands as you wish and load from or dump to multiple tape drives. However, if loading and dumping requests are active simultaneously, caution should be exercised since the two functions may overlap.

For example, if a LOAD request has loaded a file whose attributes match the specifications on a “dump purge” request, the file is dumped and purged from the system.
2. Tape handling options are provided on the LOAD, SCAN, and DUMP options and should be used for tape positioning for subsequent commands. The STOP option positions the tape. This allows the user to continue with a new command at the current position. The CANCEL option assumes that the user is done and unloads the tape. If you do not wish to continue, do not mount a new tape and issue the SPTAPE CANCEL or SPTAPE STOP command.
3. When files being dumped do not fit on a single tape reel, the current tape is rewound and unloaded, a command response is given that asks for a new tape to be mounted and that identifies the last complete file, if any, dumped. After the next tape is mounted and readied, dumping of the remaining files continues. If a file is only partially dumped when the end of a tape is reached, the entire file is dumped onto the new tape.
4. If the message “TAPE WAS NOT REWOUND” appears after you issue the SPTAPE SCAN REWIND command, you must issue another REWIND request.
5. While the SPTAPE command is processing a file in the spooling subsystem, that file can be queried, but it cannot be manipulated or changed in any way.
6. Both 3420 9-track tape drives and 3480 tape drives are supported.

7. A file loaded into the system from a tape retains the same characteristics as the original file, but is assigned a new spoolid in the spooling system. A message is issued that correlates the old and new spoolids.
8. Open files cannot be dumped.
9. Only the user who issues an SPTAPE DUMP, SCAN, or LOAD command may STOP or CANCEL that command.
10. If the user does not wish to receive SPTAPE responses, the CP command SET IMSG OFF may be used to suppress them. (This does NOT suppress the request to mount a new tape when the current one is full.)
11. On a 3420, if the mode selected is not consistent with the tape attributes supplied at system generation time or if the tape drive was generated as a 7-track drive, a message is displayed asking the SPTAPE user if SPTAPE processing should continue. If the user answers "yes," SPTAPE attempts to write at that mode. If the user answers "no," SPTAPE processing halts. For a 3480, if the mode is not 38K, an error message is issued.
12. If a spool file originated in a VM/SP system migrates to VM/XA SP, it loses those file attributes which do not have a VM/SP equivalent. Therefore, if the file is migrated back to the VM/SP system, it will not be identical to the original file.

For example, if a VM/SP 3211-type file has imbedded UNFOLD (X'23'), FOLD (X'43'), or LOAD FCB (X'63') CCWs and is loaded into a VM/SP system, those CCWs are removed because they are not supported in VM/XA SP.

13. When transporting files from VM/XA SP to VM/SP, do not dump system type files on tape. These files are: CP ABEND DUMPs (usually class D), VMDUMPs (usually class V), NSS, IMG, UCR, NLS, and TRF files. VM/SP SPTAPE detects an error and ends processing if it encounters one of these files on tape.
14. If files are being transported to the VM/XA system and the file owner's userid does not exist on the VM/XA system, the file is assigned to the SPTAPE command issuer's userid.
15. When loading NSS, IMG, UCR, NLS, or TRF files, the user may be loading files with the same filename and filetype as other NSS, IMG, UCR, NLS, or TRF files currently active on the system. If this should happen, check these files to determine which should be kept. The unnecessary files should be purged.
16. SPTAPE does not dump class W system trace files.
17. When files are being loaded back into a VM/XA SP system, if the owner of the file runs out of spoolids, the file is assigned to the SPTAPE command issuer and a message is sent to both users.
18. When a file is loaded from tape, the spool file id number changes. The new number is obtained from the pool of available ids of the recipient of the file. Thus, if user SAM changes his file #3 to user HARRY, he would get the response:

```
RDR 0003 SENT TO HARRY RDR AS 0018 RECS 1000 COPY 001 A NOHOLD NOKEEP
```

Harry would now have a file #18.

Action Taken on Tape Overflow

If the output tape fills before all files specified by the command are dumped, you can either continue or end the command:

- Mount and ready a new tape on the same tape drive. The command will continue.
- Issue SPTAPE rdev CANCEL or SPTAPE rdev STOP to end the command.
- If you take no action, the command remains dormant and the tape is not available for other users.

Action Taken on I/O Error Conditions

When an I/O error condition occurs during a DUMP operation (such as an error reading the file from disk or writing the file to tape), the following happens:

- The system issues a message.
- The tape is rewound to the last complete file.
- A second tapemark is written to indicate end-of-tape.
- The tape is positioned as specified by the user.
- The command is halted.

If the PURGE option was specified, all files successfully dumped to tape will be purged. All other files remain in the spool file system.

When loading or scanning, an I/O error can occur when reading a file from tape, or writing a file to DASD. Also, spool space can become full or there may be no more spool file id's available when loading files back into the system.

If either an I/O error occurs or spool space does become full, the following happens:

- The system issues a message.
- The tape is rewound to the last tapemark, then positioned as specified by the user.
- The command is halted.

Files that have been successfully read remain in the spooling system, but any partial file is purged.

When to Use the SPTAPE Command

When the spool file operator determines that the spooling system is overloaded, the operator can use the SPTAPE command to temporarily store files on tape.

The SPTAPE command enables the spooling operator to store to, and retrieve from tape, those spool and system data files that he or she wants to schedule later on VM/XA SP. The files can be read back into the VM/XA SP system selectively so as not to overload the spooling system. An option is provided to scan the tape for priority files.

Transferring spool files from the VM/XA SP spool file system to the VM/SP spool file system, or back again, can be accomplished using the SPTAPE command. Use the SPTAPE DUMP option to write the files of one system to tape. The files may then be loaded into the other system's spool file queues by using the SPTAPE LOAD option.

SPTAPE

Note: System data files and other system-type files (for example, CP ABEND, VMDUMP, CPTRAP, ACNT, and MONITOR) files are not transportable between systems if both are not either VM/SP or VM/XA systems.

Finally, SPTAPE can be used to back up named saved system (NSS) files, image file libraries (IMG), saved segments, user class override files (UCR), system trace files (TRF), and message repository files (NLS).

Response 1

When spool or system data files are being loaded, dumped, or scanned on tape, the user is informed of each file processed through the following response:

$\left. \begin{array}{l} \text{DUMPING} \\ \text{LOADING} \\ \text{SCANNING} \end{array} \right\}$	rdev	$\left. \begin{array}{l} \text{PRT} \\ \text{PUN} \\ \text{RDR} \\ \text{NSS} \\ \text{IMG} \\ \text{UCR} \\ \text{NLS} \\ \text{TRF} \end{array} \right\}$	userid spoolid1 class form stat [NOW spoolid2]
--	------	---	--

where:

DUMPING
LOADING
SCANNING

indicate which SPTAPE functions are active.

rdev

is the real device number of the tape drive on which the SPTAPE operation (dumping, loading, or scanning) is taking place.

PRT
PUN
RDR
NSS
IMG
UCR
NLS
TRF

indicate the types of files being processed.

userid

is the userid of the file owner.

spoolid1

is the spoolid first given to the spool file.

class

is the spool class of the file.

form

is the operator form number of the file. The form field is blank for system data files.

SPTAPE

rdev

is the real device number of the tape drive on which the SPTAPE function is taking place.

COMPLETE

STOPPED

CANCELLED

is the current status of the SPTAPE function.

Response 5

When a file is dumped with the PURGE option, or when a file is loaded, the file owner is sent the following response:

```
{RDR} FILE spd1 SENT {TO } userid {RDR} AS spd2 RECS {nnn} COPY [*]nnn c {HOLD } {KEEP }  
{PRT} {FROM} {PRT} {nnnK}  
{PUN} {PUN} {nnnM}
```

where:

RDR

PRT

PUN

is the virtual device on which the file is being created or read.

spd1

is the spool identification of the file when it is associated with the command user.

TO userid

indicates to whom the file has been sent.

FROM userid

indicates from whom the file came.

RDR

PRT

PUN

indicates the spool file queue that now contains the file.

spd2

is the spool identification of the file when it is associated with the "userid" in the response.

RECS nnnn

nnnK

nnnM

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

specifies the number of file copies you want printed or punched when the file is produced.

When the optional asterisk (*) is specified, the 3800 printer prints each page from the spooled output files nnn times before going on to the next page.

c
is the spool file class assigned to the file.

HOLD
NOHOLD

is the user HOLD status associated with the file.

KEEP
NOKEEP

is the KEEP status associated with the file.

This response is suppressed if the virtual machine user has issued the SET IMSG OFF command.

Response 6

If the end-of-reel is reached before all the selected files have been dumped, the following response is sent:

```
END OF VOLUME ON TAPE rdev, MOUNT NEXT TAPE
{LAST FILE DUMPED userid spoolid}
{NO FILES DUMPED }
```

where:

rdev

is the real device number of the tape drive on which the SPTAPE dumping operation is taking place.

userid

is the userid of the last file that was dumped.

spoolid

identifies the last spoolid, if any, that was dumped.

The SPTAPE user should mount and ready another tape. If the user does not wish to continue dumping to another tape, the user should issue the SPTAPE CANCEL or the SPTAPE STOP command.

Migration Notes

VM/SP HPO

1. In VM/XA SP you may dump and load files using up to eight classes as selection criteria.
2. In VM/XA SP you may also dump and load NSS, UCR, IMG, NLS, and TRF files.
3. In VM/XA SP if you wish to specify one or more spoolids you must also specify a userid. VM/SP HPO (Release 5) users do not see this difference.
4. VM/XA SP gives a final COMPLETE response when the total operation is complete.
5. VM/XA SP does not support the DEST option.
6. VM/XA SP has no extended attribute buffer support.
7. VM/XA SP does not support the convert setting or the X'5A' CCW content.

VM/XA SF: In VM/XA SP if you wish to specify one or more spoolids you must also specify a userid.

SPTAPE CANCEL

Privilege Class: D, E

Use the SPTAPE CANCEL command to immediately terminate dumping, loading, or scanning of files on or to tape.

SPTape	CANCEL rdev
--------	-------------

where:

CANCEL

specifies that the operation should terminate immediately. The tape will then be rewound and unloaded, regardless of the previous tape positioning options. The tape (rdev) should be scanned from the current point to the first double tape mark encountered.

rdev

is the real device number of the tape drive used by SPTAPE.

Usage Notes

For more information, see “General Usage Notes for SPTAPE.”

Responses

For more information, see “General Responses for SPTAPE.”

SPTAPE DUMP

Privilege Class: D, E

Use the SPTAPE DUMP command to dump files to tape for backup storage or for migration.

SPTape	<pre> DUMP rdev [userid] { { Printer PUnch Reader } { spoolid1 [spoolid2] Class { abcdefgh * } FORM { form * } ALL } { NSS IMG UCR NLS TRFiles } { spoolid1 [spoolid2] Class { abcdefgh * } ALL } } [LEAVE] [REWind] [RUN] [MODE { 800 1600 6250 38K }] [[SYSHold] [USERHold] [NOHold]] [PURge] </pre>
--------	---

where:

DUMP

specifies designated spool or system data files are to be written from the spool file system to tape.

rdev

is the real device number of the tape drive used by SPTAPE.

userid

indicates that only those files owned by the specified userid are to be processed. If no userid is specified, files are processed regardless of userid. A userid must be specified if selecting files by spoolid.

PRINTER

PRT

indicates that printer-type spool files are to be written. This option may be used by class D users only.

PUNCH

PCH

indicates that punch-type spool files are to be written. This option may be used by class D users only.

READER

RDR

indicates that reader-type spool files are to be written. This option may be used by class D users only.

spoolid

indicates a spool file (spoolid1), or the start of a range of spool files ending with spoolid2, to be written.

END

indicates that all spool files on the selected queue with a spoolid greater than spoolid1 are to be written.

CLASS { abcdefgh }
 { * }

indicates that only those files with one or more of the specified classes are to be written to tape. From one to eight classes (single alphanumeric characters with no intervening blanks) may be specified. * indicates files are to be processed regardless of class.

FORM { form }
 { * }

indicates that only those files with the specified operator form number are to be written to tape. The form must be no more than 8 characters. Form may not be specified on NSS, IMG, or UCR files. * indicates files are to be processed regardless of form.

ALL

indicates that all files on the selected queue with the selected hold status are to be written to tape, regardless of the file's spoolid, class, or form.

NSS

indicates that named saved system (NSS) and saved segment files are to be written to tape.

IMG

indicates that image library files are to be written to tape.

UCR

indicates that user class override files are to be written to tape.

NLS

indicates that message repository files are to be written to tape.

TRFiles

indicates that system trace files are to be written to tape.

LEAVE

specifies that no tape movement occurs after the SPTAPE operation is performed. This is the default.

REWIND

specifies that the tape be rewound after the SPTAPE operation is performed.

RUN

specifies that the tape be rewound and unloaded after the SPTAPE operation is performed.

MODE

specifies the tape density. Mode values of 800, 1600, 6250, or 38K may be specified. 38K may only be specified when a 3480 tape drive is being used, and is the default for a 3480 tape drive. 1600 is the default for all other drives.

SYSHOLD

specifies that system-held files should be included in the SPTAPE operation. This option cannot be used with the NSS, IMG, UCR, NLS, or TRF options.

USERHOLD

specifies that user-held files should be included in the SPTAPE operation. This option cannot be used with the NSS, IMG, UCR, NLS, or TRF options.

NOHOLD

specifies that files that are not held should be included in the SPTAPE operation. This is the default. This option cannot be used with the NSS, IMG, UCR, NLS, or TRF options.

PURGE

specifies that the spool files are to be deleted from the spooling system after being successfully written to tape. This option cannot be used with the NSS, IMG, UCR, NLS, or TRF options; it is valid only for DUMP.

Usage Notes

1. Users who are class E and not class D may process only NSS, IMG, UCR, NLS, or TRF files.
2. Refer to the "General Usage Notes for SPTAPE."

Responses

Refer to the "General Responses for SPTAPE."

SPTAPE LOAD

Privilege Class: D, E

Use the SPTAPE LOAD command to load files from tape.

SPTape	<pre> LOAD rdev [userid] ((Printer) (spoolid1 [spoolid2])) (PUnch) ([END]) (Reader) (Class { abcdefgh }) (*) (FORM { form }) (*) (ALL) ((NSS) (spoolid1 [spoolid2])) (IMG) ([END]) (UCR) (Class { abcdefgh }) (NLS) (*) (TRFiles) (ALL) [LEAVE] [REWind] [RUN] [[SYSHold] [USERHold] [NOHold]] </pre>
--------	--

where:

LOAD

specifies that designated spool or system data files are to be read from tape to the spool file system.

rdev

is the real device number of the tape drive used by SPTAPE.

userid

indicates that only those files owned by the specified userid are to be processed. If no userid is specified, files are processed regardless of userid.

PRINTER

PRT

indicates that printer-type spool files are to be read. This option may be used by class D users only.

PUNCH

PCH

indicates that punch-type spool files are to be read. This option may be used by class D users only.

READER

RDR

indicates that reader-type spool files are to be read. This option may be used by class D users only.

spoolid

indicates a spool file (spoolid1), or the start of a range of spool files ending with spoolid2, to be read.

END

indicates that all spool files on the selected queue with a spoolid greater than spoolid1 are to be read.

CLASS { abcdefgh }
 { * }

indicates that only those files with one or more of the specified classes are to be loaded from tape. From one to eight classes (single alphanumeric characters with no intervening blanks) may be specified. * indicates files are to be processed regardless of class.

FORM { form }
 { * }

indicates that only those files with the specified operator form number are to be loaded. The form must be no more than 8 characters. Form may not be specified on NSS, IMG, or UCR files. * indicates files are to be processed regardless of form.

ALL

indicates that all files on the selected queue with the selected hold status are to be read from tape, regardless of the file's spoolid, class, or form.

NSS

indicates that named saved system (NSS) and saved segment files are to be read from tape.

IMG

indicates that image library files are to be read from tape.

UCR

indicates that user class override files are to be read from tape.

NLS

indicates that message repository files are to be read from tape.

TRFiles

indicates that system trace files are to be read from tape.

LEAVE

specifies that no tape movement occurs after the SPTAPE operation is performed. This is the default.

REWIND

specifies that the tape be rewound after the SPTAPE operation is performed.

RUN

specifies that the tape be rewound and unloaded after the SPTAPE operation is performed.

SYSHOLD

specifies that system-held files should be included in the SPTAPE operation. This option cannot be used with the NSS, IMG, UCR, NLS, or TRF options.

USERHOLD

specifies that user-held files should be included in the SPTAPE operation. This option cannot be used with the NSS, IMG, UCR, NLS, or TRF options.

NOHOLD

specifies that files that are not held should be included in the SPTAPE operation. This is the default. This option cannot be used with the NSS, IMG, UCR, NLS, or TRF options.

SPTAPE LOAD

Usage Notes

For more information, check under “General Usage Notes for SPTAPE.”

Responses

For responses, check under “General Responses for SPTAPE.”

SPTAPE SCAN

Privilege Class: D, E

Use the SPTAPE SCAN command to determine what files exist on the tape.

SPTape	SCAN rdev [userid]	[LEAVE REWind RUN]
--------	--------------------	----------------------------

where:

SCAN

specifies that the tape should be scanned from the current point to the first double tape mark encountered.

rdev

is the real device number of the tape drive used by SPTAPE.

userid

indicates that only those files owned by the specified userid are to be processed. If no userid is specified, files are processed regardless of userid.

LEAVE

specifies that no tape movement occurs after the SPTAPE operation is performed. This is the default.

REWIND

specifies that the tape be rewound after the SPTAPE operation is performed.

RUN

specifies that the tape be rewound and unloaded after the SPTAPE operation is performed.

Usage Notes

Refer to “General Usage Notes for SPTAPE.”

Responses

Refer to “General Responses for SPTAPE.”

SPTAPE STOP

SPTAPE STOP

Privilege Class: D, E

Use the SPTAPE STOP command to terminate processing after the current file.

SPTape	STOP rdev
--------	-----------

where:

STOP

specifies that the operation should terminate after completing the current file. The tape will then be positioned at the beginning of the next file.

rdev

is the real device number of the tape drive used by SPTAPE.

Usage Notes

Issuing SPTAPE STOP between tape reel mounts during a multireel spool file dump has the same affect as issuing SPTAPE CANCEL.

Responses

Refer to "General Responses for SPTAPE."

START

Privilege Class: D

Use the START command to restart a spooling device after it has been drained. It can also be used to change the processing options currently in effect for the device.

STArt	[rdev [options] ...]
	[Reader Printer PUnch <u>ALL</u>]

where:

rdev ...

is the real device number of a particular spooling device that you want to start. You can specify more than one real device, each with its own set of options.

options

See "START: Options" on page 604.

READER

RDR

PRINTER

PRT

PUNCH

PUN

ALL

starts your readers, printers, or punches. Specify **ALL** to start all of your readers, printers, and punches.

Migration Notes

VM/SP HPO

1. VM/XA SP does not support the FILEFCB and CFILEFCB options, which are supported in VM/SP HPO.
2. VM/XA SP does not support imbedded LOAD FCB CCWs, which are supported in VM/SP HPO.
3. The VM/XA SP response does not indicate processor addresses.
4. VM/XA SP has a SEP option as well as NOSEP.
5. VM/XA SP does not support the DEST option.
6. VM/XA SP does not support logical printers.
7. Information in the VM/XA SP response has been reordered or changed: FORM information appears on a different line; the response indicates whether the device was started for the system or for a specific userid; the response provides additional information about impact printers.

VM/XA SF: NOSEP must be specified on each START command if no separator pages are to be printed. (In the past, if NOSEP was specified, it remained in effect until another START command with SEP was specified.)

If no **START** command with the **USER** option has been issued since the last cold start, **SYSTEM** is the default.

CLASS { **abcdefgh** }
 { * }

specifies the output classes that the device can service. You can specify from one to eight classes (each a single character, with no intervening blanks). If you are specifying multiple classes for your device, your files will be serviced in the same order as the classes that you have specified. For example, if you specify the following for your printer:

CLASS AD1

all class A files are processed before class D files, and all class D files before class 1 files.

The **CLASS** operand is invalid for a reader.

Specify a class of * for printers or punches so that they can service any spool file class.

If no **START** command with the **CLASS** option has been issued since the last cold start, the default classes specified in the **RDEVICE** macro (during system generation) are used.

FORM { **form** }
 { * }

specify the form that an output device can service. 'form' is a 1- to 8-character operator form number. This operand is ignored when the printer is in **AUTO** mode.

Specify '*' for printers or punches so that files can be printed or punched regardless of form.

If no **START** command with the **FORM** option has been issued since the last cold start, the default form specified in the **SYSFORM** macro (during system generation) is used. If the **SYSFORM** macro was not specified, the default is **STANDARD**.

MANUAL

indicates that the operator will schedule output forms. Only spool files with the form currently active on the device are processed. When the last spool file with the current form is printed or punched, the output device stops. The operator can then schedule a new form by issuing a **START** command with the **FORM** option.

If 'form *' is specified, spool files are printed or punched regardless of the form, and the output device will not stop between spool files with different forms.

AUTO

indicates that the operator wants CP to schedule forms output. The currently active form is processed first. When all files with the current form have been printed or punched, CP automatically selects a new form for the device. The operator will receive a message to change forms, and the output device will be idle. The operator can then put in the new form and press the **START** button (or issue the **START** command) to print files with the new form.

SETUP

indicates that the operator wants CP to print a page so that forms alignment can be verified. (This option is only valid for impact printers). The operator will get a message indicating that he or she should set up the forms. When the form has been set up, the operator should press the **STOP** button, then the **START**

button. An alignment page will then be printed. It is the first page of the file, with letters replaced with X's and numbers replaced by 9's. Each time the operator presses the STOP, START buttons after a full alignment page has been printed, another alignment page will be printed. When the forms are aligned, you should issue the 'START' command for the device. The printer will then return to its previous mode (AUTO or MANUAL), and the file will print in its entirety.

If neither MANUAL, AUTO, nor SETUP is specified and the printer is currently in SETUP mode, it will return to the previous mode, AUTO or MANUAL. If no START command has been issued with the MANUAL, AUTO, or SETUP options since a system cold start, the default is MANUAL.

SEP

NOSEP

indicates whether you want a file separator for output files. This option is invalid for readers.

If you **do not** want separator pages to be printed, you must always say NOSEP.

If no START command with the SEP or NOSEP option has been issued since the last cold start, the default is specified in the RDEVICE macro.

NO3800

specifies that no spool files will be printed that contain 3800 load CCWs. Since load CCWs issued to 3800 printers can take a great deal of time, this option insures a fast output queue.

BEG3800

specifies that only spool files without 3800 load CCWs, or that contain 3800 load CCWs only at the beginning of the file, will be printed on this device. This allows the user an initial setup with which the entire file will be printed.

ANY3800

places no restrictions on the presence or absence of 3800 load CCWs within spool files. This option can potentially cause a slower output queue, depending on the number of 3800 load CCWs issued by the selected spool files.

If no START command with the NO3800, BEG3800, or ANY3800 options has been issued since the last cold start, the default is NO3800.

IMAGE imagelib

specifies the name of the system image library file which will contain the load images (character sets, FCBs, graphic character modifications, or copy modifications) that will be used for printing files and separators. 'imagelib' may be from 1 to 8 characters in length, and must be an existing image library file.

If no START command with the image option has been issued since the last cold start, the default imagelib specified on the RDEVICE macro (at installation time) is used.

CHARS name

specifies the character set to be used by the printer when printing both the separator pages and the file. For 3800 printers, the character set specified is used for the separator pages and only those spool files which do not request a specific character set. 'name' is 1 to 4 characters long. This character set name must reside in the image library specified for this device.

For impact printers that support a character set the CHARS specified is used for the entire file.

If no START command with the CHARS option has been issued since the last cold start, the default CHARS specified on the RDEVICE macro (at installation time) is used.

Note: This option is not valid for 1403 printers without the Universal Character Set feature installed, 4245, or 4248 printers.

FCB { name }
 { 6 }
 { 8 }
 { 10 }
 { 12 }

specifies the forms control buffer used by the printer when printing both the separator pages and the file. For 3800 printers, the FCB specified is used for separator pages and only those spool files which do not request a specific FCB.

For impact printers (except the 1403 which does not support the FCB), the FCB specified is used for the entire file.

'name' is either a 1- to 4-character name of an FCB image in the image library specified for this device, or the number 6, 8, 10, or 12. This number tells the 3800 to print the entire spool file at 6, 8, 10, or 12 lines per inch, regardless of the size of the paper that is currently loaded.

Note: The number 10 is only supplied for 3800 model 3 printers.

Specifying an FCB of 6, 8, 10, or 12 are only special cases on 3800 printers. If an FCB of 6, 8, 10, or 12 is specified for an impact printer, the START command will try to find an image library member with the corresponding name (for example, "32036" if "FCB 6" is specified for a 3203 printer).

If no START command with an FCB option has been issued since the last cold start, the default FCB specified on the RDEVICE macro (at installation time) is used.

FOLD
NOFOLD

specifies the need to load the universal character set (UCS) buffer with an operation code that permits printing of uppercase and lowercase bit configurations. If FOLD is specified, all characters will be printed in uppercase. This is used when lowercase is specified in a file, but no lowercase exists in the UCS specified. If NOFOLD is specified, the characters are not translated into uppercase.

Note: If you specify the FOLD option, the control unit ignores bit positions 0 and 1 of the EBCDIC code when comparing the character codes from the UCS buffer and the print line buffer. Only uppercase characters will be printed from either uppercase or lowercase data codes. **Warning:** Misuse of the FOLD option can cause multiple characters to be printed for the same EBCDIC bit configuration.

INDEX nn

specifies the position at which to begin printing. nn is a number from 1 to 31; a leading zero need not be specified. (This option is valid only for 3211 type printers.)

FLASH { name }
 { OFF }

specifies the forms overlay negative loaded into the device. Spool files designated to be FLASHed with 'name', and spool files with no flash name associated with them, will be selected for printing. "name" is 1 to 4 alphanumeric characters.

If 'OFF' is specified, it indicates that no flash will be used.

If no START command with the FLASH option has been issued since the last cold start, the default FLASH name specified on the RDEVICE macro (at installation time) is used.

HOLD

PURGE

specifies the system action to be taken on files with 3800 load errors. If HOLD is specified, all files with load errors will be placed in system hold status. If PURGE is specified, all files with load errors will be purged.

If no START command with the HOLD or PURGE option has been issued since the last cold start, the default is HOLD.

Usage Notes

1. If a device is busy when you issue the START command, the changes that you specify will take effect after the current file is complete.
2. If you specify the "NOSEP" operand for a punch, separator cards will not be punched. Instead, for each file the operator will receive the message:

```
PUN rdev START FOR OUTPUT
```

The punch can be started for the next file by pressing the START/STOP key on the device.
3. If a real spooling printer that is not a 3800 is STARTed with the BEG3800 or ANY3800 options, all 3800 load CCWs contained in selected spool files will be removed, and the file will be printed with the available setup.
4. IBM does not recommend that you use spool file class to determine the output form (paper) for 3800 operation. If you change the paper on the 3800 printer without specifying a new form name with the FORM operand of the START command, 3800 errors will result.

Accordingly, when you change the paper on a 3800 printer, IBM recommends that you start the printer with the FORM operand and a new form name. But if you prefer, you may instead drain the printer (with the DRAIN command) and issue the START command without the FORM operand.
5. If you are working within CMS and you want to issue a CP START command, use #CP START.
6. The DEFFCB option (which allows a user to specify that the default FCB that was loaded in the printer should be used) will be accepted as a parameter on the START command for a 3211 type printer only, for compatibility, but will be ignored because VM/SP currently uses the default FCBs loaded in by the LOADBUF command.

Responses

Responses For a Real Reader

```
RDR rdev STARTED {SYSTEM}
                  {userid}
```

Responses For a Real Printer

```
PRT rdev STARTED {SYSTEM} CLASS abcdefgh
                  {userid}
```

```
PRT rdev FORM operform {MANUAL} {SEP} {N03800}
                       {AUTO}  {NOSEP} {BEG3800}
                       {SETUP}  {      } {ANY3800}
```

In addition, when a 3800 printer is started, the following response is also generated.

```
PRT rdev FLASH name IMAGE imagelib CHARS name FCB {name} {HOLD }
                                                    {PURGE }
                                                    {
 6
 8
10
12 }
```

Responses For Impact Printers Only

When an impact printer is started, the following response appears for all impact printers except the 1403 printers which do not have the universal character set feature installed.

```
PRT rdev {NOFOLD} IMAGE imagelib CHARS chars [FCB name] [INDEX nn]
         {FOLD  }
```

appears for all impact printers except for 1403 printers which do not have the universal character set feature installed.

where:

NOFOLD

FOLD

indicates whether lowercase characters are being translated (folded) into upper case for printing.

IMAGE imagelib

indicates the image library being used by the printer.

CHARS chars

indicates the character set being used by the printer.

FCB {name}

indicates the FCB being used by the printer.

STOP**Privilege Class: G**

Use the STOP command to place all your virtual processors into soft-stop, or to place selected processors into hard-stop.

STOP	[CPU { cpuaddr cpuaddr1 cpuaddr... cpuaddr1-cpuaddrn ALL... }]
------	--

where:

CPU {
 cpuaddr
 cpuaddr1 cpuaddr...
 cpuaddr1-cpuaddrn
 ALL

specifies the processor or processors you want to place in hard-stop. You can specify a single processor address, a list of processor addresses delimited by blanks, a nonwrapping pair of addresses separated by a “-” with no blanks, or ALL. If you omit this operand, all of your virtual processors are placed into soft-stop.

Usage Notes

1. If you issue the STOP command without operands, all of your virtual processors are placed into soft-stop.
2. This command is useful with a TRACE command to suspend execution when a certain condition is reached.
3. Execution of the virtual processor or processors is suspended until a BEGIN, SYSTEM RESTART, ATTN, IPL, EXTERNAL, or REQUEST command is entered.
4. The STOP command causes suspension of virtual machine execution regardless of the setting of SET RUN.

Responses

None.

STORE

Use the STORE command described in this section to alter any of the following locations in your real or virtual machine:

- First-level storage
- Second-level storage
- Third-level storage
- General registers
- Floating-point registers
- Control registers
- Storage keys
- The current program status word (CURRENT PSW)
- Channel address words (CAWs)
- Channel status words (CSWs)
- Vector Facility registers.

Complete Format for STORE

The format box shows all of the operands available with the STORE command. Complete descriptions of the operands appear on the following pages, in the same sequence as the operands appear here.

STore	$\left[\begin{array}{c} R \\ \underline{L} \\ V \\ W \end{array} \right] \left[\begin{array}{c} N \\ S \\ K \\ U \end{array} \right] \text{ hexloc data}$	G
	$H \left[\begin{array}{c} N \\ S \\ K \\ U \end{array} \right] \text{ hexloc data}$	C
	$\left\{ \begin{array}{l} \text{Psw} \\ \text{CAW} \\ \text{CSW} \end{array} \right\} \text{ hexword1 } [\text{hexword2}]$	G
	$\left\{ \begin{array}{l} \text{Grg1} \\ \text{Xrg1} \end{array} \right\} \text{ hexword1 } [\text{hexword2 } \dots]$	G
	$\{ \text{Yrg1} \} \text{ hexdword1 } [\text{hexdword2 } \dots]$	
	STATUS	G
	$\left[\begin{array}{l} \text{VR} \left[\left[\begin{array}{c} \text{reg} \\ \underline{0} \end{array} \right] \left[\begin{array}{c} \text{,elt} \\ \underline{0} \end{array} \right] \text{ hexword1 } [\text{hexword2} \dots] \end{array} \right] \end{array} \right]$	G
	$\left[\begin{array}{l} \text{VP} \left[\left[\begin{array}{c} \text{reg} \\ \underline{0} \end{array} \right] \left[\begin{array}{c} \text{,elt} \\ \underline{0} \end{array} \right] \text{ hexdword1 } [\text{hexdword2} \dots] \end{array} \right] \end{array} \right]$	
	VSR hexword1 [hexword2]	G
	VMR hexword1 [hexword2...]	G
	VAC hexdword	G

STORE

Migration Notes

VM/SP HPO: VM/XA SP does not support storing into odd-numbered floating point registers.

VM/XA SF: In VM/XA SP, the default option is N.

STORE GUEST STORAGE

Privilege Class: G

Use STORE GUEST STORAGE to store data in second-level or third-level storage.

STore	<table border="1"> <tr> <td>R</td> <td>N</td> </tr> <tr> <td>L</td> <td>S</td> </tr> <tr> <td>V</td> <td>K</td> </tr> <tr> <td>W</td> <td>U</td> </tr> </table>	R	N	L	S	V	K	W	U	hexloc data
R	N									
L	S									
V	K									
W	U									

Note: Do not put blanks between the operands and the address specified by hexloc. However, at least one blank must precede the data that you want to insert into storage.

where:

R

stores your data in second-level storage.

L

stores into second-level storage if the guest virtual machine is in DAT OFF mode, stores into third-level storage if the guest virtual machine is in DAT ON mode.

V

stores into third-level storage using the standard STD as indicated by control register 1.

W

stores into third-level storage using the alternate working segment table designation (STD) as indicated by control register 7.

N

stores the specified data (hexadecimal) in successive fullword locations starting at the address specified by hexloc. The smallest group of hexadecimal values that can be stored is one fullword. This is the default.

If hexloc is not on a fullword boundary, the store operation will begin on the next lower fullword boundary. If the value being stored is less than one fullword (eight hexadecimal digits) in length, it is right-adjusted in the word and is filled in with zeros. If two or more hexadecimal words are specified, they must be separated with one or more blanks.

S

stores the specified data (hexadecimal) without word alignment starting at the address specified in hexloc. The shortest string that can be stored is one byte (2 hexadecimal digits).

If the string contains an odd number of characters, the last character is not stored, an error message is issued, and the function terminates. If you specify the S operand, the data you enter must be a string of two or more hexadecimal digits with no embedded blanks.

K

stores the data in the storage key for the 4Kb page in which hexloc is located. Two hexadecimal digits are used to form the key for the page. If one digit is provided, it becomes the lower digit and the upper digit is 0.

STORE GUEST STORAGE

U

stores the specified data (EBCDIC characters) starting at the address specified by hexloc. The data is stored in byte segments with no word alignment.

hexloc

is the starting storage location that is to be altered. The address hexloc may be one to eight hexadecimal digits; leading zeros are optional. The hexloc parameter has no defaults; you must always specify the starting address that you want to alter.

data

is the data you want to insert into storage. data takes on different meanings, depending on the type of STORE you are performing (N, S, K, or U). See the descriptions of each of these operands for the meaning of data.

Usage Notes

An attempt to store at an address in a protected page, or to change the key at an address in a protected page, will result in an error message. A protected page is a page of read-only storage.

Responses

STORE COMPLETE

indicates that the data has been stored.

STORE HOST STORAGE

Privilege Class: C

Use STORE HOST STORAGE to store data in first level storage.

STore	<table style="display: inline-table; border: none;"> <tr> <td style="border: none; padding-right: 5px;">H</td> <td style="border: none; padding-right: 5px;">[</td> <td style="border: none; padding-right: 5px; text-align: center;">N</td> <td style="border: none; padding-right: 5px;">]</td> <td style="border: none; padding-right: 5px;">hexloc</td> <td style="border: none;">data</td> </tr> <tr> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;">[</td> <td style="border: none; padding-right: 5px; text-align: center;">S</td> <td style="border: none; padding-right: 5px;">]</td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none;"></td> </tr> <tr> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;">[</td> <td style="border: none; padding-right: 5px; text-align: center;">K</td> <td style="border: none; padding-right: 5px;">]</td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none;"></td> </tr> <tr> <td style="border: none; padding-right: 5px;"></td> <td style="border: none; padding-right: 5px;">[</td> <td style="border: none; padding-right: 5px; text-align: center;">U</td> <td style="border: none; padding-right: 5px;">]</td> <td style="border: none; padding-right: 5px;"></td> <td style="border: none;"></td> </tr> </table>	H	[N]	hexloc	data		[S]				[K]				[U]		
H	[N]	hexloc	data																				
	[S]																						
	[K]																						
	[U]																						

Note: Do not put blanks between the operands and the address specified by hexloc. However, at least one blank must precede the data that you want to insert into storage.

where:

H

indicates that first level storage is to be altered.

N

stores the specified data (hexadecimal) in successive fullword locations starting at the address specified by hexloc. The smallest group of hexadecimal values that can be stored is one fullword. This is the default.

If hexloc is not on a fullword boundary, the store operation will begin on the next lower fullword boundary. If the value being stored is less than one fullword (four bytes) in length, it is right-adjusted in the word and is filled in with zeros. If two or more hexadecimal words are specified, they must be separated with one or more blanks.

S

stores the specified data (hexadecimal) without word alignment starting at the address specified in hexloc. The shortest string that can be stored is one byte (2 hexadecimal digits). If the string contains an odd number of characters, the last character is not stored, an error message is issued, and the function terminates. If you specify the S operand, the data you enter must be a string of two or more hexadecimal digits with no embedded blanks.

K

stores the data in the storage key for the 4Kb page in which hexloc is located. Two hexadecimal digits are used to form the key for the page. If one digit is provided, it becomes the lower digit and the upper digit is 0.

U

stores the specified data (EBCDIC characters) starting at the address specified by hexloc. The data is stored in byte segments with no word alignment.

hexloc

is the starting storage location that is to be altered. The address hexloc may be one to eight hexadecimal digits with optional leading zeros. The hexloc parameter has no defaults; you must always specify the starting address that you want to alter.

data

is the data you want to insert into storage. Data take on different meanings, depending on the type of STORE you are performing (N, S, K, or U). See the descriptions of each of these operands for the meaning of data.

STORE HOST STORAGE

Warning: Exercise extreme caution in issuing the STORE HOST STORAGE command. Careless storage alterations can cause abnormal terminations of your system as well as security and integrity problems.

Usage Notes

You can only alter CP owned pages with STORE HOST STORAGE.

Responses

STORE COMPLETE

indicates that the data has been stored.

STORE PSW/CAW/CSW

Privilege Class: G

Use STORE PSW/CAW/CSW to store data in the current virtual machine program status word (PSW), channel address word (CAW), or channel status word (CSW).

Store	$\left\{ \begin{array}{l} \text{Psw} \\ \text{CAW} \\ \text{CSW} \end{array} \right\}$ hexword1 [hexword2]
-------	--

where:

PSW

alters all or part of the PSW with the data you specify in hexword1 and hexword2. If you specify only hexword1, it is stored into the address portion of your virtual machine's current PSW. If you specify both hexword1 and hexword2, hexword1 is stored into the first word of your virtual machine's current PSW and hexword2 is stored into the address portion of the PSW.

CAW

alters the channel address word (CAW). This operand is valid for System/370 mode virtual machines only; the 370-XA architecture equivalent, the operation request block (ORB), has no fixed location and cannot be located.

CSW

alters the channel status word (CSW). The value you specify in hexword1 is stored into the address portion of your virtual machine's channel status word, and the value you specify in hexword2 (if given) will be stored into the status word portion of your virtual machine's channel status word. This operand is valid for System/370 mode virtual machines only; the 370-XA architecture equivalent, the interruption response block (IRB), has no fixed location and cannot be located.

hexword1 [hexword2]

is the data you want to store in the PSW, CAW, or CSW. If the data is less than 8 digits (4 bytes), it will be right adjusted and padded with leading zeros.

Usage Notes

None.

Responses

STORE COMPLETE

is the response at the successful completion of the command.

STORE REGISTERS

Privilege Class: G

Use STORE REGISTERS to store data into general purpose registers, floating-point registers, or control registers.

STore	{Greg1} hexword1 [hexword2 ...]
	{Xreg1}
	{Yreg1} hexdword1 [hexdword2 ...]

where:

Greg1

is a decimal number from 0 to 15 or a hexadecimal number from 0 to F representing the first, or only, general register whose contents are to be assigned a value. You must prefix the value with the letter G.

Xreg1

is a decimal number from 0 to 15 or a hexadecimal number from 0 to F representing the first, or only, control register that is to be assigned a value. You must prefix the value with the letter X.

Yreg1

is an integer (0, 2, 4, or 6) representing the first, or only, floating-point register whose contents are to be assigned a value. You must prefix the value with the letter Y.

hexword1

hexword2...

each represents up to eight hexadecimal digits for G and X register stores. If the value being stored is less than the eight digits, it is right-adjusted in the register and the high-order bytes of the register are filled with zeros. If two or more values are specified, they must be separated by one or more blanks. The data is stored in successive registers.

hexdword1

hexdword2

each represents up to 16 hexadecimal digits for Y register stores. If the value being stored is less than 16 digits, it is left-adjusted in the register and the low-order bytes of the register are filled with zeros. If two or more values are specified, they must be separated by one or more blanks. The data is stored in successive registers.

Responses

STORE COMPLETE

indicates that the data has been stored.

STORE STATUS

Privilege Class: G

Use **STORE STATUS** to store selected virtual machine data in certain low storage locations.

STore	STATUS
-------	--------

where:

STATUS

stores selected virtual machine data in certain low storage locations of the virtual machine, simulating the hardware store status facility. These locations are permanently assigned locations in guest absolute storage.

The **STORE STATUS** operand causes the following data to be stored in low storage:

Decimal Address	Hexadecimal Address	Length in Bytes	Data
216	D8	8	CPU Timer
224	E0	8	Clock Comparator
256	100	8	Current PSW
264	108	4	Prefix register
352	160	32	Floating-point registers 0-6
384	180	64	General registers 0-15
448	1C0	64	Control registers 0-15

Usage Notes

1. The **STORE STATUS** command stores data in the logout area.
2. The **STORE STATUS** command stores CPU timer and clock comparator values that may then be displayed on your console via the **DISPLAY** command. This procedure is one way to get timer information on your display.

Responses

STORE COMPLETE

is the response at the successful completion of the command.

STORE VECTOR

Privilege Class: G

Use **STORE VECTOR** to alter the contents of one or more vector register elements or vector register pair elements, vector activity count, vector status register, or vector mask register.

Store	$\left[\begin{array}{l} \text{VR} \left[\left[\begin{array}{l} \text{reg} \\ \underline{0} \end{array} \right] \left[\begin{array}{l} \text{,elt} \\ \underline{0} \end{array} \right] \text{hexword1} \text{ [hexword2...]} \right] \\ \text{VP} \left[\left[\begin{array}{l} \text{reg} \\ \underline{0} \end{array} \right] \left[\begin{array}{l} \text{,elt} \\ \underline{0} \end{array} \right] \text{hexdword1} \text{ [hexdword2...]} \right] \end{array} \right]$
	VSR hexword1 [hexword2]
	VMR hexword1 [hexword2...]
	VAC hexdword

where:

VR

allows you to store into one or more elements of one vector register.

VP

allows you to store into one or more elements of one vector register pair.

reg

If you use **VR**, **reg** is a decimal number from 0 to 15 or a hexadecimal number from 0 to F indicating the vector register whose elements you want to store a new value.

If you use **VP**, **reg** is a decimal number from 0 to 14 or a hexadecimal number from 0 to E indicating the even number of the vector register pair into whose elements you want to store a new value.

If you specify **VR** or **VP** without a register number, the system will store in vector register 0 or vector register pair 0-1.

elt

elt is a hexadecimal number from 0 to one less than the section size representing the first, or only, element whose content is to be altered.

hexword 1 [hexword2 ...]

is the data you wish to store in consecutive elements of the vector register you specified. The hexword operands may each be up to eight hexadecimal digits. If the value is less than a fullword, it is right-adjusted in the register element and its high-order bytes are filled with zeros. If you specify two or more hexwords, they must be separated by one or more blanks. If the number of hexwords listed plus the starting element number exceeds the section size, then the valid hexword is stored and an error message is issued.

hexdword1 [hexdword 2...]

is the data you wish to store in consecutive elements of the vector register pair you specified. The hexdword operands may each be up to 16 hexadecimal digits. If the value is less than a doubleword, it is left-adjusted in the register element and its low-order bytes are filled with zeros. If you specify two or more hexdwords, they must be separated by one or more blanks. If the number of

hexdwords listed plus the starting element number is larger than the section size, then the system stores the valid hexdwords and ignores the excess hexdwords.

VSR hexword1 [hexword2]

stores the data in hexword1 (and hexword2 if you specify hexword2) in the virtual machine's vector status register. The hexword may each be up to eight hexadecimal digits. If the value is less than a fullword, it is right-adjusted in the register word being altered and its high-order bytes are filled with zeros. If you specify two or more hexwords, they must be separated by one or more blanks. This fullword value is then checked to conform to the architecture of the vector status register. The system corrects any inconsistencies.

VMR hexword1 [hexword2 ...]

stores the data in hexword1 and the other specified hexwords in the virtual machine's vector mask register. The hexword operands may each have up to eight hexadecimal digits. If the value is less than a fullword, it is left-adjusted in the word and its low-order bytes are filled with zeros. If you specify two or more hexwords, they must be separated by one or more blanks. The number of bytes which can be stored in the VMR is limited to the section size. If the combined operand is longer than the length of the vector mask register, the system ignores the excess digits.

VAC hexdword

stores the data in hexdword in the virtual machine's vector activity count. The operand may represent up to 16 hexadecimal digits. If the value stored is less than a doubleword, it is right-adjusted in the word and the leftmost bits are filled with zeros. This hexdword value is then checked to conform to the architecture of the vector activity count. The system corrects any inconsistencies.

Usage Notes

STORE VR and STORE VP will turn on the vector in-use and change bits corresponding to the register or pair being changed. If the in-use bit were off previously, then the elements of that register pair will be zero as nothing will be stored into them.

Examples

To store hexadecimal values 4E5400005E450000 and 4E5500005E550000 into the vector register pair 6 and 7, enter:

```
ST VP6 4E5400005E450000 4E5500005E550000
```

To store hexadecimal value 0000000200000000 into the virtual machine's vector activity count, enter:

```
ST VAC 0000000200000000
```

To store hexadecimal value 00000001 000000FF into the virtual machine's vector status register, enter:

```
STORE VSR 00000001 000000FF
```

To store hexadecimal value FFFFFFFF FFFF0000 00000000 00000000 into the virtual machine's vector mask register, enter:

```
ST VMR FFFFFFFF FFFF0000 00000000 00000000
```

STORE VECTOR

Responses

STORE COMPLETE

indicates that the data has been stored.

Migration Notes

VM/SP HPO: VM/XA SP evaluates the command for proper syntax prior to checking whether the Vector Facility is available.

SYSTEM

Privilege Class: G

Use the **SYSTEM** command to clear storage and simulate the action of **RESET** and **RESTART** on a real computer console.

SYStem	{ CLEAR RESET RESTART STORE STATUS }
--------	---

where:

CLEAR

clears virtual storage, virtual storage keys, floating and general purpose registers in all processors, and resets the I/O configuration. All processors are reset and placed in hard-stop.

RESET

resets the processors and the I/O configuration. All processors in the user's configuration are placed in hard-stop.

RESTART

simulates the hardware system **RESTART** function by storing the current PSW at second level location eight and loading, as the new PSW, the doubleword from virtual location zero. Interrupt conditions and storage remain unaffected.

STORE STATUS

stores selected virtual machine data in certain low storage locations of the virtual machine, simulating the hardware store status facility. These locations are permanently assigned locations in guest absolute storage.

The **STORE STATUS** operand causes the following data to be stored in low storage:

Decimal Address	Hexadecimal Address	Length in Bytes	Data
216	D8	8	CPU Timer
224	E0	8	Clock Comparator
256	100	8	Current PSW
264	108	4	Prefix register
352	160	32	Floating-point registers 0-6
384	180	64	General registers 0-15
448	1C0	64	Control registers 0-15

Usage Notes

1. The **RESET** function and the **CLEAR** function leave the virtual machine in a stopped state.
2. To resume operation after issuing the **SYSTEM** command with **RESET** specified, either **STORE** a PSW and issue **BEGIN** or issue **BEGIN** with a hexadecimal storage location specified. When doing either of these, be sure the wait state bit in the PSW is turned off before issuing **BEGIN**. The virtual machine automatically restarts at the location specified in the new PSW (which

SYSTEM

is loaded from the doubleword at location zero) after the SYSTEM RESTART command is processed.

3. The SYSTEM CLEAR or SYSTEM RESET commands will clear a virtual check-stop condition on the virtual processors.
4. The SYSTEM CLEAR and SYSTEM RESET commands terminate all IUCV connections and release the IUCV buffer.
5. The SYSTEM STORE STATUS command stores data in the logout area. The SYSTEM STORE STATUS command stores CPU Timer and Clock Comparator values that may then be displayed on your display via the DISPLAY command. This procedure is one way to get timer information on your display.
6. The SYSTEM CLEAR and the SYSTEM RESET commands cause the virtual machine to be reset. If the virtual machine was a member of a virtual machine group (from a prior IPL of a VMGROUP NSS), that group membership is canceled. The virtual machine's IUCV connection to the signal system service is severed and a signal-out message is broadcast by the signal system service to all other members of the group enabled for signal-out notification.
7. If your virtual machine uses the DIAGNOSE X'98' facilities for locking virtual machine pages in real storage, issuing SYSTEM CLEAR unlocks those pages in certain cases. If you have locked pages in private virtual machine storage, SYSTEM CLEAR unlocks those pages. SYSTEM CLEAR also releases all NSSs and saved segments from your virtual machine. Locked shared pages are unlocked only by DIAGNOSE X'98' or by the release of the shared segment by the last user.

Responses

STORAGE CLEARED - SYSTEM RESET

is the response given if you enter the SYSTEM CLEAR command.

SYSTEM RESET

is the response given if you enter the SYSTEM RESET command.

STORE COMPLETE

is the response given if you enter the SYSTEM STORE STATUS command.

If you enter the command SYSTEM RESTART, no response is given; the virtual machine resumes execution at the address in the virtual PSW loaded from virtual storage location zero.

Migration Notes

VM/SP HPO

1. In VM/XA SP if you specify SYSTEM CLEAR, the system clears the registers and resets the I/O configuration;
2. VM/XA SP supports an additional operand: STORE STATUS.
3. In VM/XA SP, SYSTEM STORE STATUS is the same as the VM/SP HPO STORE STATUS.

VM/XA SF: During a system reset, VM/XA SP does not reset the CONMODE setting of a virtual console.

TAG

Privilege Class: G

Use the TAG command to associate descriptive information with VM/XA SP spool files or spooled devices.

TAG	[Query]	{	[DEV]	{	Printer Punch CONSOLE vdev	}	}	[tagtext]
					FILE spoolid			

where:

QUERY

requests the display of the tag information that is associated with a particular virtual spooled device or spool file.

DEV

specifies a device is to be tagged or the tag information for a device will be displayed.

PRINTER

PRT

PUNCH

PCH

CONSOLE

vdev

specifies a spooling device whose output is to be assigned the tag information. If you enter a generic device name, such as PRINTER or CONSOLE, all current virtual devices of that type are affected.

FILE spoolid

indicates a closed spool file that you want to assign tag information. You can use this form of the TAG command only for spool files that are queued on your reader, printer, or punch queues.

spoolid is the spool file identification that CP assigned when the spool file was closed.

tagtext

is the information that you want to associate with the specified spool device or spool file. The contents and format of this data are flexible; they are the responsibility of the file originator and the end user.

Usage Notes

1. If you enter the TAG command without tagtext information and without the QUERY option, the tag area associated with the device or file is set to all blanks.
2. Even though you have created a spool file, once you spool it to another virtual machine, the other user owns it as soon as the file is closed. If you want to query or change the tag data, you must first reclaim the file by issuing the CP CHANGE or CP TRANSFER command.

TAG

3. The buffer that contains TAG information is 136 characters long, including blanks. However, because specific terminals may allow fewer than 136 characters, a user may not be able to type in that many characters on the command line.

Response 1

```
type vdev TAG:  
tagtext
```

displays the tag information associated with a virtual spooling device.

where:

type
is PRT, PUN, or CON

vdev
is the virtual device number.

tagtext
is the tag information associated with the device. This tag information may be up to 136 characters long.

Response 2

```
tagtext
```

- or -

```
(TAG BLANK)
```

displays the tag information associated with a specified pool file.

where:

tagtext
is the tag information associated with the file. This tag information may be up to 136 characters long.

(TAG BLANK)
is displayed when the tag is all blanks.

Response 3

```
type {vdev } TAG NOT SET
```

displays the tag information associated with a device if:

- You have not specified tag information
- You have explicitly set the tag fields to blanks

where:

type
is PRT, PUN, or CONS.

vdev
is the virtual device number.

Migration Notes

VM/SP HPO: The situation that produced the response TAG MISSING in VM/SP HPO now produces the response TAG BLANK in VM/XA SP.

TERMINAL

Use the **TERMINAL** command described in this section to control various characteristics of your virtual console.

The format box below shows all of the operands available with the **TERMINAL** command. Complete descriptions of the operands appear on the following pages, in the same sequence as the operands appear in the format box.

TERMinal	APL	{ ON } { OFF }
	ASCIitbl	{ VM1 } { VM2 }
	ATtn	{ OFF } { ON }
	BREakin	{ IMmed } { GUESTctl }
	BRKkey	{ key } { NONE }
	CHardel	{ ON } { OFF } { char }
	CONmode	{ 3215 } { 3270 }
	CNTL	{ SYS } { USR }
	EScape	{ ON } { OFF } { char }
	HIlight	{ ON } { OFF }
	LINEDEL	{ ON } { OFF } { char }
	LINENd	{ ON } { OFF } { char }

TERMinal	LINESize	{ nnn } { OFF }
	MODE	{ CP } { VM }
	PROMpt	{ VM } { TTY }
	SCR011	{ nnn } { CONT }
	SCRNsave	OFF
	TABchar	{ ON } { OFF } { char }
	TEXT	{ ON } { OFF }
	TYpe	{ 3101 } { TTY }

General Usage Notes for The TERMINAL Command

1. You can specify more than one function in a single entry of the TERMINAL command. For example:

```
TERMINAL CHARDEL OFF LINESIZE 80
```

specifies two terminal characteristics:
 - the character delete function CHARDEL is set OFF
 - the maximum line length for terminal output LINESIZE is set to 80 characters.
2. The settings you specify with the TERMINAL command are in effect only for the duration of that session. When you initially log on, the system defaults are in effect. However, the settings you specify for line-editing (except LINEDEL) and MODE are still in effect when you log on after disconnecting. All other operands (TEXT, APL and LINESIZE) are reset if you log on after disconnecting.
3. If an error occurs during processing of the command, all functions preceding the one with the error are in effect.
4. You cannot use any of the letters A through Z, or the numbers 0 through 9, as a symbol (char) in association with any of the CHARDEL, LINEDEL, LINEND, TABCHAR, and ESCAPE operands.
5. You can use the CP QUERY TERMINAL command to determine the values for the TERMINAL command functions.
6. CP automatically masks your password on typewriter terminals; however, to maintain compatibility with previous releases, CP accepts the MASK operand of the TERMINAL command as valid.

TERMINAL

Migration Notes

VM/SP HPO

1. VM/XA SP allows the user to set OFF the SCRNSAVE option; however, an error message results whenever it is set ON.
2. In VM/XA SP the system ignores the TERMINAL BRKKEY function in line mode.
3. VM/XA SP does not allow BREAKIN GUESTCTL for SNA/CCS terminals.

VM/XA SF: VM/XA SP does not support the operands MSG, WNG, EMSG, IMSG, and LINEDIT, which are supported in VM/XA SF Release 1.

TERMINAL APL

Privilege Class: G

Use **TERMINAL APL** to control the use of the APL character translation tables.

TERMinal	APL { ON } { OFF }
----------	-----------------------

where:

APL { ON }
{ OFF }

controls the use of the APL character translation tables. If you specify **ON**, CP will use the translation tables applicable to display stations equipped with the APL hardware feature. If you specify **OFF**, CP will use the normal terminal translation tables. Unless otherwise specified, **APL OFF** is in effect.

Usage Notes

1. If you set **APL ON**, **TERMINAL TEXT** is forced **OFF**.
2. For more information, refer to the “General Usage Notes for **TERMINAL**.”

Responses

None.

TERMINAL ASCIITBL

Privilege Class: G

Use the **TERMINAL ASCIITBL** command to select the translation table used by your TTY terminal.

TERMinal	ASCiitbl { VM1 } { VM2 }
----------	-----------------------------

where:

ASCIITBL { VM1 }
{ VM2 }

specifies the translation table to be used. VM1 specifies the use of the translation table based on the TTY ANSI X3.4 1977 Standard. VM2 specifies the use of the translation table based on the TTY ANSI X3.26 1980 Standard.

Usage Notes

TERMINAL ASCIITBL may be issued only from a TTY device; otherwise an error message is issued.

Responses

None.

TERMINAL ATTN

Privilege Class: G

Use **TERMINAL ATTN** to control the signaling of an attention interruption on keyboards and printers of ASCII terminals. This command is not valid for display type terminals.

TERMinal	ATtn	{ ON } { OFF }
----------	------	-------------------

where:

ATTN { **ON** }
{ **OFF** }

controls the signaling of an attention interruption. **ON** specifies that an exclamation point (!) is displayed when an attention interruption occurs. The **OFF** option suppresses the displaying of the exclamation point and carriage return.

OFF is the default for display terminals. **ON** is the default for ASCII devices.

Usage Notes

TERMINAL ATTN ON controls the signaling of an attention interruption only on the keyboards and printers or ASCII terminals. It is not valid for display type terminals.

Responses

None.

TERMINAL BREAKIN

Privilege Class: G

Use **TERMINAL BREAKIN** to allow the user to control break-in by CP while in full-screen mode.

TERMinal	BREakin	{ IMmed } { GUESTctl }
----------	---------	---------------------------

where:

IMmed

specifies that break-in by CP can take place on any full screen reads and full screen writes that have an inactive keyboard, as well as when the user hits the **BRKKEY**.

GUESTctl

specifies that high-priority warnings from CP and user-requested CP functions still break in, but messages only break in when the user hits the **BRKKEY**, or one of the previously mentioned tasks break in.

Usage Notes

1. For the **BREAKIN** option, only **IMMED** is supported for SNA/CCS terminals. If you issue **TERMINAL BREAKIN** with the **GUESTCTL** option from an SNA/CCS terminal, the command is not executed and you receive a message indicating an invalid device type error.
2. If you change the **TERMINAL BREAKIN** setting while logged on to a local display and then disconnect and reconnect to an SNA/CCS terminal, your **TERMINAL BREAKIN** setting is affected in the following way during the reconnection processing:
 - **BREAKIN**, if previously set to **IMMED**, is not changed.
 - **BREAKIN**, if not previously set to **IMMED**, is set to **IMMED** and a message is issued to inform you of the change.

Responses

None.

TERMINAL BRKKEY

Privilege Class: G

Use **TERMINAL BRKKEY** to define the key to control break-in by CP for full-screen guests.

TERMinal	BRKkey { key } { NONE }
----------	----------------------------

where:

key

must be one of the following: PA1, PA2, PFnn, or CLEAR. PFnn can be any PF key on your display station, from PF1 to PF24.

NONE

specifies that no **BRKKEY** should be defined.

Usage Notes

1. The break key is in effect only in a full-screen environment.
2. When you first log on to a virtual machine, the default key to control break-in by CP for full-screen guests is PA1. However, if you intend to run an operating system in your virtual machine that requires the PA1 key for some other function, you can use this command to define an alternate break key.
3. Once you've issued this command, the new break key remains in effect for the duration of the session or until you reissue this command during the current session. If you've used this command to define a new break key and then, at some point in the session, would prefer to use the default break key again, the PA1 option allows you to do so.
4. If you use this command to define a key other than PA1 as the break key, VM/SP passes control of the PA1 key to your virtual machine.
5. When NONE is specified, the full-screen guest has complete control of the keyboard. You can return to CP mode only if the full-screen program provides a way for you.
6. Entry to the protected application environment disables the **BRKKEY**. To reenable it within the protected application, issue the **TERMINAL BRKKEY** command. Upon exit from the protected application, CP restores the break key setting to what it was before entry into the same protected application. If the break key was changed while in the protected application, it remains unchanged when you exit the application.
7. For the **BRKKEY** option, only PA1 and NONE are supported for SNA/CCS terminals. If you issue **TERMINAL BRKKEY** with any other options from an SNA/CCS terminal, the command is not executed and you receive a message that indicates an invalid device type error.

TERMINAL BRKKEY

- |
- |
- |
- |
- |
- |
- |
- |
- |
- |
8. If you change the **TERMINAL BRKKEY** setting while logged on to a local display, then disconnect and reconnect to an SNA/CCS terminal, your **TERMINAL BRKKEY** setting is affected in the following way during the reconnection processing:
- If **BRKKEY** was previously set to **NONE** or **PA1**, it is not changed.
 - Otherwise, **BRKKEY** is set to **PA1** and an informational message is issued to inform you of the change.

Responses

None.

TERMINAL CHARDEL

Privilege Class: G

Use TERMINAL CHARDEL to define the logical character delete symbol.

TERMinal	Charde1 { ON } { OFF } { char }
----------	---------------------------------------

where:

CHARDEL { ON }
 { OFF }
 { char }

defines the logical character delete function for your virtual machine. If you specify ON, the system default symbol becomes your logical character delete symbol. The default symbol is @, but may be changed by what is specified in your entry in the system directory. If you specify char, that character becomes your logical character delete symbol. The character you select should be unique and not common to the data stream being entered. If you specify OFF, no logical character delete symbol is used for your virtual machine.

Usage Notes

1. When you log on, CHARDEL ON is in effect. If your virtual machine is logged on by someone else using the CP AUTOLOG or XAUTOLOG command, CHARDEL OFF is the default.
2. For more information, see the "General Usage Notes for TERMINAL."

Responses

None.

TERMINAL CONMODE

Privilege Class: G

Use **TERMINAL CONMODE** to alter the console mode of your display. The console mode (3215 or 3270) determines the type of virtual I/O support that VM/SP provides.

TERMinal	CONmode { 3215 } { 3270 }
----------	------------------------------

where:

CONMODE { 3215 }
{ 3270 }

defines the type of virtual I/O support that VM/SP provides for your display. If you specify 3215, then 3215 CCW processing occurs. If you specify 3270, then 3270 CCW processing occurs.

Usage Notes

1. In order to use this command correctly, you need to be aware of the three factors that affect your console mode:
 - a. **The CONSOLE statement:** The CONSOLE statement in your system directory determines the initial console mode of your display.
 - b. **The TERMINAL CONMODE command:** You can issue the TERMINAL CONMODE command to change your console mode.
 - c. **The DEFINE CONSOLE command:** You can also issue the DEFINE CONSOLE command to change your console mode.

In brief, you have two commands that you can use to actively change your console mode: **TERMINAL CONMODE** and **DEFINE CONSOLE**.

2. In order to change your console mode setting with **DEFINE CONSOLE**, you first have to detach your console with the **DETACH** command. For various reasons, you may not always want to detach your virtual console. **TERMINAL CONMODE** provides you with a way of changing your console mode setting without detaching your console.
3. Certain programs require a console mode that may differ from your usual or default console mode. If you attempt to run such a program with the incorrect console mode, results are unpredictable. In most instances, the program terminates abnormally.

Some programs reset the console mode internally. You may not be aware that the console mode has been changed for you automatically. Problems result if you then invoke a program that requires the alternate console mode setting.

Therefore, you should remain aware of the significance of the console mode any time that you invoke a program in your virtual machine.

4. If you issue **TERMINAL CONMODE 3270** from an SNA/CCS terminal controlled by a VTAM service machine that does not support **CONMODE 3270**, the command is not executed and you receive a message indicating an invalid device type error. **CONMODE 3270** is supported by ACF/VTAM Version 3, Release 1.1 for VM/SP (VM/VTAM) and later releases.

5. Use the QUERY VIRTUAL CONSOLE command to determine the console mode.

Responses

None.

TERMINAL CNTL

Privilege Class: G

Use the **TERMINAL CNTL** command to allow either CP or an application program to insert TTY line control characters into output directed to your TTY terminal.

TERMinal	CNTL {SYS} {USR}
----------	---------------------

where:

**CNTL {SYS}
{USR}**

determines whether CP or the application is to insert the TTY line control characters. **SYS** specifies that CP is to insert TTY line control characters into output directed to a TTY terminal. **USR** specifies that the application program is to insert TTY line control characters into output directed to a TTY terminal.

Usage Notes

1. **TERMINAL CNTL** may be issued from a TTY device only; otherwise an error message is issued.
2. If you specify **USR**, both **TERMINAL TYPE** and **TERMINAL PROMPT** must be set to **TTY**. If **TYPE** and **PROMPT** are not **TTY** and you select **USR**, **TYPE** and **PROMPT** are forced to **TTY**. These options may not be changed while **CNTL** is set to **USR** or an error message is issued and the setting remains as **TTY**.
3. Application programs that use the **CMS WRTERM** macro to write data to a terminal must specify either the **WRTERM EDIT=NO** or **EDIT=LONG** parameters. Otherwise, **TERMINAL CNTL** has no effect.
4. If you disconnect your virtual machine, **TERMINAL CNTL** is reset to **SYS**.

Responses

None.

TERMINAL ESCAPE

Privilege Class: G

Use TERMINAL ESCAPE to define the logical escape character.

TERMinal	EScape { ON OFF char }
----------	------------------------------

where:

ESCAPE { ON
OFF
char }

defines the logical escape character function. If you specify ON, the default symbol becomes the logical escape character for your virtual machine. The default symbol is “, but may be changed by what is specified in your entry in the system directory. If you specify char, then that character will become the logical escape character. The character you select should be unique and not common to the data stream being entered. If you specify OFF, no logical escape character is allowed for your virtual machine.

Usage Notes

1. When you log on, ESCAPE ON is in effect. If your virtual machine is logged on by someone else using the CP AUTOLOG or XAUTOLOG command, ESCAPE OFF is the default.
2. For more information, refer to “General Usage Notes for The TERMINAL Command” on page 631.

Responses

None.

TERMINAL HILIGHT

Privilege Class: G

Use **TERMINAL HILIGHT** to highlight the command redisplay line on the virtual machine console.

TERMinal	HILight { ON } { OFF }
----------	---------------------------

where:

HILIGHT { ON }
 { OFF }

defines whether or not the command redisplay line on a display is highlighted. When the option is OFF, user input is displayed at normal intensity. The HILIGHT option is set to OFF when you logon. The **TERMINAL HILIGHT** command has no effect on the highlighting options specified on the CP **SCREEN** command.

Usage Notes

None.

Responses

None.

TERMINAL LINEDEL

Privilege Class: G

Use TERMINAL LINEDEL to define the logical line delete symbol.

TERMinal	LINEDEL { ON } { OFF } { char }
----------	---------------------------------------

where:

LINEDEL { ON }
 { OFF }
 { char }

defines the logical line delete function for your virtual machine. If you specify ON, the system default symbol becomes the logical line delete symbol for your virtual machine. The default symbol is ϕ , but may be changed by what is specified in your entry in the system directory. If you specify char, then that character will become the logical line delete symbol. The character you select should be unique and not common to the data stream being entered. If you specify OFF, no logical line delete symbol is allowed for your virtual machine.

Note: For ASCII devices the default character is a left bracket ([).

Usage Notes

1. If you disconnect your virtual machine after you change the LINEDEL character, the new LINEDEL character remains in effect while you are disconnected and after you reconnect.
2. When you log on, LINEDEL ON is in effect. If your virtual machine is logged on by the someone else using the CP AUTOLOG or XAUTOLOG command, LINEDEL OFF is the default.
3. For more information, refer to "General Usage Notes for The TERMINAL Command" on page 631.

Responses

None.

TERMINAL LINEND

Privilege Class: G

Use **TERMINAL LINEND** to define the logical line end symbol for your virtual machine.

TERMinal	LINEND { ON } { OFF } { char }
----------	--------------------------------------

where:

LINEND { ON }
{ OFF }
{ char }

defines the logical line end symbol for your virtual machine. If you specify **ON**, the system default symbol becomes the logical line end symbol for your virtual machine. The default symbol is #, but may be changed by what is specified in your entry in the system directory. If you specify **char**, then that character becomes the logical line end character. The character you select should be unique and not common to the data stream being entered. If you specify **OFF**, no logical line end symbol is allowed for your virtual machine.

Usage Notes

1. When you log on, **LINEND ON** is in effect. If your virtual machine is logged on by someone else using the **CP AUTOLOG** or **XAUTOLOG** command, **LINEND OFF** is the default.
2. For more information, refer to "General Usage Notes for The **TERMINAL Command**" on page 631.

Responses

None.

TERMINAL LINESIZE

Privilege Class: G

Use **TERMINAL LINESIZE** to specify the maximum allowable line length for display screen output.

TERMiNa1	LiNESize { nnn } { OFF }
----------	-----------------------------

where:

LINESIZE { nnn }
{ OFF }

specifies the maximum allowable line length. The value you specify for nnn can be any number from 1 through 255. OFF specifies that the output line length is not to be adjusted by CP. In other words, CP does not split a long output line into two or more shorter ones.

If neither nnn nor OFF is specified, the default value is device dependent.

Usage Notes

1. If **TERMINAL APL ON** is specified, CP defines an output length of 1760 and assumes that the APL system has inserted the appropriate carriage control characters into the output data.
2. If a user disconnects from the system and then logs on again, the **LINESIZE** specified is reset to the system defaults.

Responses

None.

TERMINAL MODE

Privilege Class: G

Use **TERMINAL MODE** to control the display station attention environment.

TERMinal	MODE { CP } { VM }
----------	-----------------------

where:

MODE { CP }
{ VM }

controls the display station attention environment. If you specify **CP**, one or more attentions will force your virtual machine into the **CP** environment. If you specify **VM**, one or more attentions will be reflected to your virtual machine. **VM** is the default for all users except the primary system operator when the user logs on.

Usage Notes

1. This command only has meaning in line mode.
2. For more information, refer to “General Usage Notes for The **TERMINAL** Command” on page 631.

Responses

None.

TERMINAL PROMPT

Privilege Class: G

Use **TERMINAL PROMPT** to set the prompting sequence to be used on a TTY device.

TERMinal	PROMpt { VM } { TTY }
----------	--------------------------

where:

PROMPT { VM }
{ TTY }

sets the prompting sequence. The VM option causes the cursor to be positioned in column 1 of the input line for a read operation. The TTY option enables the normal TTY prompt sequence of a period in column 1 and the cursor in column 2.

Usage Notes

1. **TERMINAL PROMPT** can be issued from a TTY device only; otherwise, an error message is issued.
2. If a user disconnects from the system and then logs on again, the **PROMPT** options are reset to the system defaults.

Responses

None.

TERMINAL SCROLL

Privilege Class: G

Use **TERMINAL SCROLL** to specify the number of scrolled lines on a 3101 display screen.

TERMinal	SCROLL { nnn } { CONT }
----------	----------------------------

where:

SCROLL { nnn }
{ CONT }

controls scrolling. *nnn* specifies the number of lines that are scrolled up before scrolling stops with on the bottom line of the screen. Pressing the **NEW LINE** key (character mode) or the **SEND** key (block mode) causes another cycle of *nnn* lines to be scrolled. For normal frame by frame scrolling, make *nnn* equal to the number of data lines on the screen. **CONT** specifies continuous scrolling to the end of the output.

Usage Notes

1. **TERMINAL SCROLL** may be issued from a TTY device only; otherwise, an error message is issued.
2. If a user disconnects from the system and then logs on again, the **SCROLL** options are reset to the system default.

Responses

None.

TERMINAL TABCHAR

Privilege Class: G

Use **TERMINAL TABCHAR** to define logical tab characters on display devices. The **QUERY TERMINAL** command displays the current logical tab character.

TERMinal	TABchar { ON OFF char }
----------	-------------------------------

where:

TABCHAR { ON
OFF
char }

defines logical tab characters. ON sets the logical tab character to the one used by the system, which is a broken vertical bar. OFF resets the logical tab character to undefined. char defines the logical tab character to be used by the system internally. It may be any valid specific graphic character (that is, a character other than A-Z or 0-9).

Usage Notes

1. When an invalid character is specified, an error message is issued to the display and the logical tab character remains unchanged.
2. If a user disconnects from the system and then logs on again, the **TABCHAR** specified from the previous session will be in effect.
3. For more information, refer to the “General Usage Notes for **TERMINAL**.”

Responses

None.

TERMINAL TEXT

Privilege Class: G

Use TERMINAL TEXT to control the use of text translation tables.

TERMiNa1	TEXT { ON } { OFF }
----------	------------------------

where:

TEXT { ON }
 { OFF }

controls the use of the text translation tables. If you specify ON, CP will use the translation tables applicable to display terminals equipped with the text hardware features. If you specify OFF, CP will use the normal translation tables. Unless otherwise specified, TEXT OFF is in effect.

Usage Notes

1. If TEXT is set ON, TERMINAL APL is forced OFF.
2. For more information, refer to the "General Usage Notes for TERMINAL."

Responses

None.

TERMINAL TYPE

Privilege Class: G

Use **TERMINAL TYPE** to define a device as a 3101 or as a TTY.

TERMinal	TYpe { 3101 } { TTY }
----------	--------------------------

where:

TYPE { 3101 }
{ TTY }

defines the device. When 3101 is in effect, the terminal is regarded as a display screen and keyboard ASCII device. The 3101 operand (a) allows the use of program function keys 1 through 24, (b) causes output to begin at column 1 of line 24 on the display, and (c) allows editing of previously entered information without the necessity of rekeying the entire text. When TTY is in effect, the terminal is regarded as a typewriter terminal.

Usage Notes

1. **TERMINAL TYPE** may be issued from a TTY device only; otherwise, an error message is issued.
2. If a user disconnects from the system and then logs on again, the **TYPE** options are reset to the system defaults.

Responses

None.

TRACE

Privilege Class: G

Use the TRACE command described in this section to monitor events that occur in your virtual machine. VM/XA SP allows you to trace a number of events, including:

- Instruction execution
- Storage alteration
- Register alteration
- I/O activity.

Each traced event results in a *trace entry*, a command response that you can have sent to your virtual console, to a virtual printer, or to both. The trace entry is made up of significant information about the event. You can use trace entries to analyze the operation of your virtual machine and to debug problems.

Note that in this manual the TRACE documentation is primarily for reference. For a full explanation of *how* to use the trace facility, along with examples, see *VM/XA SP Virtual Machine Operation*.

How the TRACE Documentation Is Organized

Although it has a large number of operands and options, the TRACE command is relatively easy to use. This section is organized to make using TRACE as convenient as possible for you. There are two major parts: the overview material that concerns the entire trace facility, followed by subsections describing individual TRACE operands.

Following this introduction, the overview provides three format boxes showing all the possible operands and options available with the TRACE command. Once you become somewhat familiar with the TRACE command, you can use these format boxes to view at a glance all of the options at your disposal. The TRACE operands are divided into two types: primary and control operands. The primary operands allow you to choose the kind of event you want to trace. These operands are shown in the first format box, along with any options that are unique to a particular primary operand. The second format box shows the options that are common to all of the primary operands as well as to the TRAP control operand. The third format box shows the control operands (the operands that enable you to operate the trace facility) along with any options unique to a particular control operand.

The format boxes are followed by a brief explanation of what each contains. This section includes lists of all the operands and gives you the page on which you can find an operand's complete description. (Each operand is described in its own subsection, as though it were a separate command. Note that certain related operands are described together in a single subsection.)

Next, there are "General Usage Notes for TRACE" on page 657. These notes provide you with a capsule explanation of how to use the trace facility. Again, for a full explanation of how to use the trace facility, see *VM/XA SP Virtual Machine Operation*. The usage notes also include some important information about how the trace facility functions.

The next section, “TRACE: Common Options” on page 659, provides full explanations for the options common to all of the primary operands and to the TRAP control operand. This is followed by “TRACE: Command Responses” on page 663, showing all of the command responses (trace entries) that you may receive. Note that this response section provides a list of all of the response symbols and an explanation for each. During the tracing process, you can turn to this section to interpret the trace entries that you receive. Because all of the responses are gathered here, this section can be especially helpful when you’re tracing a number of varied events simultaneously.

The overview material is followed by the subsections that document individual TRACE operands. These subsections are in alphabetical order. Each subsection explains the particular use of an operand (or a group of related operands), and provides explanations for any options unique to that operand as well as usage notes unique to that operand. If the operand is compatible with the common options, the subsection also contains a list of the common options.

Complete Format for TRACE

The following three format boxes show all of the operands and options available with the TRACE command.

Primary operand format:

TRace	ALL BRAnch [INTO range] COunt DIAGnose [range] [SIM NOSIM] EXTernal [range] G[n[-m][,...]] [DATA range MASK string] INStRuction [DATA string] I/O [range] [INStRuction] [CCW NOCCW] [SIM NOSIM] [INTerrupt] MC [range] [SIM NOSIM] MCH mnemonic1 [SIM NOSIM] mnemonic2 [range] [SIM NOSIM]
TRace	PROGram [range] RIO [range] [SIM NOSIM] [CCW NOCCW] RSCH SIO SIOF SSCH STore [INTO { range hexLoc [DATA string [MASK string] }] SVC [range] [SIM NOSIM] TAbLe TPI [range] [SIM NOSIM]

TRACE

Common options:

<pre> [{PSWA } range] [RUN] {FRom } {Range} [NORun] </pre>
<pre> [SKIP n] [PAss n] [STOP n] [STEp n] </pre>
<pre> [TERminal] [Printer] [SUPervisor] [DAT] NOTERminal] [NOPRINTer] [PROBlEm] [NODAT] B0th </pre>
<pre> [IDentifer ident] </pre>
<pre> [CMd ... cp command ...] NOCMD </pre>

Control operand format:

TRace	<pre> APPEND name CALL name [SUSpend] CLEAR [name] DELeTe ident END [ALL] [C0unt] G0to name [SUSpend] RETRun name [SUSpend] TRAP ident [common options] </pre>
-------	--

where:

primary operand

is one of the operands you use to specify the kind of event you want to trace. (On the pages that follow, the operands are described in their own subsections, as though they were separate commands. Certain related operands are described together in single subsections. For your convenience, each operand is listed below alphabetically, along with a reference to the subsection and the page where its full description appears.) The primary operands are:

Operand	Description
ALL	See "TRACE ALL" on page 672.
BRANCH	See "TRACE BRANCH" on page 674.
COUNT	See "TRACE COUNT" on page 677.
DIAGNOSE	See "TRACE SVC/DIAGNOSE/MC" on page 697.
EXTERNAL	See "TRACE EXTERNAL/PROGRAM" on page 679.
GPRs	See "TRACE GPRs" on page 682.
INSTRUCTION	See "TRACE INSTRUCTION" on page 684.
I/O	See "TRACE I/O" on page 686.
MC	See "TRACE SVC/DIAGNOSE/MC" on page 697.

Operand	Description
MCH	See “TRACE MCH” on page 688.
mnemonic1	See “TRACE mnemonic1” on page 689.
mnemonic2	See “TRACE mnemonic2” on page 691.
PROGRAM	See “TRACE EXTERNAL/PROGRAM” on page 679.
RIO	See “TRACE RIO/RSCH/SIO/SIOF/SSCH” on page 693.
RSCH	See “TRACE RIO/RSCH/SIO/SIOF/SSCH” on page 693.
SIO	See “TRACE RIO/RSCH/SIO/SIOF/SSCH” on page 693.
SIOF	See “TRACE RIO/RSCH/SIO/SIOF/SSCH” on page 693.
SSCH	See “TRACE RIO/RSCH/SIO/SIOF/SSCH” on page 693.
STore	See “TRACE STORE” on page 695.
SVC	See “TRACE SVC/DIAGNOSE/MC” on page 697.
Table	See “TRACE TABLE” on page 699.
TPI	See “TRACE TPI” on page 700.

common options

are the options that are available to all the primary operands as well as to the TRAP control operand. These options are fully explained in “TRACE: Common Options” on page 659.

control operand

is one of the operands you use in controlling the operation of the trace facility itself. (On the pages that follow, the operands are described in their own subsections, as though they were separate commands. Certain related operands are described together in single subsections.)

For your convenience, each operand is listed below alphabetically, along with a reference to the subsection and the page where its full description appears.) The control operands are:

Operand	Description
APPEND	See “TRACE APPEND” on page 673.
CALL	See “TRACE GOTO/CALL/RETURN” on page 680.
CLEAR	See “TRACE CLEAR” on page 676.
DELeTe	See “TRACE TRAP” on page 702.
END	See “TRACE END” on page 678.
GOto	See “TRACE GOTO/CALL/RETURN” on page 680.
RETurn	See “TRACE GOTO/CALL/RETURN” on page 680.
TRAP	See “TRACE TRAP” on page 702.

General Usage Notes for TRACE

1. All of the primary TRACE operands establish trace traps. A “trace trap” is a programming function that, as its name implies, captures information about an event in your virtual machine.

When you establish a trap, you can assign it any unique one-to-four character name (with the exception of the name “ALL”). If you do not assign a name, the

TRACE

system assigns the trap a unique number from 1 to 9999. You use this unique identifier (name or number) to alter a trap with the TRACE TRAP command or to delete a trap with the TRACE DELETE command.

The options (unique and common) available with each primary operand and with the TRAP operand allow you to specify exactly how you want the trap to execute. For example, you might want the trap to function only when an event occurs within a particular address range. In this case, you would specify the common option FROM with the address range you've chosen. At the same time, you might want to trace events only when your virtual machine is running in supervisor state. In this case, you would also specify the common option SUPERVISOR.

2. All trace traps must be contained in units of control called sets. A "trace set" is a collection of one or more traps that run together.

You create a set using the TRACE GOTO command. When you create the set, you assign it a unique 1- to 8-character name. If you do not assign the set a name, the system assigns a unique name for you.

If you do not create a set before you establish a trap, VM/XA SP creates a set for you named INITIAL.

Only one set may be active at any given time.

The following commands exist for manipulating sets:

- TRACE GOTO
- TRACE CALL
- TRACE RETURN
- TRACE CLEAR
- TRACE END
- TRACE APPEND

3. You can define up to 255 trace sets and trace traps in any combination during a single trace session. (For example, you can create two trace sets with no more than 253 traps between them, five trace sets with no more than 250 traps between them, 20 trace sets with no more than 235 traps between them, and so on.)
4. If you have RUN set ON when you issue a TRACE command, making a TRACE set active, RUN will be turned off.
5. For information on how to display your current trace settings, see the QUERY TRACE command. The QUERY TRACE can be especially useful if you need to know the identifiers that VM/XA SP has assigned to your traps or sets. The command QUERY TRACE ALL displays the attributes of all trace traps within all trace sets.
6. The IPL command for VM/XA SP suspends all user tracing with the TRACE command during the IPL process. Tracing is then automatically reset as it was and continues normally.
7. DATOFF is a synonym for NODAT. DATOFF is allowed whenever NODAT is allowed.
8. Depending on the model, PER events for general-register alteration may or may not be recognized for vector instructions.

TRACE: Common Options

[PSWA range] FROM Range]	[RUN NORun]		
[SKIP n] [PAss n] [STOP n] [STEp n]			
[TErminAl NOTERMinAl BOth]	[PrInter NOPrInTEr]	[SUPErvisor PROBLEm]	[DAT NODAT]
[IDentifier ident]			
[CMd ... cp command ...] NOCMD]			

You can respecify all of the common options except the ID option. If you specify any of these options without a primary trace operand, they will be applied to all events in the current trace set. If you specify the TRAP operand, these operands will be applied to a single event.

where:

PSWA range
FROM range
RANGE range

PSWA restricts instruction tracing to a given address range. PSWA permits a trap to be taken if the PSWA (that is, the program status word address, commonly known as the instruction counter) points to an instruction in the specified range. The range may be a single address, a pair of addresses separated by a “-,” or an address followed by a period followed by a byte count. An instruction is considered to be within the range if the first byte is within the range.

FROM and RANGE are synonyms for PSWA used in those subcommands where a potential for confusion exists. For example, TRACE BRANCH allows two instruction ranges to be specified, a FROM range and an INTO range. FROM and RANGE are allowed wherever PSWA is allowed.

For executed instructions (that is, instructions that are issued by means of the assembler language EXECUTE instruction), the instruction is within the range if either:

- The first byte of the EXECUTE instruction is within the range
- The first byte of the target instruction is within the range

For external interruptions, the PSW address at the time of the interruption is inspected to see if it is within the range.

When a program event recording (PER) event is reflected to your virtual machine, the PER event address is checked.

For any other program exception, the old PSW instruction address is inspected if a nullification exception occurs; for all others, the old PSW instruction address is adjusted by the instruction length code, and then is inspected.

RUN

specifies that virtual machine execution is not to pause when the specified event occurs. **RUN** is assumed if **TERMINAL** is not specified or assumed. **RUN** is incompatible with **NORUN**, **STEP**, and **STOP**.

NORUN

specifies that virtual machine execution is to pause when the specified event occurs. **NORUN** is assumed if **TERMINAL** is specified or assumed. **NORUN** is incompatible with **RUN**, **STEP**, and **STOP**.

SKIP n

requests that the associated event be completely ignored *n* times. When used with the **NOSIM** option, however, it specifies the number of times that an instruction will be simulated normally.

STOP n

requests a number of displays for the command. Your virtual machine pauses after the specified number of displays. Unless a **STEP** count is specified, a pause occurs for each subsequent display.

PASS n

is a repeating **SKIP** specification. If you do not specify a **SKIP** option, **PASS** takes effect immediately. If you do specify a **SKIP** count, **PASS** takes effect after the **SKIP** count expires and the specified event has been recognized.

For example:

- **PASS 2** will display every third event
- **PASS 8** will display every ninth event
- **SKIP 2 PASS 8** will display the third event, the 12th event, and every ninth event thereafter.

STEP n

is a repeating **STOP** specification. If you do not specify a **STOP** option, **STEP** takes effect immediately. If you do specify a **STOP** count, **STEP** takes effect after the first pause.

For example:

- **STEP 3** will pause every third event
- **STEP 9** will pause every ninth event
- **STOP 3 STEP 9** will pause after the third and 12th displays and for every ninth display thereafter.

TERMINAL

specifies that output for the current command is to be sent to the virtual console. This is the default setting for newly-created tracing specifications with neither terminal nor printer output specifications. **TERMINAL** and **NOTERMINAL** are incompatible options.

NOTERMINAL

specifies that output for the current command is not to be sent to the virtual console. **TERMINAL** and **NOTERMINAL** are incompatible options.

BOTH

specifies that the output for the current command is to be sent to the virtual console and to the virtual printer.

PRINTER**PRT**

specifies that output for the current command is to be sent to the virtual printer having the lowest virtual device address. **PRINTER** may be abbreviated as **PRT**. **PRINTER** and **NOPRINTER** are incompatible options.

NOPRINTER**NOPRT**

specifies that output for the current command is not to be sent to the virtual printer. **NOPRINTER** may be abbreviated as **NOPRT**. **PRINTER** and **NOPRINTER** are incompatible options.

SUPERVISOR

requests that the specified tracing be restricted to events occurring while the machine is in supervisor state. If specified in addition to **PROBLEM**, or if neither **SUPERVISOR** nor **PROBLEM** is specified, then both problem state and supervisor state events will be traced.

PROBLEM

requests that the specified tracing be restricted to events occurring while the machine is in problem state. If specified in addition to **SUPERVISOR**, or if neither **SUPERVISOR** nor **PROBLEM** is specified, then both problem state and supervisor state events will be traced.

DAT

requests that the specified tracing be restricted to events occurring while the virtual machine has virtual dynamic address translation enabled. If you specify **DAT** in addition to **NODAT**, or if you specify neither **DAT** nor **NODAT**, events occurring in both **DAT** and non-**DAT** mode will be traced.

NODAT

requests that the specified tracing be restricted to events occurring while the virtual machine has virtual dynamic address translation disabled. If you specify **NODAT** in addition to **DAT**, or if you specify neither **DAT** nor **NODAT**, events occurring in both **DAT** and non-**DAT** mode will be traced.

IDENTIFIER ident

specifies an identifier for a trace trap describing the event. It must be one to four characters in length. If a trap already exists with the given identifier, it will be replaced by the trap for the newly defined event. If you do not specify **ID**, a unique number from 1 to 9999 will be assigned as the identifier, with leading zeros removed. You cannot use the word "ALL" as an identifier. Also, on the **TRACE TRAP** commands (**TRACE TRAP** and **TRACE DELETE**), this is an invalid option.

CMD... cp command ...

specifies a CP command or set of CP commands to be executed when the specified event occurs.

Multiple commands must be separated by linend characters. Actual linend characters will terminate the trace command.

For example, if the pound sign (#) is your logical line end character, and if " is your logical escape character:

```
trace svc cmd d 20#d 30
```

issues a **TRACE SVC** command that displays location 20, and then issues the CP command **DISPLAY** to display location 30.

TRACE: Common Options

The command:

```
trace svc cmd d 20"#d 30
```

issues one TRACE command; it displays locations 20 and 30 whenever an SVC instruction is executed.

You can use this option with TRACE GOTO, TRACE CALL, and TRACE RETURN to allow conditional and unconditional branching and subroutine calls between TRACE environments.

All characters between the CMD option and the end of the TRACE command line are considered to be part of the CMD field; thus, it must be the last option on the line. No command syntax verification is made on the set of CP commands to be executed until the specified event occurs. Terminal output from the set of commands is not affected by the TERM or NOTERM specification of the trace event.

NOCMD

specifies that no CP command is to be executed when the specified event occurs.

Use NOCMD to delete CMD fields from existing trace events.

TRACE: Command Responses

The following pages illustrate the command responses for each of the TRACE command operands. These response descriptions use the following symbols:

Symbol	Meaning
aaaaaaaa	Storage address, followed by a single quote (') if DAT. For external and I/O interruptions it is the PSW address at the time of interruption. When a PER event is reflected to the virtual machine, it is the program event recorder event address. For other program exceptions, it is the old PSW instruction address for nullification exceptions, and for all others it is the old PSW instruction address as adjusted by the instruction length code.
ttttttt	Transfer address, followed by a single quote (') if DAT
xxxxxxx	Instruction, CSW status, et al
ggggggg	Second level channel command word, CSW status, et al
hhhhhhh	First level channel command word, CSW status, et al
oooo	First level CCW offset
address1	Address of operand 1 (for SI and SS formats only)
address2	Address of operand 2 (for RX, RS, S, and SS formats only)
	Note: address1 and address2 may be question marks if the addresses cannot be resolved because the base or index registers used to compute them have been altered by the instruction.
content1	Contents of R1 field (RRE format only)
content2	Contents of R2 field (RRE format only)
vdev	Virtual device address
vsch	Virtual subchannel number
mnem	Mnemonic for instruction
code	Interruption code number in hexadecimal
xx	Extension-exception code
nnnnnnnn	The number representing the successful trace events in the current trace set
name	Interruption name (e.g., operation, addressing)
CC n	Condition code number (0, 1, 2 or 3)
Gnn	A general register with altered contents
ccccccc	Contents of register
ss	Status byte
FSA = nnnnnnn	The FAILING STORAGE ADDRESS if valid. If it is invalid, then nothing will be shown.

Symbol	Meaning
zz	Altered byte of an EXECUTE instruction, formed by inclusive-or'ing the low byte of R1 with the second byte of the executed instruction.
***	Prefix indicating virtual machine interruption
->	As a prefix indicating transfer of control, it is present at the beginning of a line whenever the displayed instruction does not immediately follow the last instruction displayed. As a suffix for a successful branch instruction, it precedes the address being transferred to.
> >	Indication of storage alteration, followed by the operand address.
=	Prefix used with LOAD ADDRESS to show that the actual data is shown rather than computing the operand address by base and displacement.

For operation codes that represent two mnemonics, the mnemonic for fixed-point operations is shown. The following is a list of those operation codes with the mnemonic to be returned by CP:

Number	Code	Meaning
A409	VL	Load
A40A	VLM	Load Matched
A40B	VLX	Load Expanded
A40D	VST	Store
A40E	VSTM	Store Matched
A40F	VSTK	Store Compressed
A40A	VLMR	Load Matched
A50B	VLZR	Load Zero
E400	VLI	Load Indirect
E401	VSTI	Store Indirect

Basic Instruction Display

aaaaaaaa mnem xxxxxxxx address1 address2 CC n

Executed Instructions

aaaaaaaa EX xxxxxxxx address1 mnem xxxxxxxx address1 address2 CC n

Load Address Instructions

aaaaaaaa LA xxxxxxxx = ccccccc CC n

RRE Format Instructions:

aaaaaaaa mnem xxxxxxxx content1 content2 CC n

Instructions Altering Storage

aaaaaaaa mnem xxxxxxxx >> address1 address2 CC n

Successful Branch Instructions (including SVC and LPSW)

aaaaaaaa mnem xxxxxxxx -> tttttttt CC n

Instructions Satisfying GPR Alteration Tracing

aaaaaaaa mnem xxxxxxxx address1 address2 CC n Gnn=cccccccc
Gnn=cccccccc Gnn=cccccccc Gnn=cccccccc ...

External Interruptions

*** aaaaaaaa EXT code - tttttttt

Program Interruptions

(Instruction displayed where possible)

*** aaaaaaaa PROG code - tttttttt name

Virtual Machine Check Interruptions

(Instruction displayed where possible)

*** aaaaaaaa MCH -> tttttttt type type ...
(FSA=nnnnnnnn)

where:

aaaaaaaa is the address contained in the second word of the old PSW, if valid, of the machine check. If the PSW instruction address is marked invalid, question marks will appear in place of the address.

ttttttt is the address contained in the second word of the new PSW for the machine check.

type is one or more of the machine check codes found in the *370-XA: Principles of Operation*. The major virtual machine check codes are:

Number	Code	Meaning
0	SD	System Damage
1	PD	Instruction-processing Damage
2	SR	System Recovery
3	TD	Interval-timer Damage
4	CD	Timing Facility Damage
5	ED	External Damage
6	VF	Vector Facility Failure
7	DG	Degradation
8	W	Warning
9	CP	Channel Report Pending

Table 10 (Page 2 of 2). Machine Check Codes		
Number	Code	Meaning
10	SP	Service Processor Damage
11	CK	Channel Subsystem Damage
12	Unused	
13	VS	Vector Facility Source
14	B	Backed Up

FSA = nnnnnnnn is the FAILING STORAGE ADDRESS if valid. If it is invalid, then nothing will be shown.

370 I/O Interruptions

*** aaaaaaaa I/O vdev -> tttttttt CSW xxxxxxxx xxxxxxxx

370-XA I/O interruptions

*** aaaaaaaa I/O DEV vdev -> tttttttt SCH vsch PARM xxxxxxxx

370 I/O Instructions

aaaaaaa SIO xxxxxxxx vdev CC 0 CAW xxxxxxxx
 aaaaaaaa SIOF xxxxxxxx vdev CC 0 CAW xxxxxxxx
 aaaaaaaa SIO xxxxxxxx vdev CC 1 CAW xxxxxxxx CSW xxxx
 aaaaaaaa mnem xxxxxxxx vdev CC 0
 aaaaaaaa mnem xxxxxxxx vdev CC 1 CSW xxxx
 aaaaaaaa mnem xxxxxxxx vdev CC 1 CSW xxxxxxxx xxxxxxxx
 aaaaaaaa mnem xxxxxxxx vdev CC 2
 aaaaaaaa mnem xxxxxxxx vdev CC 3

370-XA I/O Instructions

aaaaaaa SSCH xxxxxxxx address2 CC 0 SCH vsch DEV vdev
 CPA xxxxxxxx PARM xxxxxxxx KEY x FPI xx LPM xx

where:

CPA is the channel program address specifying the absolute address of the first channel command word to be executed on the device.

PARM is the interrupt parameter to be returned with any interrupt for the specified channel command words.

KEY is the storage protection key to be used.

FPI is the Format, Prefetch, Initial Status control byte.

LPM is the Logical Path Mask

aaaaaaa TSCH xxxxxxxx address2 CC 0 SCH vsch DEV vdev
 CCWA xxxxxxxx DEV STS xx SCH STS xx CNT xxxx
 KEY x FPI xx CC x CTLS xxxx

aaaaaaa TSCH xxxxxxxx address2 CC 1 SCH vsch DEV vdev
 KEY x FPI xx CC x CTLS xxxx

where:

CCWA is the ending channel command word address when I/O completed.

DEV STS is the unit status for the device.

SCH STS is the channel status for the subchannel.

CNT is the residual CCW count upon I/O completion.

KEY is the key used for storage protection.

FPI is the format, prefetch, initial status control byte used. It also indicates a no-path-available condition and a confirmed I/O condition code 0.

CC x is the condition code for the requested I/O operation. Unlike 370 I/O, this is always returned separately from the SSCH instruction.

CTLS are the function and activity control bytes.

```
aaaaaaaa TPI xxxxxxxx address2 CC 1
      SCH xxxx DEV xxxx PARM xxxxxxxx
```

where:

SCH is the subchannel requesting the interruption.

DEV is the device ID for the subchannel.

PARM is the interruption parameter supplied by the virtual machine, zero if the interrupt is unsolicited.

Instructions/Condition Codes

```
aaaaaaaa mnem xxxxxxxx address2 CC n SCH vsch DEV vdev
```

Executed I/O Instructions

```
aaaaaaaa EX xxxxxxxx aaaaaaaaa mnem xxzxxxxx address2 CC n
      SCH vsch DEV vdev
```

(Followed by fields as before)

CCW Tracing Display Format:

Seek CCWs for Mini-DASDs:

```
CCW   aaaaaaaaa gggggggg gggggggg  oooo  hh..... hhhhhhhh
SEEK           gggggggg gggg           hhhhhhhh hhhh
```

Seek CCWs for Attached DASDs:

```
CCW   aaaaaaaaa gggggggg gggggggg  oooo  hhhhhhhh hhhhhhhh
SEEK           gggggggg gggg           hhhhhhhh hhhh
```


TRACE: Command Responses

CCWs with a Virtual Channel Simulator:

```
CCW          0000 hh..... hhhhhhhh
SEEK        hhhhhhhh hhhh
```

Seek CCW — Set File Mask CCW Sequence with a Virtual Channel Simulator:

```
VDEV vvvv  CCW  gggggggg gggggggg STS ss ss DATA STS ss
           CCW  gggggggg gggggggg STS ss DATA gg STS ss
```

Seek CCW — Set File Mask CCW Sequence with a Real Channel:

```
CCW          0000 hh..... hhhhhhhh
SEEK        hhhhhhhh hhhhhhhh
CCW          0000 hh..... hhhhhhhh
           hh
```

Search CCWs for Attached DASDs:

```
CCW  aaaaaaaa gggggggg gggggggg 0000 hhhhhhhh hhhhhhhh
SEARCH          gggggggg gg          hhhhhhhh hh
```

Search CCWs for Mini-DASDs:

```
CCW  aaaaaaaa gggggggg gggggggg 0000 hh..... hhhhhhhh
SEARCH          gggggggg gg          hhhhhhhh hh
```

TIC CCWs:

```
CCW  aaaaaaaa gggggggg gggggggg 0000 hh00oooo hhhhhhhh
```

TIC CCWs with Virtual Channel Simulator:

```
CCW  gggggggg gggggggg TIC  aaaaaaaa
```

Other CCWs:

```
CCW  aaaaaaaa gggggggg gggggggg 0000 hhhhhhhh hhhhhhhh
```

CCWs with Data Crossing Page Boundaries:

```

CCW   aaaaaaaaa gggggggg gggggggg  0000  hh..... hhhhhhhh
IDAL
IDAL
...   as many IDALs as needed      ...

```

CCWs with User IDALs:

```

CCW   aaaaaaaaa gggggggg gggggggg  0000  hh..... hhhhhhhh
IDAL          gggggggg             hhhhhhhh
IDAL          gggggggg             hhhhhhhh
...

```

CCWs with Virtual Channel IDAWs:

```

CCW
IDAW  gggggggg DATA gggg

```

Data Transfer over 8 Bytes with Virtual Channel Simulator:

```

CCW                                     hhhhhhhh hhhhhhhh
                                     hhhhhhhh hhhh

```

Suspend with Virtual Channel Simulator:

```

CCW   gggggggg gggggggg ***SUSPENSION***

```

Define Extent CCWs for Mini-DASDs:

```

CCW   aaaaaaaaa gggggggg gggggggg  0000  hh..... hhhhhhhh
EXTENT          hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh

```

Locate Record CCWs for Mini-DASDs:

```

CCW   aaaaaaaaa gggggggg gggggggg  0000  hh..... hhhhhhhh
LOCATE          hhhhhhhh hhhhhhhh hhhhhhhh hhhhhhhh

```

Search CCWs for Attached DASDs:

```

CCW   aaaaaaaaa gggggggg gggggggg  0000  hhhhhhhh hhhhhhhh
SEARCH          gggggggg gg         hhhhhhhh hh

```

TRACE: Command Responses

Search CCWs for Mini-DASDs:

```
CCW   aaaaaaaaa gggggggg gggggggg   0000 hh..... hhhhhhhh  
SEARCH          gggggggg gg          hhhhhhhh hh
```

TIC CCWs:

```
CCW   aaaaaaaaa gggggggg gggggggg   0000 hh000000 hhhhhhhh
```

TIC CCWs with Virtual Channel Simulator:

```
CCW   gggggggg gggggggg TIC aaaaaaaa
```

Other CCWs:

```
CCW   aaaaaaaaa gggggggg gggggggg   0000 hhhhhhhh hhhhhhhh
```

CCWs with Data Crossing Page Boundaries:

```
CCW   aaaaaaaaa gggggggg gggggggg   0000 hh..... hhhhhhhh  
IDAL          hhhhhhhh  
IDAL          hhhhhhhh  
...      as many IDALs as needed   ...
```

CCWs with User IDALs:

```
CCW   aaaaaaaaa gggggggg gggggggg   0000 hh..... hhhhhhhh  
IDAL          gggggggg          hhhhhhhh  
IDAL          gggggggg          hhhhhhhh  
...      ...      ...
```

CCWs with Virtual Channel IDAWs:

```
CCW  
IDAW gggggggg DATA gggg
```

Data Transfer over 8 Bytes with Virtual Channel Simulator:

```
CCW          hhhhhhhh hhhhhhhh  
          hhhhhhhh hhhh
```

Suspend with Virtual Channel Simulator:

```
CCW   gggggggg gggggggg ***SUSPENSION***
```

Migration Notes

VM/SP HPO

1. In VM/XA SP, TRACE stops execution after each traced instruction, instead of before each traced instruction, as in VM/SP HPO.
2. In VM/XA SP, the traceback table shows only those branch instructions which are in the range of branches being traced.
3. If the maximum branch count is exceeded in VM/XA SP, the traceback table shows the maximum branch count. It is not set back to one as in VM/SP HPO.
4. VM/XA SP allows a minimum truncation of TERM for TERMINAL.
5. VM/XA SP supports additional operands and options including many of the VM/SP HPO PER command functions.
6. VM/SP HPO accepts TERM BOTH or PRINTER BOTH; VM/XA SP considers TERM BOTH and PRINTER BOTH conflicting options.
7. The VM/SP HPO options GUESTR, GUESTV, CURRENT, PRIV, CCW, and CSW are not accepted in VM/XA SP.
8. VM/SP HPO accepts EN for END; VM/XA does not.
9. The VM/SP HPO PER END tracetype (like PER END BR) is not accepted in VM/XA SP.
10. VM/SP HPO accepts PRO as a minimum truncation for PROGRAM; PROG is the minimum truncation in VM/XA SP.
11. When VM/SP HPO does not recognize a word on the TRACE command, it displays "003 Invalid option - word." VM/XA SP displays "002 Invalid operand - word."
12. An attempt to modify a PER trace set when none exists results in return code 180 and the message "TRACE NOT IN EFFECT." In VM/XA SP, an attempt to modify a trace set when none exists results in return code 47 and the message "The current traceset does not exist."
13. In VM/SP HPO, the output for PER STORE into 20500 includes an MVCL. VM/XA SP does not.
14. VM/XA SP does not allow the MASK operand.
15. When no trace is active, the VM/XA SP response to a QUERY PER command is RC 0, "NO TRACE SETS DEFINED." In VM/SP HPO you receive RC 180, "DMKPEQ180E PER trace not in effect."

TRACE ALL**Privilege Class: G**

Use TRACE ALL to trace all program, external, and I/O interruptions in addition to all instructions executed.

TRace	[ALL] [common options]
-------	------------------------

where:

ALL

causes all program, external, and I/O interruptions to be traced in addition to all instructions executed. If you omit ALL, any options you specify are valid only for the trace currently in effect. A new trace will not be started.

common options

You can specify any of the common options:

[PSWA FRom Range]	range	[RUN NORun]	
[TERminal NOTERMinAl Both]	[Printer NOPRINTEr]	[SUPervisor PROBLEm]	[DAT NODAT]
[IDentifer ident]			
[CMd ... cp command ...]			
[NOCMD]			

For a full explanation of the common options, see "TRACE: Common Options" on page 659.

Usage Notes

1. Because ALL encompasses system events in your entire address range, the PSWA option has no function if used with TRACE ALL. If an address range is specified by the PSWA operand, it will be ignored.
2. Each TRACE ALL command replaces an active TRACE ALL command.

Responses

See "TRACE: Command Responses" on page 663.

TRACE APPEND

Privilege Class: G

Use the TRACE APPEND command to merge the trace traps from two trace sets into one trace set.

TRace	APPEND name
-------	-------------

where:

APPEND

merges the traps from the named trace set with the current trace set.

name

specifies the trace set whose traps are to be merged into the current trace set.

Usage Notes

1. The TRACE APPEND command causes all trace traps from the named trace set to be added to the current trace set.

If a trap exists in both trace sets with the same identifier and the trap was named using the ID option on the TRACE command, then the trap in the named set will replace the trap in the current trace set.

If two traps have the same identifier but were not named with the ID option (for example, the identifier was assigned by the system), then both traps will remain, and the trap from the named trace set will be assigned a new identifier.

2. The current trace set cannot be appended to itself.

Responses

None.

TRACE BRANCH**Privilege Class:** G

Use TRACE BRANCH to trace successful branch instructions, including LPSW and SVC instructions that do not load a wait PSW.

TRace	BRAnch [INTO range] [common options]
-------	--------------------------------------

where:

BRANCH

traces successful branch instructions

INTO range

permits a trap to be taken if the destination of a successful branch instruction is within the specified range.

You may specify the range as a single address or a pair of addresses separated by a “,” or “-.” A dash denotes a starting and ending range. A period denotes a starting range for a byte count. A branch destination is considered to be within the range if any byte is within the range.

The addresses are real or virtual as determined by the setting of the DAT bit in the PSW at the time the event is detected. For BRANCH tracing, the address or range of addresses can be odd or outside of the address space; the event will be detected after the branch instruction but before any attempt is made to fetch the next instruction.

common options

You can specify any of the common options:

[{ PSWA } range]	[RUN]		
[FFrom]	[NORun]		
[Range]			
[SKIP n]	[PAss n]	[STOP n]	[STEp n]
[TERminal]	[PrInTer]	[SUPerVIsor]	[DAT]
[NOTERmInal]	[NOPRINTer]	[PROBlem]	[NODAT]
[Both]			
[IDentifer ident]			
[CMd ... cp command ...]			
[NOCMD]			

For a full explanation of the common options, see “TRACE: Common Options” on page 659.

Usage Notes

1. Your virtual machine will run very slowly after you have issued the **TRACE BRANCH** instruction.
2. While **TRACE BRANCH** is active, a record of branch instructions is maintained in the traceback table. You can use the **TRACE TABLE** command to display this information.

Responses

See “**TRACE: Command Responses**” on page 663.

TRACE CLEAR

Privilege Class: G

Use TRACE CLEAR to delete all traps in a named trace set. The set itself will be deleted if it is not active and it is not found in the RETURN list constructed by TRACE CALL commands.

TRace	CLEAR [name]
-------	--------------

where:

CLEAR [name]

deletes all traps in the named trace set. If you do not specify a name, the active trace set is cleared.

Usage Notes

None.

Responses

See "TRACE: Command Responses" on page 663.

TRACE COUNT

Privilege Class: G

Use TRACE COUNT to count the successful trace events in your virtual machine.

TRace	COunt
-------	-------

where:

COUNT

requests either the display of the current value of the TRACE COUNT, if active, or the initiation of the TRACE COUNT.

Usage Notes

1. The TRACE COUNT function allows you to count the number of successful trace events in the current trace set. While TRACE COUNT is active, no other trace output is produced. The virtual machine does not stop after each event, even if you specify NORUN.
2. When you enter a valid TRACE command, the system displays the value of the current count and TRACE COUNT terminates.

If you enter TRACE COUNT when COUNT is already active, then the system displays the current count and the count resumes from 0.

When the TRACE COUNT reaches 16 million, the system displays that count and resumes from 0.

3. The current COUNT is also included in the QUERY TRACE response.

Responses

Trace count is nnnnnnnn

precedes the display of trace traps if TRACE COUNT is active.

where:

nnnnnnnn

is the number representing the successful trace events in the current trace set.

TRACE END**Privilege Class:** G

Use TRACE END to delete all trace traps and all trace sets. No return list will remain.

Trace	END [<u>ALL</u>] [COunt]
-------	---------------------------------

where:

END [ALL]

deletes all trace traps and trace sets.

END [COUNT]

requests the display of the current value of the TRACE COUNT and the termination of TRACE COUNT.

Usage Notes

None.

Response 1

Trace ended

is displayed if tracing was in effect and you issued TRACE END ALL. If tracing was not in effect, the system displays no message.

Response 2

Trace count is nnnnnnnn

Trace count ended

is displayed if TRACE COUNT was active and you issued TRACE END COUNT.

where:

nnnnnnnn

is the number of successful trace events in the current trace set.

TRACE EXTERNAL/PROGRAM**Privilege Class:** G

Use TRACE EXTERNAL/PROGRAM to trace external or program interruptions or a range of external or program interruptions. When possible, the instruction that caused the program interruption will also be displayed.

TRace	{ EXTERNAL PROGRAM PRG }	[range]	[common options]
-------	--------------------------------	---------	------------------

where:

EXTERNAL
PROGRAM
PRG

traces external or program interruptions, respectively.

range

is a hexadecimal range of external or program interruptions. You can express this range as a single value, a pair of values separated by a “-,” or an address followed by a period “.” and followed by a byte count. Wrap-around ranges are not allowed.

common options

You can specify any of the common options:

{ PSWA FRom Range }	range	[RUN NORun]
[SKIP n]	[PAss n]	[STOp n]
[TErMinal NOTErMinal BOth]	[PPrinter NOPRINTEr]	[SUPerVIsor PROBlEm]
[DAT NODAT]		
[IDentifer ident]		
[CMd ... cp command ...]		
[NOCMD]		

For a full explanation of the common options, see “TRACE: Common Options” on page 659.

Usage Notes

None.

Responses

See “TRACE: Command Responses” on page 663.

TRACE GOTO/CALL/RETURN**Privilege Class:** G

Use TRACE GOTO, TRACE CALL, and TRACE RETURN to activate TRACE sets by name.

TRace	{ Goto name CALL name RETurn }	[SUSpend]
-------	--------------------------------------	-------------

where:

GOTO name

causes a trace set by the given name to become active. Specify name as a unique 1 to 8 character identifier. If no trace set exists by that name, a new, empty set is created. No program degradation occurs from an empty (null) trace set. The previous trace set is unchanged but no longer active. If empty, it may be deleted automatically.

You can use TRACE GOTO as the first command in a tracing session to establish the name of an initial trace set. If traps are created without issuing a TRACE GOTO, the initial trace set containing them will be named INITIAL.

CALL name

causes a trace set by the given name to become active. If no trace set exists by that name, a new, empty set is created. No program degradation occurs from an empty (null) trace set. The previous trace set is unchanged and is placed in a "Return" list on a last-in, first-out basis.

You can issue the CP QUERY TRACE RETURNS command to display a list of trace sets currently in the return list; the order is from most to least recent. Recursive calling is not allowed; it is illegal to call any trace set that has itself called a trace set.

TRACE CALL is designed to be used with TRACE RETURN.

RETURN

causes the first trace set on the return list to be removed from the list and become active. The previous trace set is unchanged but no longer active; if empty, the previous trace set is deleted automatically.

SUSPEND

causes the trace set being loaded to have no effect until the next GOTO, CALL, or RETURN occurs. You can issue TRACE commands to place traps for trace events into the set without affecting the program that is currently in operation. A suspended trace set will cause no program degradation.

Usage Notes

1. You can construct trace sets using a program that runs in your virtual machine. The program will retrieve commands from storage and pass them to the control program using DIAGNOSE instructions.

In most cases, the commands are to be applied to a subsequent program that is not yet in operation. You can use the SUSPEND option to insure that the trace set will have no effect until the next TRACE GOTO, TRACE CALL, or TRACE RETURN occurs.

Responses

See "TRACE: Command Responses" on page 663.

TRACE GPRs
Privilege Class: G

Use TRACE GPRs to trace alterations to your virtual machine's general purpose registers. You can trace all alterations to your registers, or only those that cause a particular register or set of registers to assume a specified value.

TRace	G[n[-m][,...]] [DATA range MASK string] [common options]
-------	---

where:

G

is the general register or range of general registers whose alterations you want to trace. If you are specifying a range of general registers, use the hyphen between the first register number and last register number in the range.

You may specify the register numbers in either decimal or hexadecimal. Wrap-around ranges (that is ranges whose ending register is less than the starting register) are permitted. For example:

trace G10-2

is a valid range; it establishes tracing of alterations to registers 10, 11, 12, 13, 14, 15, 0, 1, and 2.

DATA range

specifies a data value or range of data values to be checked to see if any of the altered register(s) assume a value within the range. The range may be a single value or a pair of values separated by a "." or "-." A dash denotes a starting and ending range. A period denotes a starting range for a byte count. Range values are viewed as unsigned 32-bit integers. If no data range is specified for general register alteration tracing, 0-FFFFFFFF is assumed. Wrap-around data ranges are not allowed.

MASK string

specifies a hexadecimal string which forms a mask for data comparison. The MASK string is logically applied to both the register and the data range before a comparison is made.

common options

You can specify any of the common options:

```

[ {PSWA } range ] [ RUN ]
[ {FRom } ] [ NORun ]
[ {Range } ]

[SKIP n] [PASS n] [STOP n] [STEp n]

[TERminal ] [Printer ] [SUPervisor ] [DAT ]
[NOTERminal ] [NOPRINTer ] [PROBlem ] [NODAT ]
[BOth ]

[IDentifer ident]

[CMd ... cp command ... ]
[NOCMD ]

```

For a full explanation of the common options, see “TRACE: Common Options” on page 659.

Usage Notes

1. Your virtual machine will run very slowly when you have issued the TRACE GPRs command.

Responses

See “TRACE: Command Responses” on page 663.

TRACE INSTRUCTION

Privilege Class: G

Use TRACE INSTRUCTION to trace instructions in your virtual machine, regardless of successful execution.

TTrace	INSTRuction [DATA string] [common options]
--------	--

where:

INSTRUCTION

causes the tracing of instructions.

DATA string

specifies a string of data to be compared against the instruction data.

common options

You can specify any of the common options:

$\left[\begin{array}{l} \{ \text{PSWA} \\ \text{FRom} \\ \text{Range} \} \text{ range} \end{array} \right]$	$\left[\begin{array}{l} \text{RUN} \\ \text{NORun} \end{array} \right]$
$[\text{SKIP } n] [\text{PASS } n] [\text{STOP } n] [\text{STEP } n]$	
$\left[\begin{array}{l} \text{TERminal} \\ \text{NOTERminal} \\ \text{BOTH} \end{array} \right]$	$\left[\begin{array}{l} \text{Printer} \\ \text{NOPRINTer} \end{array} \right]$
	$\left[\begin{array}{l} \text{SUPervisor} \\ \text{PROBlem} \end{array} \right]$
	$\left[\begin{array}{l} \text{DAT} \\ \text{NODAT} \end{array} \right]$
$[\text{IDentifer ident}]$	
$\left[\begin{array}{l} \text{CMd ... cp command ...} \\ \text{NOCMD} \end{array} \right]$	

For a full explanation of the common options, see "TRACE: Common Options" on page 659.

Usage Notes

1. If you use TRACE INSTRUCTION in conjunction with the PSWA option, only instructions within the range of the address specified by PSWA are traced.
2. You may use the DATA option to limit tracing to certain instructions or sequences of instructions. For example:

TRACE INSTR DATA 183F

will cause the tracing of all LR 3,15 instructions.

TRACE INSTR DATA 183F50

will trace all LR 3,15 instructions only if they are followed by a STORE (ST) instruction.

Responses

See “TRACE: Command Responses” on page 663.

TRACE I/O

Privilege Class: G

Use TRACE I/O to trace input/output instructions and/or input/output interruptions for all devices or for a range of devices.

TRace	[I/O IO]	[range]	[INSTRUCTION]	[CCW NOCCW]	[SIM NOSIM]	[INTERRUPT]	[common options]
-------	---------------	---------	---------------	----------------	----------------	-------------	------------------

where:

I/O

IO

traces I/O instructions or I/O interruptions.

range

is a range of devices. The range may be expressed as a single device number or pair of device numbers separated by a “-,” or a starting device number followed by a period “.” and followed by a number representing the number of consecutive device addressed to be traced. Wrap-around ranges are not allowed.

INSTRUCTION

specifies tracing of I/O instructions for a TRACE I/O command. If neither INSTRUCTION nor CCW nor INTERRUPT are specified, INSTRUCTION and INTERRUPT will be assumed by default.

CCW

specifies that CCW's will be displayed with RIO, SIO, SIOF, SSCH, and RSCH instructions that successfully initiate I/O. CCW and NOCCW are incompatible options.

NOCCW

specifies that CCW's will not be displayed. NOCCW will be assumed if CCW is not specified. CCW and NOCCW are incompatible options.

SIM

requests normal simulation of instructions. If neither SIM nor NOSIM are specified, SIM will be assumed by default. SIM and NOSIM are incompatible options.

NOSIM

requests deletion of instruction simulation. Through use of the CMD option, you can use it to define another operation for the instruction to perform. The PSW will be left pointing to the next sequential instruction. SIM and NOSIM are incompatible options.

INTERRUPT

specifies tracing of I/O interruptions for a TRACE I/O command. If neither INSTRUCTION nor CCW nor INTERRUPT are specified, INSTRUCTION and INTERRUPT will be assumed by default.

common options

You can specify any of the common options:

[PSWA } FRom } Range] range]	[RUN NORun]		
[SKIP n] [PAss n] [STOP n] [STEp n]			
[TERminal NOTERminal BOth]	[PrINTER NOPRINTER]	[SUPervisor PROBlem]	[DAT NODAT]
[IDentifer ident]			
[CMd ... cp command ...]			
[NOCMD]			

For a full explanation of the common options, see “TRACE: Common Options” on page 659.

Usage Notes

1. For 370-XA I/O, the device range applies to the device number, and not the subchannel number.
2. I/O interruption addresses are not subject to the instruction range (if any) specified by PSWA, FROM, or RANGE.

Responses

See “TRACE: Command Responses” on page 663.

TRACE MCH**Privilege Class:** G

Use TRACE MCH to intercept machine check interruptions. When possible, the instruction that caused the program interruptions will also be displayed. If you have issued TRACE MCH and a machine check occurs, the tracing facility saves all the necessary information prior to the machine check.

TRace	MCH [common options]
-------	----------------------

where:

MCH

indicates that virtual machine checks are to be trapped.

common options

You can specify any of the common options:

<table> <tr> <td rowspan="3"> $\left. \begin{array}{l} \text{PSWA} \\ \text{FRom} \\ \text{Range} \end{array} \right\}$ </td> <td>range</td> <td> <table> <tr> <td>RUN</td> </tr> <tr> <td>NORun</td> </tr> </table> </td> </tr> </table>	$\left. \begin{array}{l} \text{PSWA} \\ \text{FRom} \\ \text{Range} \end{array} \right\}$	range	<table> <tr> <td>RUN</td> </tr> <tr> <td>NORun</td> </tr> </table>	RUN	NORun	[SKIP n]	[PASS n]	[STOP n]	[STEP n]							
$\left. \begin{array}{l} \text{PSWA} \\ \text{FRom} \\ \text{Range} \end{array} \right\}$		range	<table> <tr> <td>RUN</td> </tr> <tr> <td>NORun</td> </tr> </table>	RUN	NORun											
		RUN														
	NORun															
<table> <tr> <td>TERminal</td> <td>Printer</td> <td>SUPERvIsor</td> <td>DAT</td> </tr> <tr> <td>NOTERminal</td> <td>NOPRINTER</td> <td>PROBLEM</td> <td>NODAT</td> </tr> <tr> <td>Both</td> <td></td> <td></td> <td></td> </tr> </table>	TERminal	Printer	SUPERvIsor	DAT	NOTERminal	NOPRINTER	PROBLEM	NODAT	Both							
TERminal	Printer	SUPERvIsor	DAT													
NOTERminal	NOPRINTER	PROBLEM	NODAT													
Both																
[IDentifer ident]																
<table> <tr> <td>CMd ... cp command ...</td> </tr> <tr> <td>NOCMD</td> </tr> </table>	CMd ... cp command ...	NOCMD														
CMd ... cp command ...																
NOCMD																

For a full explanation of the common options, see "TRACE: Common Options" on page 659.

Usage Notes

None

Responses

See "TRACE: Command Responses" on page 663.

TRACE mnemonic1

Privilege Class: G

Use TRACE mnemonic1 to trace the execution of specific assembler language instructions.

TRace	mnemonic1	[<u>SIM</u> NOSIM]	[common options]
-------	-----------	------------------------	------------------

where:

mnemonic1

must be one of the following standard assembler opcode mnemonics:

CONCS	PGOUT	SCK	SSYN	STNSM
DISCS	PTLB	SERV	STAP	STOSM
IPTE	RCHP	SIE	STCPS	STPX
ISK	RDD	SIGP	STCRW	STSYN
ISKE	RRB	SPX	STCTL	TB
LCTL	RRBE	SSK	STIDC	TPROT
LPSW	SAL	SSKE	STIDP	WRD
PGIN	SCHM	SSM	IUCV	

Tracing is limited to those instructions that do not produce an operation or privileged operation exception.

SIM

requests normal simulation of instructions. If neither SIM nor NOSIM are specified, SIM will be assumed by default. SIM and NOSIM are incompatible options.

NOSIM

requests deletion of instruction simulation. Through use of the CMD option, you can use it to define another operation for the instruction to perform.

You can embed instructions in your assembled code as breakpoints and use TRACE mnemonic1 to stop execution; or you can use TRACE mnemonic1 to stop program execution before a questionable instruction is executed. The PSW will be left pointing to the next sequential instruction.

SIM and NOSIM are incompatible options.

common options

You can specify any of the common options:

```
[ {PSWA } range ] [ RUN ]  
[ FRom }  
[ Range } ] [ NORun ]  
  
[SKIP n] [PAss n] [STOP n] [STEp n]  
  
[TERminal ] [Printer ] [SUPervisor ] [DAT ]  
[NOTERminal ] [NOPRINTer ] [PROBlem ] [NODAT ]  
[Both ]  
  
[IDentifer ident]  
  
[CMd ... cp command ... ]  
[NOCMD ]
```

For a full explanation of the common options, see "TRACE: Common Options" on page 659.

Usage Notes

None.

Responses

See "TRACE: Command Responses" on page 663.

TRACE mnemonic2

Privilege Class: G

Use TRACE mnemonic2 to trace specific input/output (I/O) instructions within a device range.

TRace	mnemonic2 [range] [SIM NOSIM]	[common options]
-------	----------------------------------	------------------

where:

mnemonic2

must be one of the following standard assembler opcode mnemonics:

CLRCH	HDV	HSCH	STSCH	TIO
CLRIO	HIO	MSCH	TCH	TSCH
CSCH				

Tracing is limited to those instructions that do not produce an operation or privileged operation exception.

range

is a range of devices. The range may be expressed as a single device number or pair of device numbers separated by a “-,” or a device number followed by a number which represents the number of consecutive device addresses to be traced. Wrap around ranges are not allowed.

SIM

requests normal simulation of instructions. If neither SIM nor NOSIM are specified, SIM will be assumed by default. SIM and NOSIM are incompatible options.

NOSIM

requests deletion of instruction simulation. Through use of the CMD option, you can use it to define another operation for the instruction to perform. The PSW will be left pointing to the next sequential instruction. SIM and NOSIM are incompatible options.

common options

You can specify any of the common options:

[{PSWA } FRom Range]	[range]	[RUN NORun]	
[SKIP n]	[PAss n]	[STOP n]	[STEp n]
[TERminal NOTERminal Both]	[Printer NOPRINTer]	[SUPervisor PROBlem]	[DAT NODAT]
[IDentifer ident]			
[CMd ... cp command ...]			
[NOCMD]			

TRACE mnemonic2

For a full explanation of the common options, see “TRACE: Common Options” on page 659.

Usage Notes

None.

Responses

See “TRACE: Command Responses” on page 663.

TRACE RIO/RSCH/SIO/SIOF/SSCH**Privilege Class:** G

Use TRACE RIO/RSCH/SIO/SIOF/SSCH to trace the SIO, SIOF, SSCH, RIO, or RSCH instructions for all input/output (I/O) devices, or for a range of devices. Tracing is limited to those instructions that do not produce an operation or privileged operation exception.

TRace	$\left. \begin{array}{l} \text{RIO} \\ \text{RSCH} \\ \text{SIO} \\ \text{SIOF} \\ \text{SSCH} \end{array} \right\} [\text{range}] \left[\begin{array}{l} \text{SIM} \\ \text{NOSIM} \end{array} \right] \left[\begin{array}{l} \text{CCW} \\ \text{NOCCW} \end{array} \right] [\text{common options}]$
-------	---

where:

RIO
RSCH
SIO
SIOF
SSCH

is the instruction that you want to trace.

range

is a hexadecimal range of device numbers. A range may be expressed as a single device number, a device number followed by a period followed by a number which represents the number of consecutive device addresses to be traced, or a pair of device numbers separated by a "." or "-." Wrap-around ranges are not allowed.

SIM

requests normal simulation of instructions. If neither SIM nor NOSIM are specified, SIM will be assumed by default. SIM and NOSIM are incompatible options.

NOSIM

requests deletion of instruction simulation. Through use of the CMD option, you can use it to define another operation for the instruction to perform. SIM and NOSIM are incompatible options.

CCW

specifies that CCWs will be displayed with the SIO, SIOF, SSCH, RIO, and RSCH instructions that successfully initiate I/O. CCW and NOCCW are incompatible options.

NOCCW

specifies that CCW's will not be displayed. If you specify neither CCW or NOCCW, NOCCW will be assumed. CCW and NOCCW are incompatible options.

TRACE RIO/RSCH/SIO/SIOF/SSCH

common options

You can specify any of the common options:

[PSWA } FRom } Range]	range	[RUN NORun]	
[SKIP n] [PASS n] [STOP n] [STEP n]			
[TERminal NOTERminal Both]	[PRINTer NOPRINTer]	[SUPervisor PROBlEm]	[DAT NODAT]
[IDentifer ident]			
[CMd ... cp command ...]			
[NOCMD]			

For a full explanation of the common options, see "TRACE: Common Options" on page 659.

Usage Notes

None.

Responses

See "TRACE: Command Responses" on page 663.

TRACE STORE

Privilege Class: G

Use TRACE STORE to trace alterations to virtual machine storage that were made by the processor. You can trace all alterations, or only those alterations causing a specified set of logical locations to assume a specified value.

TRace	STore	[INTO { range hexloc [DATA string [MASK string]] }]	[common options]
-------	-------	---	------------------

where:

STORE

traces alterations to virtual machine storage by the processor.

range

permits a trap to be taken if any data is stored into any location within the specified range.

The range may be a single address or a pair of addresses separated by a “.” or “-.” A dash denotes a starting and ending range. A period denotes a starting range for a byte count. A data specification is considered to be within the range if the first byte is within the range.

hexloc

permits a trap to be taken when the following DATA or MASK options for the location are met. See DATA and MASK below. The hexloc may be any valid address. It may not be a range of addresses.

DATA string

specifies a hexadecimal data string that is to be matched when a store occurs into the specified hexloc. The length of the data string determines the bounds of the test. If MASK is specified with DATA, the MASK is logically applied to both storage and the DATA string before the comparison is made. The DATA operand requires the INTO hexloc operand.

MASK string

specifies a hexadecimal string that forms a mask for data comparison when a store occurs into the specified hexloc. When you specify MASK, the MASK is logically applied to both storage and the DATA string before the comparison is made. Only those bits and bytes that are binary ones are checked. The MASK operand requires the INTO hexloc operand and the DATA option. The length of the MASK string must be the same as the length of the string you specified for the DATA option.

TRACE STORE

common options

You can specify any of the common options:

[PSWA } range]	[RUN]		
[FRom }	[NORun]		
[Range]			
[SKIP n] [PASS n] [STOP n] [STEP n]			
[TERminal]	[PRinter]	[SUPervisor]	[DAT]
[NOTERminal]	[NOPRINTer]	[PROBlEm]	[NODAT]
[Both]			
[IDentifer ident]			
[CMd ... cp command ...]			
[NOCMD]			

For a full explanation of the common options, see "TRACE: Common Options" on page 659.

Usage Notes

1. TRACE STORE does not trace "MVCL" instructions.
2. If the DATA option is specified, the length of the virtual storage that is monitored for alteration is defined by the length of the DATA string.

Responses

See "TRACE: Command Responses" on page 663.

TRACE SVC/DIAGNOSE/MC

Privilege Class: G

Use TRACE SVC/DIAGNOSE/MC to trace supervisor calls (SVC), DIAGNOSE instructions, or monitor calls (MC), or a range of these events.

TRace	$\left\{ \begin{array}{l} \text{SVC} \\ \text{DIAGnose} \\ \text{MC} \end{array} \right\}$	[range]	$\left[\begin{array}{l} \text{SIM} \\ \text{NOSIM} \end{array} \right]$	[common options]
-------	--	---------	--	------------------

where:

SVC
DIAGNOSE
MC

traces supervisor calls, diagnose instructions, or monitor calls, respectively.

range

is a hexadecimal range of SVC numbers, DIAGNOSE codes, or monitor call codes. A range may be specified as a single number or pair of numbers separated by a "." or "-." Wrap-around ranges are not allowed.

SIM

requests normal simulation of instructions. If neither SIM nor NOSIM are specified, SIM will be assumed by default. SIM and NOSIM are incompatible options.

NOSIM

requests deletion of instruction simulation. Through the use of the CMD option, you can use it to define another operation for the instruction to perform.

common options

You can specify any of the common options:

$\left[\left\{ \begin{array}{l} \text{PSWA} \\ \text{FRom} \\ \text{Range} \end{array} \right\} \text{ range} \right]$	$\left[\begin{array}{l} \text{RUN} \\ \text{NORun} \end{array} \right]$		
[SKIP n] [PAss n] [STOP n] [STEp n]			
$\left[\begin{array}{l} \text{TERminal} \\ \text{NOTERminal} \\ \text{Both} \end{array} \right]$	$\left[\begin{array}{l} \text{Printer} \\ \text{NOPRINTer} \end{array} \right]$	$\left[\begin{array}{l} \text{SUPervisor} \\ \text{PROBlem} \end{array} \right]$	$\left[\begin{array}{l} \text{DAT} \\ \text{NODAT} \end{array} \right]$
[IDentifer ident]			
$\left[\begin{array}{l} \text{CMd ... cp command ...} \\ \text{NOCMD} \end{array} \right]$			

For a full explanation of the common options, see "TRACE: Common Options" on page 659.

TRACE SVC/DIAGNOSE/MC

Usage Notes

None.

Responses

See "TRACE: Command Responses" on page 663.

TRACE TABLE

Privilege Class: G

Use TRACE TABLE to display the traceback table.

TRace	TAble
-------	-------

where:

TABLE

causes the traceback table to be displayed.

The traceback table is a list of successful branch instructions which is maintained while branch tracing is active. It contains the most recent branch instructions executed. If the same instruction is executed with no intervening branch instructions, then the number of times the instruction was executed will be maintained.

Up to six instructions are maintained in the traceback table. The most recent branch will be displayed as the last entry in the table.

Usage Notes

None.

Response 1

```
aaaaaaaa mnem xxxxxxxx -> ttttttt nn times
```

is displayed if there is at least one entry in the traceback table. This may be repeated for up to six entries. *nn times* specifies the number of times in a row this branch instruction was executed. This value is not displayed if $n = 1$.

Response 2

Traceback table is empty

is the response received if there are no entries in the traceback table.

TRACE TPI
Privilege Class: G

Use TRACE TPI to trace Test-Pending-Interrupt instructions. TRACE TPI is accepted only if your virtual machine is operating in 370-XA mode, and tracing is limited to those instructions that do not produce an operation or privileged operation exception.

TRace	TPI [range]	<u>SIM</u> NOSIM	[common options]
-------	-------------	---------------------	------------------

where:

TPI

traces Test-Pending-Interrupt instructions.

range

is a range of device numbers. If you specify a device range, only those TPI instructions that store an interruption will be traced. If you do not specify a range, all TPI instructions will be traced. The device range applies to the device name, and not the subchannel address.

The range may be a single device number, a device number followed by a period and followed by a number which represents the number of consecutive device addresses to be traced, or pair of device numbers separated by a "." or "-." Wrap-around ranges are not allowed.

NOSIM and a device range are incompatible options.

SIM

requests normal simulation of instructions. If neither SIM nor NOSIM are specified, SIM will be assumed by default. SIM and NOSIM are incompatible options.

NOSIM

requests deletion of instruction simulation. This is invalid when TPI specifies a device range. Through use of the CMD option, you can use it to define another operation for the instruction to perform. The PSW will be left pointing to the next sequential instruction. SIM and NOSIM are incompatible options.

common options

You can specify any of the common options:

<u>PSWA</u>	} range	<u>RUN</u>	
FRom		<u>NORun</u>	
Range			
[SKIP n]	[PAss n]	[STOP n]	[STEp n]
<u>TErMinal</u>	<u>PRinter</u>	<u>SUPervisor</u>	<u>DAT</u>
NOTERMinAl	<u>NOPRINTEr</u>	<u>PROBlEm</u>	<u>NODAT</u>
BOth			
[IDentifer ident]			
<u>CMd ... cp command ...</u>			
<u>NOCMD</u>			

For a full explanation of the common options, see “TRACE: Common Options” on page 659.

Usage Notes

None.

Responses

See “TRACE: Command Responses” on page 663.

TRACE TRAP

Privilege Class: G

Use the TRACE TRAP commands to alter or delete a trace trap by name.

TRace	{ TRAP ident [common options] } { DELEte }
-------	---

where:

TRAP

applies the options you specify to a single event. All primary TRACE operands establish units of control called “traps.” Each trap, except for TRACE ALL, is assigned a unique identifier and has its own set of options: TERM, PRINT, RUN, STEP, and so on. TRACE ALL creates a unique set of traps replacing any active TRACE ALL command. These traps each have an identifier of “ALL,” which may be used for changing trace options or deleting the trap. For those traps for which SIM and NOSIM are valid options, you can specify SIM or NOSIM with TRAP.

Any valid option you may specify with the TRAP operand you may specify without any primary TRACE operand. Not specifying a primary TRACE operand has the same effect as a TRACE TRAP command executed for each event in the current TRACE set.

DELETE

The DELETE operand deletes an event from the current set.

ident

specifies an identifier for a trace trap describing the event. It must be one to four characters in length. If a trap already exists with the given identifier, it will be replaced by the trap for the newly defined event. You cannot use the word “ALL” as an identifier; on the TRACE TRAP commands, this is an invalid option.

common options

Except for ID, you can specify any of the common options:

[{ PSWA } range]	[RUN]		
[{ FRom } Range]	[NORun]		
[SKIP n]	[PAss n]	[STOP n]	[STEp n]
[TERminal]	[PRinter]	[SUPervisor]	[DAT]
[NOTERminal]	[NOPRINTER]	[PROBLEM]	[NODAT]
[BOth]			
[CMd ... cp command ...]			
[NOCMD]			

For a full explanation of the common options, see “TRACE: Common Options” on page 659.

Usage Notes

1. Note that you cannot specify the ID option with TRACE TRAP to change the identifier of an existing trap. In order to change the identifier, you must redefine the trap.
2. With the exception of SIM and NOSIM, you can use TRACE TRAP to modify *only* those options listed here. To change unique options not listed here, you must redefine the trap. However, you can use TRACE TRAP to modify the SIM and NOSIM on any trap for which they are valid options.

Responses

See "TRACE: Command Responses" on page 663.

TRANSFER

Privilege Class: D,G

Use the TRANSFER command to send files to or retrieve files from another user's virtual reader, printer, or punch queue.

Class G users can manipulate files that they own or originate.

If you do not specify any of the options (SYSTEM, userid, or asterisk "*"), the default is to transfer or reclaim your own spool files ("**").

A class D user can transfer any spool files in the system, regardless of their ownership or origin.

TRANSfer	[SYSTEM]	[Printer PUnch Reader]	{Class c FORM form ALL}	(options)
	[userid *]	{Reader Printer PUnch}	{Class c spoolid FORM form ALL}	(options)
	(options)	{[To] {userid *} From {userid ALL}}	[Printer PUnch Reader]	

where:

SYSTEM

allows the class D user to manipulate all spool files in the system, regardless of their ownership. This is not allowed with the "spoolid" option.

userid

*

is the user identification of the user whose spool files are to be transferred or reclaimed. If you are manipulating your own files, you may specify userid as an asterisk (*). This is the default.

The userid option allows the class D user to manipulate files belonging to any user on the system. Class G users can use the userid option to manipulate files that they currently own or have originated.

PRINTER

PRT

PUNCH

PCH

READER

RDR

specifies the queues to which CP transfers files and from which files are reclaimed.

CLASS c

is the class of spool files you wish to transfer. The c is a one character alphanumeric field with values from A to Z and from 0 to 9.

FORM form

selects spool files with the specified form for transferring. For the class G user, the "form" is a 1- to 8-character form name. For the class D user, "form" is a form name when you specify "*" or default to your userid. Otherwise, "form" is the operator form number.

spoolid

is the spool file identifier of the file that you want to transfer or reclaim. This is not valid with the SYSTEM keyword.

ALL

specifies that you want to transfer all files in a reader/printer/punch queue.

[TO] {userid }
 * }

is the user to whom you want to transfer the files. If you omit the optional keyword TO, userid cannot be TO or T. You can use "*" to denote your own userid.

FROM {userid }
 ALL }

is the userid from whom you want to reclaim spool files of the specified reader, printer, or punch queue. You can specify ALL to reclaim all files that you originally created and transferred to the reader/printer/punch queues of other users.

Usage Notes

1. You cannot use the TRANSFER command to transfer any active spool files.
2. TRANSFER *spoolid* FROM ALL is no longer allowed because a spool ID without a userid specified is no longer meaningful.
3. If the NORETURN option was specified when the file was created by DIAGNOSE X'94', the originator of the file cannot retrieve that file from the current owner. If an attempt is made to retrieve the file, the system issues the following message:
 FILE NOT TRANSFERRED; FILE CREATED WITH THE NORETURN OPTION' is issued.
4. When a file changes ownership, the spool file ID number also changes. The new number is obtained from the pool of available IDs of the recipient of the file. Thus, if user BINGLE changes file #3 to user BOGUS, BINGLE gets the response:
 RDR 0003 SENT TO BOGUS RDR AS 0018 RECS 1000 COPY 001 A NOHOLD NOKEEP
 BOGUS would now have a file #18.

Examples

```
TRANSFER 0123 TO USERA
```

transfers the spool file with spoolid 0123 from your reader queue to the reader queue of USERA. Note that the file is in the reader queue, which is the default. If you want to transfer a printer or punch file, you must specify the device type in the command line.

TRANSFER

```
TRANSFER PRT 0123 TO USERA
```

transfers the spool file with spoolid 0123 from your print queue to the reader queue of USERA.

```
TRANSFER PRT ALL TO * RDR
```

transfers all the spool files on your print queue to your reader queue. Note that you didn't need to specify the default, RDR.

```
TRANSFER RDR CLASS B TO * PCH
```

transfers all the class B files on your reader queue to your own punch queue.

```
TRANSFER PRT ALL FROM ALL RDR
```

reclaims all spool files that you sent to the reader queues of other users and places them on your printer queue.

Response 1

Both the current and the previous owners of the file receive the following response when the previous owner issues a TRANSFER command with the TO option; it is also the response that the previous owner of the file receives if he or she issues a TRANSFER command with the FROM option.

```
{RDR} FILE spd1 SENT {TO } userid {RDR} AS spd2 RECS {nnnn} COPY [*]nnn c {HOLD } {KEEP }  
{PRT} {FROM} {PRT} {nnnK}  
{PUN} {PUN} {nnnM} {NOHOLD} {NOKEEP}
```

where:

RDR
PRT
PUN

(first occurrence) If you are transferring a file to another user's virtual machine, this field indicates the queue to which the file originated. If you are receiving a file from another user, this field indicates the file queue from which the file has been added.

spd1

is the spool identification number of the file at the time the command is issued.

TO userid

is the user identification of the new file owner.

FROM userid

is the user identification of the previous file owner.

RDR
PRT
PUN

(second occurrence) If you are transferring a file to another user's virtual machine, this field indicates the queue in which the file has been added. If you are receiving a file from another user, this field indicates the file queue in which the file originated.

spd2

is the spool identification number of the file when the command is completed.

RECS $\left\{ \begin{array}{l} \text{nnnn} \\ \text{nnnK} \\ \text{nnnM} \end{array} \right\}$

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

is the remaining number of copies of the file to be produced. The asterisk (*) is for the 3800 printer and allows the printer to copy each page nnn times before going on to the next page.

c

is a 1-character alphanumeric field representing the class with values from A to Z and from 0 to 9.

HOLD**NOHOLD**

is the user HOLD status associated with the file.

KEEP**NOKEEP**

is the KEEP status associated with the file.

Response 2

You receive the following response if you issue a TRANSFER command with the FROM option; and this is the response that the new owner of a file receives if you issue a TRANSFER command with the TO option.

$$\left\{ \begin{array}{l} \text{RDR} \\ \text{PRT} \\ \text{PUN} \end{array} \right\} \text{ FILE spd1 SENT } \left\{ \begin{array}{l} \text{TO} \\ \text{FROM} \end{array} \right\} \text{ userid } \left\{ \begin{array}{l} \text{RDR} \\ \text{PRT} \\ \text{PUN} \end{array} \right\} \text{ AS spd2 RECS } \left\{ \begin{array}{l} \text{nnnn} \\ \text{nnnK} \\ \text{nnnM} \end{array} \right\} \text{ COPY } [*] \text{ nnn c } \left\{ \begin{array}{l} \text{HOLD} \\ \text{NOHOLD} \end{array} \right\} \left\{ \begin{array}{l} \text{KEEP} \\ \text{NOKEEP} \end{array} \right\}$$

where:

RDR**PRT****PUN**

(first occurrence) is the queue on which the file now resides.

spd1

is the spool identification number of the file at the time the command is issued.

TO userid

is the user identification of the previous file owner.

FROM userid

is the user identification of the new file owner.

TRANSFER

RDR
PRT
PUN

(second occurrence) is the queue on which the file previously resided.

spd2

is the spool identification number of the file when the command is completed.

RECS $\left[\begin{array}{l} \text{nnnn} \\ \text{nnnK} \\ \text{nnnM} \end{array} \right]$

is the number of logical records in the spool file. If the number is greater than 9999, the number is shown as nnnK, where K represents the number of lines rounded to the nearest 1000 records. If the number is greater than 999499, then the number is shown as nnnM, where M represents the number of lines rounded to the nearest million.

COPY [*]nnn

is the remaining number of copies of the file to be produced. The asterisk (*) is for the 3800 printer and allows the printer to copy each page nnn times before going on to the next page.

c

is a 1-character alphanumeric field representing the class with values from A to Z and from 0 to 9.

HOLD
NOHOLD

is the user HOLD status associated with the file.

KEEP
NOKEEP

is the KEEP status associated with the file.

Response 3

$\left. \begin{array}{l} \text{nnnnnnn} \\ \text{NO} \end{array} \right\} \text{FILE(S) TRANSFERRED}$

is the number of files that were transferred. "NO" indicates there are no files matching the characteristics (such as CLASS and FORM) specified on the TRANSFER command.

Note: You do not receive any of these responses if you have issued the CP SET IMSG OFF command.

Migration Notes

VM/SP HPO

1. VM/XA SP does not allow users to specify SYSTEM and spoolid on the same command line. VM/SP HPO (Release 5) users do not see this difference.
2. The VM/XA SP response adds the number of records and KEEP setting.
3. VM/XA SP does not support the DEST option.
4. If the file is transferred to another user, the VM/XA SP response adds the new spoolid.

VM/XA SF

1. VM/XA SP does not allow users to specify SYSTEM and spoolid on the same command line;
2. If the file is changed to another user, the VM/XA SP response adds the new spoolid.

TRSAVE

Privilege Class: A, C

Use the TRSAVE command to specify where CP trace table data or data from traces defined by the TRSOURCE command are to be saved. CP trace table data may be saved either in system trace files or on tape. Data from traces defined by the TRSOURCE command may be saved only in system trace files.

TRSAve	<pre>[FOR] { ID traceid } [destination parameters] { [CP] { ON OFF CANCEL } }</pre> <p>destination parameters:</p> <pre>[DASD [TO userid] [NAME fname] [SIZE nnnn] [KEEP n] TAPE {rdev1 [rdev2]} [MODE nnnn] [RUN REWinc]]</pre>
--------	--

where:

FOR

is an optional keyword.

ID traceid

indicates that this TRSAVE applies to data generated by the specified trace ID defined by the TRSOURCE command.

CP ON

indicates that the recording of CP trace table data should begin.

CP OFF

stops CP system trace table data recording. Any remaining buffers of data are written to the output medium. Either a tape mark is written or the DASD file is closed, depending on the output medium. Tapes are rewound and unloaded.

CP CANCEL

stops CP system trace table data recording. Any pending I/O is canceled. Any data collected but not written is lost. When tape is the output medium, a tape mark is written and the tape is then rewound and unloaded, regardless of the previous tape positioning option.

DASD

indicates that trace data is to be recorded in system trace files. DASD is the default.

TO userid

* indicates the destination of the data. userid specifies which userid should receive the system trace file containing the trace data. * specifies that the user who issues the command should receive the system trace file containing the trace data. * is the default.

NAME fname

specifies the filename to be assigned to the system trace file. For CP system trace table data, CPTRACE is the default name. For trace data generated by traces defined by the TRSOURCE command, the trace ID is the default name.

SIZE nnnn

indicates the maximum size of the system trace files. nnnn is a decimal number that specifies 4Kb pages of trace data. Once the file reaches the limit, it is closed and a new file is opened when the next trace record is received. The default is 256 pages. The maximum is 9999 pages.

KEEP n

specifies the total number of trace files to be kept on DASD for this recording function. The default is two files. The maximum is nine files.

TAPE

indicates that trace data is to be recorded on tape. This option is valid only for recording CP system trace tables.

rdev1 [rdev2]

indicates that the CP trace table data captured by this trace should be recorded to the tape drive at the specified device number. Either one or two tape drives can be specified on any tape option. These tape drives must be system drives and must not be attached to any users.

If two tape drives are specified, CP automatically switches to the alternate tape drive when it encounters an end-of-tape mark.

MODE nnnn

specifies the tape density. If you specify two tape drives, MODE specifies the tape density to be used by both tape drives. You may specify a mode value of 800, 1600, 6250, or 38K (for 3480 devices only). The default is 1600 for 3420 devices and 38K for 3480 devices.

RUN

specifies that the tape is rewound and unloaded when it fills. This is the default.

REWIND

specifies that the tape is rewound but not unloaded when it fills. Recording continues either to the other tape drive devoted to CP tracing or to the tape drive on which the tape was just filled.

Usage Notes

1. The destinations for trace data are not saved across system hard abends. Therefore, the destination for the trace data should be saved in EXECs.
2. The parameter TAPE and tape options can only be used on TRSAVE FOR CP ... commands.
3. TRSAVE FOR ID *traceid* cannot be issued when the specified trace ID has been enabled using the TRSOURCE command.
4. The optional destination parameters (TO, NAME, SIZE, and KEEP for DASD, and MODE, RUN, REWIND for TAPE) may be specified in any order.
5. For TRSAVE FOR ID *traceid*, the specified parameters remain in effect until the trace ID is dropped or they are changed on a subsequent TRSAVE command. For example, if you issued the commands TRSAVE ID DEBUG

KEEP 5 SIZE 9 and then TRSAVE ID DEBUG SIZE 20, the trace ID DEBUG would have TRSAVE options being recorded to a system trace file with a filename of DEBUG, a SIZE of 20 pages, a KEEP of 5 files, and your userid as receiver.

6. The use of two tape drives is recommended to minimize loss of data. With one tape drive, data may be lost while you rewind and reset for the next tape. If two tape drives are used, both must be 3420 devices or both must be 3480 devices.

The use of the REWIND option is also recommended. If the REWIND option is not used, a full tape is rewound and unloaded. If one tape drive is used without the REWIND option, data may be lost until the operator mounts another tape. Even if two tape drives are used, data may be lost on heavily loaded systems when the second tape drive becomes full and the operator has not yet (a) mounted another tape on the first tape drive or (b) satisfied the operation intervention on the first drive.

7. The following chart shows the default filenames for system trace files and their filetypes. The filenames may be changed with the NAME fname option.

Table 11. Default Filenames and Filetypes for System Trace Files		
Source of Trace	Filename	Filetype
CP system trace data	CPTRACE	CP
TRSOURCE ID traceid TYPE DATA	traceid	DATA
TRSOURCE ID traceid TYPE IO	traceid	IO
TRSOURCE ID traceid TYPE GT FOR USER	traceid	VM
TRSOURCE ID traceid TYPE GT FOR VMGROUP	traceid	VMG

8. When a trace is active, the files associated with it have a class of W. After tracing is complete, the associated files have a class of A.
9. On stressed systems, the trace table may wrap before all entries are saved. In this case, a special trace entry is made to denote data loss.
10. TRSAVE CANCEL immediately terminates the TRSAVE function. Any I/O that is pending is canceled and any information that has been merged and has not been written is lost. To ensure that all information collected has been written, use the OFF option on the TRSAVE command.
11. TRSAVE ON is valid only when TRSAVE is not currently active. Likewise, TRSAVE OFF is valid only when TRSAVE is currently active. You may use TRSAVE CANCEL while TRSAVE OFF is in process (for example, when the tape is in intervention required), but it terminates any I/O that is currently pending.
12. The system attempts to write a tape mark at the end of the data on tape when OFF and CANCEL processing is completed. This attempt may fail because of unrecoverable I/O errors on the tape, but you may still process the tape using the dump viewing facility TRACERED command.
13. KEEP and SIZE may be used together to manage the system spool space. They limit the total amount of space available to record trace data for this trace request. CP limits the amount of data in each file, according to the SIZE specified. By designating a number of files to be kept, TRSAVE allows CP to purge the oldest trace file, keeping the most recent data and continuing to trace when the allowable space limit is reached.

14. Each time a trace becomes active, the trace data is recorded in another file or set of files. For example, if you were to enter the following commands, you could generate up to six files:

```
TRSAVE FOR CP DASD KEEP 3
TRSAVE OFF
TRSAVE FOR CP DASD KEEP 3
TRSAVE OFF
```

Examples

- To record system trace table entries on DASD and specify USERA as the file owner, enter:
TRSAVE FOR CP ON DASD TO USERA
- To turn off TRSAVE for CP, enter:
TRSAVE FOR CP OFF
- To turn on TRSAVE for the trace defined by TRSOURCE ID DEBUG... , save the recorded data in a file named FILEB, and allow DUSER to process the files, enter:

```
TRSAVE FOR ID DEBUG DASD TO DUSER NAME FILEB
```

Responses

For CP tracing, the system responds with:

$$\text{TRSAVE function } \left\{ \begin{array}{l} \text{started} \\ \text{completed} \\ \text{cancelled} \end{array} \right\} \left\{ \begin{array}{l} \text{: nnnn files purged} \\ \text{: nnnn files purged} \end{array} \right\}$$

where:

nnnn

is the number of trace files that have been purged during the time the trace service tool function was active due to the wrapping of the KEEP limit. When the KEEP limit is reached, the oldest file is purged. If the response states that 9999 files were purged, it is possible that more than 9999 files were purged.

You receive the above response when you issue TRSAVE FOR CP... and the TRSAVE command is started, completed, or canceled. If you are not logged on, the completed and canceled responses are sent to the system operator.

No response is issued when TRSAVE FOR ID *traceid* ... commands complete because, while TRSAVE defines the destination of traces defined by TRSOURCE, recording of those traces is started and stopped by the ENABLE and DISABLE parameter of the TRSOURCE command.

With both the TRSAVE FUNCTION COMPLETED and TRSAVE FUNCTION CANCELLED responses, the trace service tool function has been turned off. The second half of these responses is issued only when DASD was specified on the TRSAVE ON command (it is not issued for CP traces to tape).

TRSAVE

Migration Notes

VM/SP Release 6: TRSAVE replaces the destination operands on the CPTRAP command. VM/SP saves all trace data in one spool file. VM/XA SP creates a system data file for each trace defined by TRSOURCE and for CP system trace table data if that data is being recorded. Therefore, additional parameters have been added to specify which trace data the command applies to.

VM/XA SP Release 1: Because trace data can now be saved on both tape and in system data files, the TAPE option must be used to send CP trace table data to tape.

Examples:

TRSAVE ON 181 MODE 38K ... -- in VM/XA SP Release 1 is equivalent to

TRSAVE ON TAPE 181 MODE 38K ... -- in VM/XA SP Release 2.

TRSOURCE

Privilege Class: C

Use the TRSOURCE command to support definitions and control of I/O, data, and guest tracing. The TRSOURCE command is functionally organized as follows:

- **TRSOURCE ID** — Defines a trace. TRSOURCE ID associates an event (such as execution of a particular piece of code, an interrupt on a specific device, and so on) with a name (trace ID) and, optionally, with a set of other traces (trace set).

The TYPE operand of TRSOURCE ID assigns a trace ID to one of the three trace types. These types are:

1. **IO (I/O tracing)** — The IO trace allows you to trace all IO activity for a specific real device or range of real devices. The times, channel programs, SCSWs and data transferred for each I/O operation to the specified devices are made available for data analysis.
 2. **DATA (data tracing)** — The data trace allows the user to define trace entries dynamically to record execution of most code paths in CP. The LOC parameter specifies what instruction, when executed, triggers recording of information. Data link parameters define what information is collected when the trap starts.
 3. **GT (guest tracing)** — The guest trace allows a virtual machine or a group of virtual machines to record events in a system trace file.
- **TRSOURCE ENABLE** — Activates the tracing of individual trace IDs, and of trace IDs within trace sets.
 - **TRSOURCE DISABLE** — Deactivates the tracing of individual trace IDs, and of trace IDs within trace sets.
 - **TRSOURCE DROP** — Disables and drops all CP knowledge of individual trace IDs and trace IDs within trace sets.
 - **TRSOURCE DISPLAY** — Displays the status resulting from the TRSOURCE commands you have issued.

TRSOURCE Format

The syntax diagrams show the operands available with the TRSOURCE command. Complete descriptions of the operands appear on the following pages:

TRSource	<p>DISAble { [ID t1 t2...tx] [SET s1 s2...sy] } [DISPlay]</p> <p>DROp</p> <p>ENAbLe { ALL }</p> <p>DISPlay</p> <p>ID traceid [SET { traceset }] [TYPE { IO }] [type parameters] [DISPlay]</p> <p style="margin-left: 20px;"> { <u>NULL</u> }</p> <p style="margin-left: 20px;"> { GT } { DATA }</p> <p>Parameters for Type IO:</p> <p>DEvIce rdev1[-rdev2] [rdev3[-rdev4]] ...</p> <p>USER { userid }</p> <p style="margin-left: 20px;"> { <u>ALL</u> }</p> <p style="margin-left: 20px;"> { SYSTEM }</p> <p>IOData { nnnn }</p> <p style="margin-left: 20px;"> { 0 }</p> <p>Parameters for Type GT:</p> <p>FOR { [USER] userid }</p> <p style="margin-left: 20px;"> { VMGROUP sysname { <u>ALL</u> } }</p> <p style="margin-left: 40px;"> { ONLY u1 u2...ux }</p> <p style="margin-left: 40px;"> { EXCLUDE u1 u2...ux }</p> <p style="margin-left: 40px;"> { INCLUDE u1 u2...ux }</p> <p>Parameters For Type DATA:</p> <p>LOC { address instruction }</p> <p style="margin-left: 20px;"> { modname + disp instruction }</p> <p>DL datalinks</p>
----------	---

General TRSOURCE Command Usage Notes

1. All traces are disabled at system termination.
2. The definitions of traces are not saved across system IPLs. Therefore, the definitions of any commonly defined traces should be saved in EXECs.
3. When a new member (trace ID) of an enabled set is defined, that member is not automatically enabled. It must be enabled separately. Attempts to enable trace IDs that are already enabled do not alter anything and do not result in an error message.
4. The definition of what is being traced for each trace ID cannot be altered while the trace ID is enabled. Any trace ID may be altered while it is disabled.
5. Use of the DROP keyword directed toward a trace set removes any record of all the trace IDs in the set. To remove or "ungroup" a member of a set, the user should either assign the trace ID to another, possibly unused, trace set name or assign the trace ID to a trace set name of NULL.

6. The destination of trace data can be altered with the TRSAVE command while the trace is disabled. See “TRSAVE” on page 710 for details.
7. Enabling a trace for which TRSAVE has not been issued invokes the TRSAVE destination defaults.

TRSOURCE Command Examples

The following examples provide an overview of how TRSOURCE can be used:

1. This example defines and enables a trace ID of type DATA. The trace ID is called DSPLOOP, it is defined at address 4B248, and the data link string specified is G4+8%.2. The destination is specified as a system trace file for the issuer with a maximum size of 2000.

```
trsource id dsploop type data loc 4b248 5820c846
trsource id dsploop dl g4+8%.2
trsave id dsploop dasd to * size 2000
trsource enable id dsploop
```

2. This example defines, enables, and disables two trace IDs of type IO. These trace IDs, IO1 and IO2, are put into trace set IOSET. IO1 traces I/O to devices in the range c41-c47; up to 200 bytes of transferred data is traced. IO2 traces I/O for USER1 to devices in the range a81-a87; up to 400 bytes of transferred data is traced. The destination options will default (because TRSAVE was not issued) and the trace files will be named IO1 and IO2 with a maximum size of 256 pages and a maximum number of 2.

```
trsource id io1 set ioset type io dev c41-c47 iodata 200
trsource id io2 set ioset type io dev a81-a87 iodata 400 user user1
trsource enable set ioset
trsource disable id io1
trsource disable id io2
```

3. This example defines, enables, and drops two trace IDs of type GT. These trace IDs, GT1 and GT2, are put into trace set GTSET. The destinations are specified as system trace files for userid USER3 with a default size of 256 pages.

```
trsource id gt1 set gtset type gt for user user2
trsource id gt2 set gtset type gt for vmgroup gcsgroup
trsave id gt1 dasd to user3
trsave id gt2 dasd to user3
trsource enable id gt2
trsource enable id gt1
trsource drop set gtset
```

TRSOURCE DISABLE

Privilege Class: C

Use TRSOURCE DISABLE to terminate the tracing of individual trace IDs and of trace IDs within trace sets.

TRSource	DISABle { [ID t1 t2...tx] [SET s1 s2...sy] } [DISPlay] ALL
----------	---

where:

DISABLE

deactivates the specified trace IDs or trace sets. The trace file containing the trace data gathered by the trace ID is closed when the trace ID is disabled.

ID t1 t2 ... tx

specifies the trace ID(s) to be deactivated.

SET s1 s2...sy

specifies the trace set(s) to be deactivated.

ALL

deactivates all existing trace IDs.

DISPLAY

displays the status for each trace ID or trace set specified on this command. If you specify DISPLAY, it must be the last operand on the command line.

TRSOURCE DISPLAY

Privilege Class: C

Use TRSOURCE DISPLAY to display the status of trace IDs and trace sets.

TRSource	DISPly
----------	--------

where:

DISPLAY

displays the status of all defined traces, or indicates there are none. TRSOURCE DISPLAY provides the same information as QUERY TRSOURCE ALL. For more information on these responses, see "QUERY TRSOURCE" on page 423.

Usage Notes

1. To display the status of a particular trace ID or trace set, include the DISPLAY operand at the end of a TRSOURCE command. For example, if you issue the following command:

```
trso id me type gt for user operator display
```

you would receive the following output:

```
ID      TYPE   SET    STATUS  ORIGINID
ME      GT     NULL   DISABLED OPERATOR
```

TRSOURCE DROP

Privilege Class: C

Use TRSOURCE DROP to delete from CP all individual trace IDs and trace IDs within trace sets.

TRSource	DROp	{ [ID t1 t2...tx] [SET s1 s2...sy] } ALL	[DISPlay]
----------	------	---	-----------

where:

DROP

disables the specified trace IDs or trace sets and deletes from CP all the trace IDs or trace sets. The specified trace IDs or trace sets cannot be reenabled without being redefined.

ID t1 t2 ... tx

specifies the trace IDs to be disabled and deleted.

SET s1 s2...sy

specifies the trace sets to be disabled and deleted.

ALL

disables and deletes all existing trace IDs regardless of whether they are enabled.

DISPLAY

displays the status for each trace ID or trace set specified on this command. If you specify DISPLAY, it must be the last operand on the command line.

Usage Notes

1. Use of the DROP keyword directed toward a trace set removes any record of all the trace IDs in the set. To remove or "ungroup" a member of a set, the user should either assign the trace ID to another, possibly unused, trace set name or assign the trace ID to a trace set name of NULL.

TRSOURCE ENABLE

Privilege Class: C

Use TRSOURCE to activate the tracing of individual trace IDs and of trace IDs within trace sets.

TRSource	ENable	$\left\{ \begin{array}{l} \text{[ID t1 t2...tx] [SET s1 s2...sy]} \\ \text{ALL} \end{array} \right\}$	[DISPlay]
----------	--------	---	-----------

where:

ENABLE

activates the specified trace IDs (t1 t2...tx) or trace sets (s1 s2...sy). ALL activates all existing trace IDs.

ID t1 t2 ... tx

specifies the trace IDs to be activated.

SET s1 s2...sy

specifies the trace sets to be activated.

ALL

activates all existing trace IDs.

DISPLAY

displays the status for each trace ID or trace set specified on this command. If you specify DISPLAY, it must be the last operand on the command line.

TRSOURCE ID (Definition Operands)

Privilege Class: C

Use TRSOURCE ID to define trace traps, that is, to associate a tracing condition with a name (trace ID) and, optionally, with a set of other traces (trace set). Also, use the TYPE operand to assign a trace ID to one of the three trace types (I/O, data, or guest tracing).

You do not need to specify all of the definition operands on a particular TRSOURCE command; the definition may be built piecemeal on a number of TRSOURCE commands. Note, however, that the TYPE operand and certain type parameters must be associated with a trace ID before that trace can be enabled.

TRSource	<p>ID traceid [SET {traceset}] [TYPE { IO }] [type parameters] [DISPlay]</p> <p style="margin-left: 20px;"> { NULL }</p> <p style="margin-left: 20px;"> { GT }</p> <p style="margin-left: 20px;"> { DATA }</p> <p>Parameters for Type IO:</p> <p>DEVICE rdev1[-rdev2] [rdev3[-rdev4]] ...</p> <p>USER { userid }</p> <p style="margin-left: 20px;"> { ALL }</p> <p style="margin-left: 20px;"> { SYSTEM }</p> <p>IOData { nnnn }</p> <p style="margin-left: 20px;"> { 0 }</p> <p>Parameters for Type GT:</p> <p>FOR { [USER] userid }</p> <p style="margin-left: 20px;"> { VMGROUP sysname { ALL }</p> <p style="margin-left: 40px;"> { ONLY u1 u2...ux }</p> <p style="margin-left: 40px;"> { EXCLUDE u1 u2...ux }</p> <p style="margin-left: 40px;"> { INCLUDE u1 u2...ux } }</p> <p>Parameters For Type DATA:</p> <p>LOC { address instruction }</p> <p style="margin-left: 20px;"> { modname + disp instruction }</p> <p>DL datalinks</p>
----------	---

where:

ID traceid

is a user-defined, alphanumeric, 1- to 8-character string that identifies the trace. You can specify the trace ID as a target on the ENABLE, DISABLE, and DROP operands to activate, deactivate, or delete a particular trace; you can also use the trace ID as a target for the TRSAVE command, which can alter the recording options for a trace.

Used with the SET operand, trace IDs can be grouped to form trace sets. A trace ID cannot be a member of more than one trace set at a time.

The trace ID also serves as the default name of the system trace file for the trace data.

Trace IDs cannot duplicate any of the following TRSOURCE keywords or any accepted abbreviation of these keywords: ID, SET, TYPE, ENABLE, DISABLE, DROP, DISPLAY, and ALL.

SET { *traceset*
NULL }

is a user-defined, alphanumeric, 1- to 8-character string which identifies a trace ID as a member of a specific trace set. Creating trace sets allows you to Group individual traces that are functionally or logically related. You can then specify *traceset* as the target of the ENABLE, DISABLE, and DROP operands to activate, deactivate, or delete all of the trace IDs in *traceset*.

A trace set can contain trace IDs of more than one type. A trace set of NULL indicates that the trace ID is not a member of any specific set. Set operations cannot be performed on the NULL set.

A trace ID cannot be made a member of a set while the trace ID is enabled. Tracesets cannot duplicate any of the following TRSOURCE keywords or any accepted abbreviation of these keywords: ID, SET, TYPE, ENABLE, DISABLE, DROP, DISPLAY, and ALL.

TYPE { IO
GT
DATA }

is a system-defined trace type. Each trace-type (IO, GT, and DATA) The trace type determines which parameters are allowed and which are required for the specified trace ID. The TYPE operand need only be specified once for a given trace ID but must be specified prior to or concurrently with any type parameters for that trace ID.

Different trace IDs of the same trace type can be placed in the same or different trace sets. Trace types cannot be used as targets of the ENABLE, DISABLE, and DROP operands. Grouping of trace IDs is done with the SET operand, not the TYPE operand.

type parameters

The parameters and descriptions for the respective trace types are documented in the following sections:

- “The I/O Trace”
- “The GT Trace” on page 724
- “The DATA Trace” on page 725.

DISPLAY

displays the status for each trace ID or trace set specified on this command. If you specify DISPLAY, it must be the last operand on the command line.

The I/O Trace

The I/O trace allows you to trace all I/O activity on a user-specified real device, a list of real devices, or range of real devices. The times, channel programs, SCSWs and data transferred for each I/O operation to the specified devices are made available for data analysis.

Note: Any or none of the following I/O parameters may be specified in any order on a TRSOURCE command.

TRSOURCE ID (Definition Operands)

DEVICE	rdev1[-rdev2] [rdev3[-rdev4]] ...
USER	{ userid ALL SYSTEM }
IOData	{ nnnn 0 }

where:

DEVICE rdev1(-rdev2),rdev3(-rdev4)...

specifies the device number or ranges of device numbers whose I/O activity is to be traced. The number of devices or ranges that can be traced with one trace ID of type IO is limited to 255. DEVICE is the only mandatory parameter before enabling of trace IDs of type IO. There are no defined defaults for the DEVICE parameter and any respecification of DEVICE for a previously defined trace results in new devices being appended to the definition. A device can be a member of only one enabled trace ID. Attempts to enable the same device from two or more different trace IDs result in an error message.

USER {
userid
ALL
SYSTEM
}

specifies which user's I/O should be traced. I/O done by the system on behalf of the user is traced. Tracing can be done only for one user per trace ID and any respecification of the user parameter replaces the existing definition. The userid SYSTEM is a valid userid for I/O traces. Only one user's I/O or all I/O is traced. The default value, ALL, specifies that all I/O to the specified device is traced for all userids.

IODATA nnnn

specifies how many bytes, if possible, of the transferred data should be traced. The valid range for nnnn is 0 to 3800 decimal and the default is 0 bytes.

The GT Trace

Traces of type GT allow a guest virtual machine to record events. Guests can use the monitor call class 10 interface to pass a trace event record to CP.

The parameters for type GT are as follows:

FOR	{	[USER] userid	}
	{	VMGROUP sysname	{
		ALL	
		ONLY u1 u2...ux	
		EXCLUDE u1 u2...ux	
		INCLUDE u1 u2...ux	
		}	}

where:

FOR USER userid

specifies the virtual machine can use the monitor call class 10 interface to record trace events.

FOR VMGROUP *sysname*

specifies the virtual machines in the specified VMGROUP can use the monitor call class 10 interface to record trace events.

ALL

specifies that every virtual machine in the VMGROUP *sysname* is included in the defined GT trace. This is the default.

ONLY

specifies that the virtual machines included in the defined GT trace are limited to userids (u1 u2...ux).

EXCLUDE

specifies that every virtual machine in the VMGROUP *sysname* is included in the defined GT trace except those virtual machines with userids u1 u2...ux.

INCLUDE

specifies that userid(s) (u1 u2...ux) should be added to those virtual machines which currently comprise the VMGROUP *sysname* in the defined GT trace.

Parameters may be specified after a VMGROUP trace has been defined to allow further modification of which virtual machines comprise the GT trace. The VMGROUP trace is identified using the ID parameter. These modifications can be entered at any time; the trace specified does not need to be disabled.

The DATA Trace

Trace type DATA allows the user to define trace entries dynamically to record execution of most code paths in CP. The LOC parameter specifies which instruction, when executed, triggers recording of information. Data link parameters define what information is collected when the trace begins.

Note: Any or none of the following DATA parameters may be specified on a TRSOURCE command.

LOC	{ address instruction }
	{ modname + disp instruction }
DL	datalinks

where:

LOC address

specifies the virtual address in the CP nucleus that identifies the trace point at which the data is to be collected.

LOC modname + disp

specifies the CP module name and displacement into that module of the trace point at which the data is to be collected. If the module is written in a higher-level language, then an 8-character module name must be supplied.

Refer to 7 on page 728 for a list of restricted locations.

instruction

is the hexadecimal representation of the contents of storage at the specified trace point. Matching the specified instruction with the contents of storage verifies that the trace point is at the intended location in the CP nucleus. The instruction supplied must be at least as long as the instruction to be verified. Only that instruction is verified and any extraneous hexadecimal data is ignored. If the verification fails then the LOC parameters are not accepted and a message is sent to the user who entered the command.

TRSOURCE ID (Definition Operands)

DL datalinks

specify what data should be traced. There are six character groups that can be used to make up a data link string;

1. Constants: substrings consisting of 0-9, A-F.
2. Registers: substrings beginning with G (for general purpose registers) or X (for control registers) followed by a constant less than decimal 16.
3. Indirection: a percentage sign (%). Indirection means that the contents of an address or register are used instead of the address or register itself.
4. Arithmetic: a plus sign (+) or a minus sign (-).
5. Length delimiter: a period (.).
6. Mnemonic marker: an equals sign (=).

These data link strings define how to combine constants and indirection to reach the data to be traced. The following is a Backus/Naur definition of the syntax of data link strings:

```
<datalink> ::= <main> | <main>=<mnemonic> | NULL
           <main> ::= <anchor> | <anchor><vector> | <anchor>.<length> |
                    <anchor><vector>.<length> | <range>
           <anchor> ::= <reg> | 0...FFFFFF | <anchor>%
           <vector> ::= <modifier> | <vector>% | <vector><modifier>
           <modifier> ::= +<d> | -<d> | +<reg> | -<reg>
           <length> ::= 0...F00
           <range> ::= G<digit>:<digit> | X<digit>:<digit>
           <reg> ::= G<digit> | X<digit>
           <digit> ::= 0...15 | 0...9, A, B, C, D, E, F
           <d> ::= 0...7FFF
           <mnemonic> ::= '...' (user generated 1-to 8-character
                                alphanumeric string)
```

Data can be identified either by using the current register contents as pointers or by knowing the real location of the data to be traced. Data links define the procedure to be used to locate the desired data.

General Usage Notes

The following list contains the syntax rules that apply to the TRSOURCE command:

1. If you use the ID operand to define a trace ID, it must be the first operand on the command line.
2. While specific type-dependent operands are optional on the TRSOURCE command, certain type-dependent operands may be required for a trace ID before it can be enabled. Refer to the description of the specific trace types to

determine which operands are required before a trace ID of that trace type can be enabled.

3. The TYPE operand, if used, must directly precede any type dependent operands.
4. All type-dependent operands must be contiguous. The order depends on the requirements of the trace type.

I/O Trace Usage Notes

1. An I/O trace may be defined for a nonexistent device (that is, a device that has no RDEVICE entry in module HCPRIO). If a trace with *only* such devices is enabled, message HCPTXI477I is issued:

Traceid (TRACE ID) not enabled; incomplete definition

Note, however, that a trace with a nonexistent device and a device defined in HCPRIO (whether the latter be online or offline) *can* be enabled, and the response to QUERY TRSOURCE will indicate that a trace for the nonexistent device is activated.

2. If and when a device is subject to I/O assist, then I/O activity to that device is not traced.
3. System trace files reside on system spool space. Tracing I/O to system spool space causes trace recursion and may cause severe performance degradation, especially for large values of IODATA.

Type GT Usage Notes

1. Specification of a TYPE GT trace for a VMGROUP makes it easier to control and define guest traces for the members of a group.

A trace TYPE GT for a VMGROUP is used to sift TRSOURCE definition and TRSAVE destination information from the VMGROUP to the virtual machines that are members of the VMGROUP. (The *sysname* specified on the VMGROUP parameter is the name of the named saved system which was IPLed by its members.)
2. Multiple TYPE GT traces may be defined for each virtual machine. Only one TYPE GT trace may be enabled for each virtual machine.
3. A TYPE GT USER trace for a virtual machine that has been defined before the virtual machine becomes a member of a VMGROUP for which a TYPE GT trace is enabled is automatically disabled when the virtual machine becomes a member of the VMGROUP. The USER TYPE GT trace remains in its disabled state after the virtual machine either severs from the VMGROUP, is removed from the VMGROUP due to a subsequent TRSOURCE command that excludes the virtual machine userid from the VMGROUP trace, or the VMGROUP trace is disabled. A TRSOURCE ENABLE command is required to re-enable the USER TYPE GT trace.
4. If you need to redefine the destination of the trace data where trace data is to be placed for one of the members of a VMGROUP trace, you must exclude the virtual machine's userid from the VMGROUP trace and then enable a TYPE GT USER trace containing the desired destination information for the virtual machine.
5. A message is sent to the user who defined a USER TYPE GT trace whenever a defined and enabled USER TYPE GT trace is disabled due to VMGROUP TYPE GT trace activity for a VMGROUP member.

6. A message is sent to the user who defined a USER TYPE GT trace whenever a virtual machine that is a member of a VMGROUP trace is excluded from the VMGROUP by a subsequent TRSOURCE command. This user is also notified if the virtual machine was a member of a VMGROUP for which tracing was disabled.

Type DATA Usage Notes

1. If a trace point is defined in a pageable module then the module is automatically locked into storage while the trace is enabled.
2. Data collection for trace IDs of trace type DATA occurs before execution of the overlaid instruction. If no data links are specified for the trace ID, a trace entry is created in the system trace file, but no data is collected.
3. Usually, the total amount of data collected in a trace record cannot exceed 4000 decimal bytes. To calculate the exact amount of data that can be collected in a trace record, use the following formula:

$$\begin{array}{r}
 (3 \text{ times the number of data links specified}) \\
 + \quad (\text{the total lengths of the data links}) \\
 + \quad (\text{the total number of bytes to be collected}) \\
 \hline
 \end{array}$$

SUM

SUM must be less than or equal to 4024 (decimal).

Also, the maximum number of data links allowed for any given trace ID is 255. The maximum data link length is 250.

4. DATA traces should always be placed at instruction boundaries. The TRSOURCE command can only verify that the specified location is on a halfword boundary and that it contains a valid operation code; if these conditions are not met, the command fails with an error message.

If the verification is successful, the TRSOURCE command overlays the specified location with a data trace SVC instruction. If the location is not an instruction boundary, this can cause unpredictable system results, including a probable system failure.
5. As there is no way of deleting a data link from a trace definition, if an error is made while entering a data link the entire trace must be dropped and redefined to remove the data link from the definition. For this reason, it is recommended that the definitions of commonly used type DATA traces be saved in EXECs. Locations can be changed for defined but disabled DATA traces.
6. Specification of extra data links for a defined but disabled trace result in an addition to the definition.
7. The following is a list of restricted locations for the definition of trace points. These locations are restricted because enabling DATA traces in them would cause recursion or interlocks. If an attempt is made to define or enable a trace that violates one of these restrictions then the command is rejected and a message indicating invalid location for tracing is issued.

- a. Locations within the following modules:

HCPABN, HCPBMS, HCPSTK, HCPSVC, HCPTSR, HCPTSV,
 HCPTXD, HCPTXE, HCPTXG, HCPTXI, HCPTXJ, HCPTXL, HCPTXP,
 HCPTXR.

Note: HCPSTK, HCPSVC, HCPABN, and HCPRUN are only partially restricted.

- b. Locations not on halfword boundaries.
 - c. Locations not mapped in CP virtual storage.
 - d. Locations of already enabled DATA traces. Note that multiple DATA traces may be defined at the same location but only one trace at a time may be enabled at that location.
8. In addition to the modules that would definitely cause recursive situations, there exist modules that are invoked after collection of a page worth of data. No restrictions prevent users from placing traces in these modules. However, if the records produced by these traces are one page in length, this causes recursion. Therefore it is recommended that records defined for any code path within system trace file (including paging and spooling code paths and some code paths in IOS) be kept under a page in length.
 9. Users should avoid placing traces in code paths that hold spin locks as this situation increases the path length under the spin lock and seriously decreases system performance.
 10. Users should avoid placing data traces on instructions that are targets of EXECUTE instructions; otherwise, unpredictable results may occur.
 11. If interpretation of a data link string during execution results in a negative or invalid address, a record is added to the system trace file to indicate that the error occurred. However, there will be no external indication that the error occurred.
 12. To trace data located in the absolute page zero, reverse prefixing must be used. When the value of the prefix register is added to the datalink string, values are obtained from absolute page zero. On the other hand, if the address calculated from a data link is less than hexadecimal 1000, the value recorded is obtained from the prefix page of the processor in which the data trace event occurs.

Examples of Data Link Usage: The following examples are based on the status of the registers and the storage pictured below.

<u>Registers</u>		<u>Main Storage</u>				
	+-----+	4000	00005008	00000000	0000A000	00000000
G1	00001000					
	+-----+	5000	0000A004	00001144	00005555	00000000
		6000	00000000	00000000	0000A000	00000000
	+-----+					
G4	00004000				
	+-----+					
		A000	0000B000	0000C000	0000D000	0000E000
	+-----+	B000	C4C1E3C3	E3D9C1D7	11110000	00000000
G5	00005000					
	+-----+	C000	C4C1E3C3	E3D9C1D7	FFFF0000	00000000
		D000	DDDDDDDD	00000000	00000000	00000000
	+-----+					
G6	00006000					
	+-----+					
Ex.	<u>Datalink</u>	<u>Data Collected</u>				
---	-----	-----				
1)	4000	00005008				
2)	4000%	00005555				
3)	5000%.8	0000C000 0000D000				
4)	G5	00005000				
5)	G5.4	0000A004				
6)	G5+4	00001144				
7)	G5+4%	contents of 1144				
8)	G5+G6+8%	contents of 11110000				
9)	G5%.C	C4C1E3C3 E3D9C1D7 FFFF0000				
10)	G4:6	00004000 00005000 00006000				
11)	G6+A.2	A000				
12)	G4+8%+4%.A	C4C1E3C3 E3D9C1D7 FFFF				
13)	G5%+4	0000D000				
14)	G5%+4%	DDDDDDDD				

Note: Registers, when specified alone as in example 4 above, do not default to an indirection and a length of 4, as other data link values do. In this case, the data collected is simply the contents of the register.

UNDEDICATE

Privilege Class: A

Use this command to remove a real processor from a user's exclusive use.

UNDEDicate	[USER {userid} *]	[CPU cpuaddr CPU ALL]
------------	------------------------	----------------------------

where:

USER {userid}
*

is the user whose virtual CPU will be undedicated from a real processor. If the USER operand is specified then either a userid or "*" must be used. If the USER operand is not specified, then the V=R user is the default.

CPU cpuaddr

is the defined virtual CPU address in a virtual machine configuration from which a real processor, if it was dedicated, will become undedicated.

CPU ALL

will undedicate every dedicated real processor from the corresponding virtual CPU in a virtual machine configuration. If a CPU is not specified, CPU ALL is the default.

Usage Notes

1. The default for the UNDEDICATE command is CPU ALL for the V=R userid. An attempt will be made to undedicate every real processor from each virtual CPU defined in the V=R virtual machine configuration.
2. If UNDEDICATE USER *userid* CPU ALL is specified, all the currently defined virtual processors for the userid are affected. Also, if the userid is running a V=R machine with automatic dedication in effect, then the command disables automatic dedication for the V=R machine. To avoid this, use UNDEDICATE USER *userid* CPU *cpuaddr* instead.

Responses

COMMAND COMPLETE

is the response issued upon the successful completion of the UNDEDICATE CPU command having specified a CPU address in the command line.

UNLOCK**Privilege Class: A**

Use the UNLOCK command to unlock page frames previously locked by a LOCK command.

UNLOCK	<pre> userid page1 pagen * page1 pagen SYSTEM page1 pagen symbol VIRT=REAL V=R RIO370 </pre>
--------	---

where:

userid

is the user identification of the virtual machine previously specified in a CP LOCK command. If you are unlocking your own virtual pages, you may specify userid as an asterisk (*).

SYSTEM

unlocks a previously locked pageable CP nucleus page.

page1

is the hexadecimal value of the first page to be unlocked.

pagen

is the hexadecimal value of the last page to be unlocked.

symbol

causes the page containing the system symbol to be unlocked, if the symbol is contained within the CP pageable area. No action is taken if the symbol is contained within the CP resident nucleus or if the page has not been locked by the CP LOCK command.

VIRT = REAL**V = R**

releases for normal paging use all the page frames normally reserved for preferred virtual machines if the area is not occupied by a preferred virtual machine. Once you release the virtual = real area, you can not reactivate it to run another preferred virtual machine until you re-IPL VM/XA SP.

RIO370

requests the release, for normal paging use, of the RIO370 frames reserved at system generation for DIAGNOSE X'98'. (Only programs running in 24-bit addressing mode can use RIO370 frames.) If any RIO370 frames are in use, the area is not released. Once released, the RIO370 area can be reinstated only by re-IPLing VM/XA SP.

Usage Notes

1. Only specify the page numbers for the fields “page1” and “pagen.” For example, to unlock USERA’s virtual storage locations X'12000' to X'2CFFF', issue the following command:
UNLOCK USERA 12 2C
2. Exercise caution when you issue the UNLOCK symbol command. The page containing the symbol may also contain other symbols that have been processed using the CP LOCK command. A single UNLOCK command unlocks all of the symbols locked in that page, even if you locked them using multiple CP LOCK commands.
3. Unlocking V=R page frames unlocks the entire V=R region. This area is unlocked only if no V=R or V=F guests are logged on.
4. Use the QUERY FRAMES command to determine the status of the RIO370 storage area. If (a) the LOCKRIO count is zero, (b) your program is running in 24-bit addressing mode, and (c) further lock requests are not issued, you can issue UNLOCK RIO370.
5. Pages locked by programs running in 31-bit addressing mode using DIAGNOSE code X'98' do not use RIO370 frames, so that real I/O is available to 31-bit addressing mode programs whether or not the RIO370 area exists.
6. The LOCK command issued for a userid’s virtual page or pages operates independently of the DIAGNOSE X'98' page locking facility. Pages locked by command cannot be unlocked by DIAGNOSE X'98', nor can REAL I/O pages locked by DIAGNOSE X'98' be unlocked by command.
7. If you issue LOCK to lock a page in a shared segment and do not issue an UNLOCK, the page is unlocked when the last user of the shared segment releases the storage with the LOGOFF command or another command that clears storage.
8. If the UNLOCK command is issued for an address range that includes all or part of the RIO370 area, the command does not execute for those frames. No message or response is issued to indicate this. The RIO370 frames remain locked, and possibly in use, as they were before the command was issued. To unlock this storage, use the RIO370 option of the UNLOCK command when the entire RIO370 area is not in use.

Responses

COMMAND COMPLETE

indicates the successful completion of the command. This response is displayed even if you did not have any pages locked in storage.

Migration Notes

VM/SP HPO

1. In VM/XA SP, you can specify a CP system symbol to unlock the page containing that symbol.
2. In VM/XA SP, the RIO370 parameter on the SYSSTORE macro is used to create the RIO370 area.

VARY

Privilege Class: B

Use the VARY command to make a device or processor usable or unusable by the control program or a user.

VARY	$\left. \begin{array}{l} \{ \text{ONline} \} \\ \{ \text{OFFline} \} \end{array} \right\} \left. \begin{array}{l} \text{rdev} \\ \text{rdev rdev ...} \\ \text{rdev-rdev} \\ \text{PROCESSOR nn} \end{array} \right\}$
------	--

where:

ONLINE

makes available a device or devices previously offline.

OFFLINE

makes unavailable the device or devices.

rdev

rdev rdev ...

rdev-rdev

is the real device number of the device to be varied online or offline. You may specify a list or range of devices. If you specify a range of real device numbers, all valid devices in the range, including the first and last devices you specified, will be varied online or offline. The maximum number of devices allowed in a range is 256.

PROCESSOR nn

is the processor to be made available or unavailable. The value of nn is the processor address (hexadecimal value from 00 through 3F) established when the processor is installed.

Usage Notes

1. If the device you specify is a 2305, the address you indicate must be for the first of eight "exposures;" that is address xxx0 or xxx8. If the device you specify is a 3350 Model 11 or Model 21, the address you indicate must be the first address (or base path address) of its multi-exposure addresses.
2. If you specify a list or range of devices, an attempt is made to VARY them online or offline regardless of whether an error is encountered with any one of the specified devices. An error message is issued for every device that cannot be varied online or offline. A summary message is issued to indicate the number of devices that were successfully varied online or offline.
3. The VARY PROCESSOR command will not affect the connection of a Vector Facility associated with the varied real processor. If the associated Vector Facility is not connected at the time a VARY ONLINE PROCESSOR command is issued, it will still not be connected after the VARY command has completed. Conversely, if a Vector Facility is connected at the time that a VARY OFFLINE PROCESSOR command is issued, it will remain connected to its associated processor even after that processor is fully varied offline.

4. When a VARY PROCESSOR or VARY VECTOR command affects a processor dedicated to a preferred-guest virtual CPU, rededication is attempted to:

- Keep real processors with Vector Facilities dedicated to preferred-guest virtual CPUs with Vector Facilities defined
- Keep real processors without Vector Facilities dedicated to preferred-guest virtual CPUs with no Vector Facilities defined.

Dedication will occur whether or not the above may be accomplished.

5. When a processor is successfully varied online to a system with a V=R virtual machine, dedication will occur to an available virtual CPU if automatic dedication is enabled. The above goals are attempted (that is, match real VF to virtual VF or no real VF to no virtual VF); however, dedication will occur whether they can be met.

6. If a virtual CPU with a Vector Facility defined is left without a real Vector Facility available, subsequent vector instructions will result in one of two guest interruptions:

- A vector operation exception is presented to those guests who had not been previously informed by CP that a real Vector Facility was available for their use
- A Vector Facility failure machine check is presented to those guests who had been informed by CP that a real Vector Facility was available for their use.

These situations can occur as a result of VARY PROCESSOR, VARY VECTOR, or changes in dedication. The system operator will always receive a warning message when such a circumstance arises.

Responses

rdev VARIED { ONLINE }
 { OFFLINE }

indicates that the specified device is now online or offline as you requested in the command.

rdev-rdev VARIED { ONLINE }
 { OFFLINE }

indicates that all exposures of a multiple exposure device are now online or offline as requested in the command.

type rdev IS ALREADY { ONLINE }
 { OFFLINE }

indicates that the specified device is already online or offline.

rdev-rdev ALREADY { ONLINE }
 { OFFLINE }

indicates that the specified multiple exposure device is already online or offline.

nnn DEVICE(S) SPECIFIED; nnn DEVICE(S) SUCCESSFULLY VARIED { ONLINE }
 { OFFLINE }

is a summary response indicating how many devices were successfully varied online or offline.

VARY

PROCESSOR nn ONLINE

indicates that the specified processor is now online and operational.

PROCESSOR nn OFFLINE

indicates that the specified processor is now offline and not operational.

VECTOR FACILITY nn ONLINE

indicates that the Vector Facility nn is now available or online.

VECTOR FACILITY nn OFFLINE

indicates that the Vector Facility nn is now unavailable for use or offline.

Migration Notes

VM/SP HPO

1. VM/XA SP does not support the FORCE, VPHY, and VLOG options.
2. The VM/XA SP response provides an additional line indicating the number of devices successfully varied on line.
3. VM/XA SP does not support logical printers.
4. VM/XA SP limits the range of devices varied to 256.
5. VM/XA SP allows you to spell out the entire operand PROCESSOR.

VARY PATH

Privilege Class: B

Use the VARY PATH command to make a path to one or more devices logically available or logically unavailable for use by the system.

VARY	<pre>{ONline } PATH yy [TO] { rdev rdev rdev rdev ... rdev-rdev ALL }</pre> <pre>{Offline} PATH yy [FROM] { rdev rdev rdev rdev ... rdev-rdev ALL }</pre>
------	--

where:

yy

is the channel path ID of the path that is to become logically available or logically unavailable for use by the system to communicate with the specified device or devices.

Note: If the specified path is the last available path to the specified device, CP will attempt to vary the device offline. If the device cannot be varied offline, the VARY PATH OFFLINE request will be denied.

rdev

rdev rdev ...

rdev-rdev

is the real device number of the devices to whom the specified path will be made logically available or logically unavailable. You may specify a single device, a list, or a range of devices. If you specify a range of real device numbers, all the valid devices in the range, including the first and last devices you specified, will be processed. The maximum number of devices allowed in a range is 256.

ALL

specifies that the path will be made logically available or logically unavailable to all real devices that have access to it.

Usage Notes

1. The VARY PATH command does not invoke the service processor or cause any hardware functions to be performed.
2. If you specify a list or range of devices, an attempt is made to vary the specified path online or offline to each device. An error message is issued for every device for which the specified path cannot be varied online or offline. Check examples after these usage notes.

VARY PATH

3. **Warning:** Use care when varying offline a path leading to a 370/XA guest's dedicated device. If the 370-XA guest is using RESERVE/RELEASE to that device, you may vary its reserved path offline, causing that system to become hung.

To determine if RESERVE/RELEASE is allowable, look at the SHARED option on the RDEVICE statement of the HCPRIO block or issue the QUERY DASD command and look at that SHARED setting.

4. When varying a path offline:
 - Vary the path offline from the guest operating system if the guest (such as MVS/XA) supports a VARY PATH command.
 - Vary the path offline from the host system (VM/XA SP).
5. When varying a path online:
 - Make the path physically available.
 - Vary the path online to the host (VM/XA SP).
 - Vary the path online to the guest operating system if the guest supports a VARY PATH command.

Examples

Enter the following command to make path C2 leading to device 140 available to that device.

```
VARY ON PATH C2 TO 140
```

A typical response may appear as follows:

```
VARY PATH C2 ONLINE COMMAND INITIATED  
VARY PATH C2 ONLINE COMMAND COMPLETE
```

Enter the following command to make path C2 unavailable to devices 0190 through 0197.

```
VARY OFF PATH C2 FROM 0190-0197
```

A typical response may appear as follows:

```
VARY PATH C2 OFFLINE COMMAND INITIATED  
PATH C2 NOT VARIED OFFLINE FROM DEVICE 0192; LAST PATH TO DEVICE  
VARY PATH C2 OFFLINE COMMAND COMPLETE
```

Response 1: Vary Initiated

```
VARY PATH yy {ONLINE } COMMAND INITIATED  
              {OFFLINE }
```

is the result of issuing the VARY PATH command and shows that the specified VARY PATH request has been initiated.

where:

yy

is the path requested to be varied online or offline.

Response 2: Vary Complete

```
VARY PATH yy {ONLINE } COMMAND COMPLETE
              {OFFLINE }
```

indicates that the VARY PATH processing is complete. An attempt was made to make the specified path available or unavailable (whichever was specified – online or offline) to each device specified. While command processing is now complete, check carefully any console messages which might indicate that not all processing was successful.

where:

yy

is the path requested to be varied online or offline.

Response 3: Path Not Operational

```
PATH yy NOT VARIED ONLINE TO DEVICE rdev; PATH NOT OPERATIONAL
```

indicates that the specified path has not been made available (varied online) for the specified device because the path is not operational. Make the path operational and then retry the command.

where:

yy

is the path not made operational.

rdev

is the real device number.

Response 4: Last Path to Device

```
PATH yy NOT VARIED OFFLINE FROM DEVICE rdev; LAST PATH TO DEVICE
```

indicates that the specified logical path is the last path to the specified device. CP tried unsuccessfully to vary the device offline. Use the VARY command to vary the device offline, and then retry the VARY PATH command.

where:

yy

is the path which CP tried to make unavailable (vary offline) to the specified device.

rdev

is the real device number.

VARY PATH

Response 5: Invalid Channel Path ID

PATH yy NOT VARIED { ONLINE TO } DEVICE rdev; PATH DOES NOT EXIST TO THE DEVICE
 { OFFLINE FROM }

indicates that the specified path was not installed for the specified device.

where:

yy

is the path requested to be made logically available (varied online) or unavailable (varied offline) to the specified device.

rdev

is the real device number.

Response 6: Device Reserve Held

PATH yy NOT VARIED ONLINE FROM DEVICE rdev; RESERVED PATH

indicates that the specified path was not taken offline because there is a reserved allegiance to that path.

where:

yy

is the number for the path not taken offline.

rdev

is the real device number.

Response 7: First Path:

FIRST PATH yy TO type rdev VARIED ONLINE; type rdev VARIED ONLINE

indicates that the device was previously offline because all of its channel paths have been varied offline. The path requested to be varied online was the first path to the device to be varied online, and therefore the device as well as the path was varied online.

where:

yy

is the path that was requested to be made available (varied online) to the specified device.

type

is the device type (for example, DASD or TAPE).

rdev

is the real device number.

Response 8: Last Path:

LAST PATH yy TO type rdev VARIED OFFLINE; type rdev VARIED OFFLINE

indicates that the path requested to be varied offline was the last path to the device and therefore the device (and the path) were varied offline.

where:

yy

is the path that was requested to be made unavailable (varied offline) to the specified device.

type

is the device type (for example, DASD or TAPE).

rdev

is the real device number.

Response 9: No Online Paths to Device:

type rdev CANNOT BE VARIED ONLINE; DEVICE HAS NO ONLINE CHANNEL PATHS

indicates that the device specified to be varied online has no online (logically available) channel paths.

where:

type

is the device type (for example, DASD or TAPE).

rdev

is the real device number.

Response 10: Pending Reserve

PATH yy NOT VARIED OFFLINE FROM DEVICE rdev; DEVICE RESERVE PENDING

indicates that the path requested to be varied offline could not be varied offline because a reserve request was pending to that device. Check the other system sharing this device; it may be hung, holding the reserve, thus prohibiting CP's reserve request from being granted.

where:

yy

is the number of the path not varied offline because of a pending reserve request.

rdev

is the real device number.

VARY PATH

Response 11: Path Not Physically Available

PATH yy NOT VARIED { ONLINE TO } DEVICE rdev; PATH NOT AVAILABLE TO THE DEVICE
 { OFFLINE FROM }

indicates that the path is not physically available to the device. Ensure that the path is switched in and varied online from the system's service console.

where:

yy
is the number of the path not physically available to the device.

rdev
is the real device number.

Response 12: Path Already Varied Offline

PATH yy HAS ALREADY BEEN VARIED OFFLINE FOR DEVICE rdev

indicates that the system operator has already varied the path offline.

yy
is the path that was both requested to be and then was varied offline.

rdev
is the real device number.

Response 13: Path Already Online

PATH yy ALREADY ONLINE FOR DEVICE rdev

indicates that the path is already online.

yy
is the number of the path already online.

rdev
is the real device number.

VARY VECTOR

Privilege Class: B

Use the VARY VECTOR command to make a device previously online or offline either available or unavailable.

This command does not change the status of the associated CPU. If the CPU is offline, the Vector Facility cannot be brought online.

Similarly, the VARY PROCESSORS command does not change the connection status of the vector facility. If the facility was connected to the CPU, it becomes available for use after the CPU has been configured; if it was not connected, it remains in that state after the command has been executed.

VARY	$\left. \begin{array}{l} \text{ONline} \\ \text{OFFline} \end{array} \right\} \text{VECTOR nn}$
------	---

where:

ONLINE

makes a device(s) previously offline available.

OFFLINE

makes the device(s) unavailable.

Vector nn

is the Vector Facility associated with processor nn.

Usage Notes

1. A Vector Facility (VF) is permanently associated with single CPU and can be used only by the associated CPU. The address of the Vector Facility is the same as that of the associated CPU.
2. A Vector Facility that is connected to a real dedicated processor may be varied offline. If the virtual CPU dedicated to the processor with the varied offline VF already has a virtual VF, subsequent use of the virtual VF will result in virtual VF failure machine checks. If an attempt is made to vary off a dedicated VF, the system will issue an error message.
3. If the last nondedicated Vector Facility is being taken by a VARY OFFLINE command, the system will issue a warning message. Any V=V guest which has a virtual VF defined when the last real VF becomes unavailable will subsequently receive a VF failure machine check if it issues any vector instructions. V=V guests which define virtual Vector Facilities after the last real vector has gone away will receive a warning message.

VARY VECTOR

Response 1

VECTOR FACILITY nn OFFLINE

if the Vector Facility is now unavailable.

Response 2

VECTOR FACILITY ONLINE

if the Vector Facility is now available.

VMDUMP

Privilege Class: G

Use the VMDUMP command to produce a dump of all or selected pages of the storage that appears real to your virtual machine (second-level storage). The resulting dump is in a format that can be used by the dump viewing facility. The dump will include selected information for up to eight CPUs in your configuration.

VMDump	$\left[\begin{array}{l} \left[\text{hexloc1} \right] \\ \left[\underline{0} \right] \end{array} \right] \left[\begin{array}{l} \{-\} \\ \{:\} \end{array} \right] \left[\begin{array}{l} \text{hexloc2} \\ \text{END} \end{array} \right] \left[\begin{array}{l} \{.\} \\ \text{bytecount} \\ \text{END} \end{array} \right] \left[\begin{array}{l} \text{TO } * \\ \text{TO userid} \\ \text{SYSTEM} \end{array} \right]$
	$\left[\text{FORMAT } \left\{ \begin{array}{l} \text{vmtype} \\ \text{FILE} \end{array} \right\} \right] \left[\begin{array}{l} \text{DCSS} \\ \text{DSS} \end{array} \right] \left[\text{XA} \right] \left[*dumpid \right]$

Note: You can specify multiple address ranges on a single VMDUMP command. The ranges can be in any order. However, you can assign only one dump identifier (**dumpid*) per command.

where:

hexloc1

0

is the starting virtual storage location to be dumped. The address hexloc1 may be one to eight hexadecimal digits; leading zeros are optional. If the value you specify for hexloc1 is not a multiple of 4K, VM/XA SP rounds down the value to a 4K boundary. If you omit the hexloc1 operand, the default is zero.

{-} hexloc2
{:} END

is the ending address in the range of virtual storage locations to be dumped. The address hexloc2 may be one to eight hexadecimal digits; leading zeros are optional. The value you specify for hexloc2 must be equal to or greater than hexloc1, and within the virtual storage size defined for your virtual machine. If the value you specify for hexloc2 is equal to the starting location, VM/XA SP dumps the single page of storage containing that location. If the value you specify for hexloc2 is not equal to the starting location and is not a multiple of 4K, VM/XA SP rounds up the value to the next 4K boundary. Specify END to dump the remaining contents of storage.

If you specify only a range delimiter ('-' or ':') and omit either hexloc2 or END, END is the default. However, if you omit both the range delimiter and hexloc2 or END, VM/XA SP dumps the single page of storage starting at hexloc1. If you omit both hexloc1 and hexloc2 or END, hexloc1 defaults to zero, hexloc2 defaults to END, and VM/XA SP dumps the entire contents of your virtual machine's storage. (Note that in this case specification of the range delimiter is irrelevant.)

{.} bytecount
END

is a hexadecimal number indicating the number of bytes of storage to be dumped, starting with the first byte at hexloc1. The value you specify for bytecount must be at least one and can be from one to eight hexadecimal digits; leading zeros are optional. The ending storage address resulting from the

specified bytecount must be within the virtual storage size defined for your virtual machine. If the ending storage address is not a multiple of 4K, VM/XA SP rounds up the value to the next 4K boundary. Specify END to dump the remaining contents of storage.

If you specify only the range delimiter (‘.’) and omit either bytecount or END, END is the default. However, if you omit both the range delimiter and bytecount or END, VM/XA SP dumps the single page of storage starting at hexloc1. If you omit both hexloc1 and bytecount or END, hexloc1 defaults to zero, bytecount defaults to END, and VM/XA SP dumps the entire contents of your virtual machine’s storage. (Note that in this case specification of the range delimiter is irrelevant.)

TO *

TO userid

transfers the dump to the virtual card reader of the specified userid. If you enter an asterisk (*) after TO, or do not specify the TO operand at all, VM/XA SP sends the dump to the virtual card reader of the virtual machine from which you issued the VMDUMP command.

SYSTEM

transfers the dump to the virtual card reader of the userid specified on the SYSDUMP macro in HCPSYS. (This is the userid designated to receive any CP abend dumps. For further details, refer to the description of the SYSDUMP macro in *VM/XA SP Planning and Administration*.)

FORMAT { vmtyp }
{ FILE }

specifies the type of virtual machine being dumped. For vmtyp, specify the 1- to 8-character name of the system that had been running in the virtual machine (for example, CMS, RSCSNET, or PVM). If you omit the FORMAT operand, the default is FILE.

The dump viewing facility uses the FORMAT operand to identify what type of system or virtual machine was dumped.

DCSS

DSS

specifies that CP take a dump of all the user’s discontinuous saved segments.

XA

specifies that the dump produced should be in the VM/XA SP dump format. If you do not specify XA, VM/XA SP uses the mode, the storage size of your virtual machine, and the number of virtual CPUs defined to determine the format of the dump. For a virtual machine running in System/370 uniprocessor mode with 16 megabytes or less of storage, VM/XA SP produces a dump in the same format as the VM/SP (370) VMDUMP command. For all other virtual machines (System/370 mode virtual machines with more than 16 megabytes of storage, all 370-XA mode machines, and all virtual MP or AP machines), the dump produced is in the VM/XA SP dump format.

***dumpid**

is a 1- to 100-character identifier (including any imbedded blanks and special characters) that VM/XA SP assigns to the dump. When specifying the dumpid, you must precede it with an asterisk (*). Note that VM/XA SP eliminates any blanks between the asterisk and the first nonblank character of the dumpid. If you specify a dumpid that is more than 100 characters in length *after* the blanks following the asterisk are eliminated, VM/XA SP truncates it on the right to a

length of 100 characters. If specified, the dumpid must be the last operand on the command line.

Usage Notes

1. With the exception of *dumpid, you may specify the operands in any order. If you specify *dumpid, it must be the last operand on the command line.
2. When you issue the VMDUMP command, the dump that results consists of 4K-byte pages of storage written to a reader spool file. You must dump at least one 4K-byte page. In addition, VM/XA SP dumps the following information (not necessarily in this order):
 - Program status word (PSW)
 - General registers
 - Control registers
 - Floating point registers
 - Vector registers
 - Storage keys of dumped storage pages
 - TOD clock
 - TOD clock comparator
 - CPU timer
 - Prefix register
 - Prefix page.

You can view the dump interactively using the dump viewing facility. For more information on viewing the dump produced by the VMDUMP command, see *VM/XA SP Dump Viewing Facility Operation Guide and Reference*.

3. For virtual machines running in System/370 uniprocessor mode with 16 megabytes or less of virtual storage, the VM/XA SP VMDUMP command provides full compatibility with the VM/SP (370) VMDUMP command. For these virtual machines, the dump produced by the VMDUMP command can be either in the format of a VM/XA SP CP abend or stand-alone dump, or in the format produced by the VM/SP (370) VMDUMP command. For virtual machines running in System/370 mode with *more* than 16 megabytes of virtual storage, for all 370-XA mode virtual machines, and for all virtual MP or AP machines, the VMDUMP command produces only a dump in VM/XA SP dump format. For more information, see the description of the XA operand above. For a description of the VM/SP (370) VMDUMP format, see *VM/XA SP Diagnosis Reference - CP (Volume 1)*.
4. After VM/XA SP has written the dump to a reader spool file, you can use the CMS DUMPLOAD command to load it from the spool file into a CMS file. You can also use the DUMPLOAD command to print a hexadecimal dump (with EBCDIC translation) from a virtual machine dump that has already been loaded into a CMS file. For a complete description of the DUMPLOAD command, see *VM/XA SP CMS Command Reference*.
5. In general, use the VMDUMP command to create dump files that can be used with the VM/XA SP dump viewing facility, and use the DUMP command to create dumps that you intend to read without the aid of the dump viewing facility. For more information, see *VM/XA SP Virtual Machine Operation*.

VMDUMP

6. For certain programs running in virtual machines, the dump viewing facility provides a special dump processing format. The dump viewing facility identifies these programs by checking the filetype of the reader spool file created by VMDUMP. The filetype is an assigned name that you should specify for vmtpe on the FORMAT operand when dumping a virtual machine in which one of these programs is running. The IBM-defined programs and their assigned names are:

<u>Program:</u>	<u>Assigned name:</u>
Remote Spooling Communication Subsystem (RSCS) Networking Program Product	RSCSNET
VM/Pass-Through Facility Program Product	PVM
Conversational monitor system (CMS)	CMS

There may be additional assigned names, unique to your installation, for programs that are a part of your system. Ask your system programmer for a complete list of FORMAT names.

7. If you specify TO SYSTEM, VM/XA SP sends the dump file to the userid of SYSTEM. However, if you specify only SYSTEM, VM/XA SP sends the dump file to the userid designated at your installation to receive dumps.
8. Vector registers are dumped for XA format dumps only. You can ensure that the VMDUMP you create will be XA format and will include the vector registers (if any) by specifying "XA" on the command line.

Responses

COMMAND COMPLETE

is the response issued by VM/XA SP when it has completed VMDUMP command processing.

Migration Notes

VM/SP HPO: VM/XA SP supports the additional operands DCSS and XA.

WARNING

Privilege Classes: A, B, or C

Use the **WARNING** command to transmit high priority messages to a specified user or to all users. The **WARNING** does not wait for input operations at the terminal to complete, but breaks in on the line currently being displayed and sends the message immediately (unless the **CP SET WNG OFF** command was previously issued for the virtual machine).

Warning Wng	$\left. \begin{array}{l} \text{userid} \\ * \\ \text{OPERATOR} \\ \text{ALL} \end{array} \right\} \text{msgtext}$
----------------	---

Note: Only class A and B users may issue the **ALL** operand.

where:

userid

OPERATOR

ALL

is the identification of the user to whom you are sending the warning message. If you are sending the warning to yourself, you can specify **userid** as an asterisk (*).

Specify **OPERATOR** if you want to send a warning to the primary system operator, regardless of his or her **userid**. If you specify **ALL**, you broadcast the warning message to all users receiving warning messages.

msgtext

is the text of the message to be sent. The length of the message is limited by the number of characters remaining on the input line after the command and appropriate operand are entered.

Usage Notes

1. **WARNING** commands ring the alarm if the **ALARM** feature is installed, and put the screen in a **HOLD** status if the screen is full. When this occurs, press the **PA2** key or the **CLEAR** key to clear the screen and put the screen in a **RUNNING** status.
2. If a warning is issued from a **CMS** environment, the command and the message text cannot be longer than 240 characters.

If a warning is issued from a **CP** environment, the command and message length is limited by the input area of the terminal.
3. The **WARNING** command will route the message to the secondary user of a disconnected virtual machine. If the secondary user is connected or the secondary user is disconnected but has a valid path to a ***MSG** or ***MSGALL** **CP** system service, they will receive the message. If the secondary user is disconnected and there is no valid path to a ***MSG** or ***MSGALL** **CP** system service the message 057I **userid NOT RECEIVING, DISCONNECTED** is sent to the issuer of the command.

WARNING

Responses

hh:mm:dd * WNG FROM userid: msgtext

is the warning message received by the specified users. The hh:mm:dd represents the time when the message was sent to the user.

Migration Notes

VM/SP HPO: VM/XA SP provides the time stamp and warning text on the same line without blank lines in between as in VM/SP HPO.

XAUTOLOG

Privilege Class: A, B, G

Use the XAUTOLOG command to log on another user's virtual machine. If you have class G privileges (and not class A or B), users whose virtual machines you want to log on must have the appropriate XAUTOLOG statement in their VM/XA SP directory entries. If you have class A or B privileges and you are not specifying the userids' passwords, you are automatically authorized to log on their virtual machines.

XAUTOLog	userid [Ipl { vdev } { sysname }] [#console-input-data] [NOipl] [ACcount nnnnnnnn] [Storage { nnnnnnnK } { nnnM }] [PASSword password] [PRompt] [DIRectory] [SYNch]
----------	--

where:

userid

is the user identification of the virtual machine you are logging on.

IPL { vdev } { sysname }

specifies the virtual device number or named saved system that you want IPLed after logon processing. This option overrides the IPL (if any) specified in the user's entry in the system directory.

If SYNCH is not specified, the IPL command processor determines whether the IPL operand is valid after the logon is completed; it is not verified as part of LOGON or XAUTOLOG processing.

If SYNCH is specified with IPL, the IPL operand is checked for validity before control returns to the XAUTOLOG command issuer.

NOIPL

specifies that the IPL device specified in the user's entry in the system directory should not be used for an automatic IPL.

ACCOUNT nnnnnnnn

specifies an alternate account number to be assigned to the user's virtual machine at logon time. This account number must be valid in order for the logon to complete successfully. If you do not specify this option, the primary (or first) account number specified in the user's entry in the system directory is used.

STORAGE { nnnnnnnK }
 { nnnM }

specifies an alternate storage size to be assigned to the user's virtual machine. The value that you specify for this option must be valid in order for the logon to complete successfully. If you do not specify a value, the minimum storage size indicated in the user's entry in the system directory is used.

The minimum value that you can specify is 64K; the maximum value allowed is the maximum value specified on the USER statement for the user's virtual machine in the system directory and cannot exceed 999M.

If the value that you specify is not a multiple of 64K, and is less than 16M, it is rounded up to the next higher 64K boundary. If the value is greater than 16M, it is rounded up to the next 1M boundary.

PASSWORD password

is the user's logon password. The ability to specify the password is only available to class A and B users. Class A and B users are authorized to automatically log on any virtual machine without specifying a password, but this PASSWORD option is available when password verification is desired. If your installation is using the password suppression facility and you want to specify a password, use the PROMPT parameter; the PASSWORD parameter is not accepted.

PROMPT

specifies that you should be prompted for the password at the terminal. If your installation has password suppression active, then you must specify PROMPT in order to use password authorization when entering the command at the terminal. The ability to specify PROMPT is only available to class A and B users.

DIRECTORY

indicates that the user's entry in the VM/XA SP directory must contain an XAUTOLOG statement to authorize you to log on the virtual machine if you do not have class A or B privileges.

SYNCH

indicates that the command is to be executed synchronously. VM returns control to the issuer of the XAUTOLOG command when the target userid is logged on and IPL command is validated (if an automatic IPL is performed). Unless NOIPL is specified, there must be an IPL statement in the target userid's directory entry or the IPL parameter must be specified on the XAUTOLOG command. (Note: If NOIPL is specified, then VM returns control when the target userid is logged on.)

If SYNCH is not specified, the XAUTOLOG command executes asynchronously and VM returns control after a LOGON is scheduled. The XAUTOLOG command issuer receives a message when the target user is logged on. For asynchronous execution, no IPL statement or parameter is required.

#console-input-data

is data that you want used as virtual console input in response to the first read issued by the user's virtual machine. You may use the remainder of the command line to enter console data; the entire line including any logical line-end characters (#) is passed to the virtual machine.

Note: If you are using CMS and specify console-input-data on the XAUTOLOG command line, CMS interprets the input line as two separate commands because of the logical-line-end character (#). You can avoid this situation by preceding the logical-line-end character with a

logical-escape-character or by preceding the XAUTOLOG command line with #CP, where # is your logical-line-end character.

Usage Notes

1. XAUTOLOG was a synonym for the AUTOLOG command. It was provided to ease future migration for those who wanted to use the extended options of the VM/XA SP XAUTOLOG command. In VM/XA SP Release 2, the AUTOLOG command is changed to be compatible with the VM/SP HPO AUTOLOG command.

2. If neither IPL nor NOIPL is specified, the virtual machine that you log on using the XAUTOLOG command must have an IPL statement in its system directory entry.

Unless NOIPL is specified, an automatic IPL is scheduled. If the IPL statement is invalid, you receive a message and the userid is forced off the system.

3. Authorization to log on a user via the XAUTOLOG command is granted:

- If you have class A or B privileges and you specify or default to the DIRECTORY option
- If you have class A or B command privileges and you specify a correct password via a prompt or the PASSWORD option
- If you have class G (and not class A or B) command privileges and your userid appears in the XAUTOLOG directory statement belonging to the virtual machine that you want to XAUTOLOG.

4. Users with a password of NOLOG in their directory entries cannot be logged on by the XAUTOLOG command.

For a complete description of the XAUTOLOG directory statement, see *VM/XA SP Planning and Administration*.

5. The virtual machine that you log on using the XAUTOLOG command operates in disconnect mode. The same restraints that apply to any disconnected machine also apply to virtual machines that are logged on with the XAUTOLOG command.
6. If the userid you specify has a secondary userid specified on the CONSOLE statement in its directory entry, messages from CP (and from the virtual machine if automatic IPL takes place) are routed to the secondary user if that user is logged on and not disconnected. If your userid (XAUTOLOG issuer) is the designated secondary userid and you are not running in disconnect mode, you receive messages resulting from the LOGON of the target user.
7. If your installation is using the password suppression facility (controlled by the SET PASSWORD command) and you want to use password authorization, you should enter the PROMPT parameter on the XAUTOLOG command line. Wait for the "ENTER PASSWORD" prompt before you type in the password. The password is automatically masked, thus improving system security.
8. VM/XA SP disables the XAUTOLOG and AUTOLOG commands for the rest of a user's logon session if the user enters excessive incorrect passwords to log on another userid automatically. The initial system limit is four incorrect passwords. Use the QUERY CMDLIMIT command to see the system limit value set by your installation.

The PASSWORD parameter may be used on an XAUTOLOG command in an EXEC or module (even if your installation is using the password suppression facility) unless you set the DIAGNOSE X'08' control bit to reject a password

Response 4

hh:mm:ss USER DISC LOGOFF AS userid USERS = nnnnn FORCED

is the response to the system operator upon failure of the IPL process for the user specified on the XAUTOLOG command. The system operator receives this response only if SYNCH is specified and an automatic IPL was scheduled for the userid.

where:

DISC

indicates that the user you autologged is operating in disconnect mode.

userid

is the identification of the virtual machine initiated by the XAUTOLOG command.

USERS = nnnnn

is the number of logged-on users.

Response 5

HCPmmmm6056I XAUTOLOG information for userid : The IPL
command is verified by the IPL command processor.

is the response to the issuer of the XAUTOLOG command when the IPL statement has been verified by the IPL processor for the user specified in the XAUTOLOG command. This response is issued when XAUTOLOG is asynchronous (the SYNCH parameter is not specified).

where:

userid

is the identification of the virtual machine initiated by the XAUTOLOG command.

Response 6

USER DSC LOGOFF AS userid USERS = nnnnn FORCED

is the response to the system operator if an invalid IPL statement is processed for the user specified on the XAUTOLOG command. The system operator receives this response only if an automatic IPL was scheduled for the userid.

where:

userid

is the identification of the virtual machine initiated by the XAUTOLOG command.

USERS = nnnnn

is the number of logged-on users.

Migration Notes

VM/SP HPO

1. No migration to the XAUTOLOG command is required.
2. To remove the password from AUTOLOG commands issued from EXECs and programs, convert those commands to XAUTOLOG commands (for class A and class B users).
3. In addition, class G users may be authorized to use the XAUTOLOG command for a given virtual machine if they are listed on an XAUTOLOG statement in the virtual machine's system directory entry.

VM/XA SP Release 1

1. The XAUTOLOG command replaces the VM/XA SP Release 1 AUTOLOG command. It supplies all the functions of the VM/XA SP Release 1 AUTOLOG plus additional functions. To convert a Release 1 AUTOLOG command to XAUTOLOG, change the command name to XAUTOLOG. You do not have to convert an existing Release 1 AUTOLOG command in the following situations:
 - You have class A or class B privileges.
 - The existing command is of the form:
AUTOLOG userid
 - There is an IPL statement in the user's system directory entry.
 - You can tolerate synchronous processing.
 - Your installation is using the password suppression facility.
 - You want to supply a password to complete the automatic log on.
2. Class G users *must* use the XAUTOLOG command. The AUTOLOG command is only class A or class B.
3. XAUTOLOG requires an IPL statement in the system directory entry for the userid unless either the IPL or the NOIPL parameter is specified.
4. If SYNCH is not specified, you receive asynchronous messages about the LOGON and IPL for the userid specified in the XAUTOLOG command.
5. If NOIPL is not specified and thus an automatic IPL *is* scheduled but the IPL statement is invalid, you receive a message and the userid will be forced off the system. In VM/XA SP Release 1, the userid was not forced off the system.

Appendix A. Restricted Userids

Many keywords have a specific meaning to your VM/XA SP when they are encountered in a command input line.

The command syntax of the VM/XA SP CP commands often allows you to enter one of these keywords or a userid with equal frequency. If you have assigned one of these keywords as a userid, an ambiguity results that may cause the system to improperly analyze the command line.

You can avoid this problem by not using any of the following words as a userid:

*	Files	PFnn	TApes
ABEND	GRAF	PRIORity	TAG
ACCount	Hold	PROCEssor	TDisk
ALL	IMG	PUnch	TERMinal
ALLOC	IMSG	Rdr	TIMER
BRKKEY	Links	Reader	TRace
CACHE	LINEs	REAL	TYpe
CCWtran	LINEDit	RECOding	Users
CONsole	LOGmsg	REPLY	UCR
COPY	MACHine	REServed	UR
CPFORMAT	MITime	RUN	V = F
CPTRace	MOD	Set	V = R
CPUs	MODEL	SDF	Virtual
CPUID	MSG	SPECIFIC	VECTor
CTCa	Names	SPOOL	WNG
CTLR	NSS	STORAge	XSTORAGE
DAsd	OFF	STORE	XSTORE
DCSS	Printer	SVC76	
DISPlay	Prt	SYStem	
DUMP	PASSWORD	Time	

Note: Uppercase letters in this table represent the minimum truncations that the system will accept for any of these keywords. Where a keyword is specified in both uppercase and lowercase letters, you should not assign a userid that is exactly the same as any acceptable truncation of the keyword. For example:

STORAge

is a reserved keyword. You should not assign any of the following as userids:

STOR
STORA
STORAG
STORAGE

Summary of Changes

Technical changes throughout this manual are indicated by vertical bars to the left of the changed material.

Second Edition

Form of Publication: SC23-0358-1

Level of Product: VM/XA System Product Release 2

Date of Publication: November 1988

Changes to this publication:

- **New device support**

VM/XA System Product Release 2 fully supports, unless otherwise noted, the following devices:

- 3101 Models 10, 11, 12, 13, 20, 21, 23 (supported through VM/VTAM or VCNA)
- 3174 Model 1L (non-SNA channel attachment or SNA channel attachment through VM/VTAM or VCNA)
- 3174 1R, 51R, 81R, 2R, 3R, 52R, 53R, 82R (remote SNA attachment through VM/VTAM or VCNA)
- 3274 Models 1A, 21A, 31A, or 41A (SNA attachment through VM/VTAM or VCNA)
- 3274 Models 1C, 21C, 31C, 41C, 51C, 61C (remote SNA attachment through VM/VTAM or VCNA)
- 3705, 3720, 3725, 3745 (SNA attachment through VM/VTAM or VCNA)
- 7171 ASCII Device Attachment Control Unit.

- **ASCII device support**

VM/XA System Product Release 2 allows you to connect ASCII TTY display terminals to the system through a VM/VTAM service machine or through a 7171 Device Attachment Control Unit (DACU).

- **Processor Resource/Systems Manager (PR/SM) feature**

The PR/SM feature supersedes the Multiple High Performance Guests Support feature on 3090 Enhanced Model Processors, and offers additional functions.

- **Group control system (GCS)**

GCS is a new component that allows you to implement a native SNA communication network or run RSCS Version 2.

- **CP system trace facility enhancements**

The CP system trace facility allows you to save CP system trace data in system trace files on disk or tape.

- **Data trace facility**

VM/XA System Product Release 2 provides a facility that allows you to set three different types of data tracing for recording in system data files. These three data tracing types are:

- CP data trace
- I/O data trace
- Guest data trace.

- **Soft abend dump enhancements**

VM/XA System Product Release 2 enhances existing soft abend dump support by:

- Allowing more CP functions to request soft abends rather than hard abends
- Providing snapshot information about abending modules' save areas and associated data areas
- Flagging storage frames in the soft abend dump that have changed between the time CP started the dump and the time CP actually dumped the frames
- Changing the format of the soft abend dump to be compatible with the hard abend dump format.

- **Dump viewing facility enhancements**

VM/XA System Product Release 2 enhances the dump viewing facility to allow you to:

- Format CP control blocks or virtual machine control blocks
- Select which trace table entries you wish to view in a CP, standalone, or soft abend dump
- Process GCS and RSCS Version 2 load maps (with the MAP and ADDMAP commands) in addition to the load maps currently supported
- Process data trace information as well as CP trace table data
- Collect GCS guest trace records in a simulated OS QSAM file
- Process CP soft abend dumps
- Process dumps with any CMS file name
- View dump data via the XEDIT interface
- Use the XEDIT interface to extract data from dumps for use in a REXX EXEC
- Use an IBM-supplied EXEC to search a dump for a hung user.

- **CP national language support (NLS)**

VM/XA System Product Release 2 provides CP HELP files and messages in the following languages:

- Mixed-case American English (base version)
- Uppercase American English
- Brazilian Portuguese
- French
- German
- Japanese (Kanji).

- **Managing 37xx communication controllers**

VM/XA System Product Release 2 provides new commands that allow you to manage your 37xx communication controllers.

- **SPTAPE enhancements**

VM/XA System Product Release 2 enhances the SPTAPE command to allow you to dump and load spool files and system data files that span more than one tape volume.

- **DASD dumping and restoring enhancements**

VM/XA System Product Release 2 enhances the DDR and DDRXA programs to allow you to save the DASD data to tape in compacted form.

- **Expanded Storage caching**

VM/XA System Product Release 2 uses expanded storage for caching during logon and IPL processing and for CMS file system minidisk caching.

- **DIRECTXA enhancements**

VM/XA System Product Release 2 enhances the function of the DIRECTXA command and the directory build process by providing automatic deactivation of restricted passwords (ADRP).

- **AUTOLOG command enhancements**

VM/XA System Product Release 2 provides two commands (AUTOLOG and XAUTOLOG) to allow you to log on virtual machines automatically in a disconnected state.

- **LOGON command enhancement**

VM/XA System Product Release 2 enhances the LOGON command to allow users to type in their passwords on the LOGON command line.

- **PF key enhancements**

VM/XA System Product Release 2 enhances PF key functions by allowing you to:

- Issue a command to clear the RETRIEVE PF key buffer
- Specify whether or not you wish a PF key's definition displayed when you press the PF key
- Specify a PF key definition with substitution characters.

- **New and changed publications**

VM/XA System Product Release 2 adds two new books to the VM/XA SP library and combines two existing books into one book. The two new books are:

- *VM/XA System Product: Group Control System Command and Macro Reference*, SC23-0433
- *VM/XA System Product: Group Control System Diagnosis Reference*, LY27-8060

The combined book is *VM/XA SP Release 2 Planning and Administration*. This book replaces the *VM/XA SP Release 1 Administration* (SC23-0353) and *VM/XA SP Release 1 Planning* (GC23-0378) manuals.

- **Application programming enhancements**

VM/XA System Product Release 2 provides new and changed DIAGNOSE code and IUCV functions.

Glossary

A

automatic software re-IPL. The process by which the control program attempts to restart the system after abnormal termination. This process does not involve the hardware IPL process. See also virtual=real machine recovery.

C

CCS. Console communication services.

CCW. Channel command word.

channel command word (CCW). A doubleword structure that directs an I/O operation on a device or channel and includes pointers to any storage areas associated with the operation. One or more CCWs make up a channel program.

CMS. Conversational monitor system.

console communication services (CCS). A group of CP routines that interface with the VTAM service machine, providing full VM/XA™ SP console capabilities for SNA/CCS terminal users.

control program (CP). The component of VM/XA SP that manages the resources of a single System/370-Extended Architecture system so that multiple computing systems appear to exist. Each virtual machine is the functional equivalent of either a System/370 computing system or a System/370-Extended Architecture computing system.

conversational monitor system (CMS). The component of VM/XA SP that, as a virtual machine operating system, provides interactive time-sharing. CMS allows users to communicate with the system and with each other, to create and edit files, and to develop and run application programs. It operates in either System/370 mode or 370-XA mode under the control of CP.

CP. Control program.

D

DCSS. Discontiguous saved segment.

directory. A CP disk file that includes an entry for each user in the system. The entry defines the characteristics of the user's initial virtual machine configuration. These characteristics include the userid, the password, normal and maximum allowable virtual storage, virtual device definitions, the privilege class, the dispatching priority, logical line editing characters, and the account number.

discontiguous saved segment (DCSS). A saved segment that occupies one or more architecturally-defined segments. It begins and ends on segment boundaries. It is accessed by its own name. Contrast with member saved segment. See also saved segment, segment, segment space.

dump viewing facility. A VM/XA SP component that allows users to display, format, and print data interactively from CP hard and softabend, stand-alone, and virtual machine dumps, and to process CP trace table data stored on tape or in a system trace file.

dynamic paging area. The area of real storage allocated by CP for V=V machine paging. This area also contains CP nonresident modules, CP control blocks, CP trace tables, free storage pages, and the alternate processor's prefix storage areas.

E

Expanded Storage. Optional integrated high-speed storage. In VM/XA SP, Expanded Storage may be shared by CP and one or more virtual machines. It may also be dedicated to CP or to a particular virtual machine.

F

full-pack minidisk. A virtual disk that contains all of the addressable cylinders of a real DASD volume.

full-screen mode. In VM/XA SP, the environment in which an entire 3270 display screen is under the control of a program running in a virtual machine.

G

GCS. Group control system.

group control system (GCS). The component of VM/XA SP that, as a virtual machine supervisor, executes in a group of System/370 virtual machines under CP control to provide an interface that helps support a native Systems Network Architecture (SNA) network.

guest. An operating system running in a virtual machine managed by the VM/XA SP control program. Contrast with host.

guest real storage. The storage that appears real to the operating system running in a virtual machine. Contrast with guest virtual storage, host real storage, and host virtual storage.

guest virtual storage. The storage that appears virtual to the operating system running in a virtual machine. Contrast with guest real storage, host real storage, and host virtual storage.

H

host. The VM/XA SP control program in its capacity as manager of a virtual machine in which another operating system is running. Contrast with guest.

host real storage. The storage that appears real to the control program. If VM/XA SP is running native, this is real storage; if VM/XA SP is running

in a virtual machine, this is virtual storage. Contrast with guest real storage, guest virtual storage, and host virtual storage.

host virtual storage. The storage that appears virtual to the control program. Contrast with guest real storage, guest virtual storage, and host real storage.

I

image library. A set of modules, contained in a system data file, that define the spacing, characters, and copy modification data that a 3800 printer uses to print a spool file or that define the spacing and character set that an impact printer uses to print a spool file. See also system data file.

inter-user communication vehicle (IUCV). A generalized CP interface that facilitates the transfer of data among virtual machines.

IUCV. Inter-user communication vehicle.

M

member saved segment. A saved segment that begins and ends on a page boundary. It belongs to up to 64 segment spaces and is accessed either by the segment space name or by its own name. Contrast with discontinuous saved segment. See also saved segment, segment, segment space.

message repository file. A type of system data file that contains a set of VM/XA SP messages translated into a national language.

missing interrupt handler. A CP function for detecting and dealing with real I/O operations that do not complete within a specified time.

multiple preferred guests. A VM/XA SP facility that supports up to six preferred virtual machines when the Processor Resource/Systems Manager™ (PR/SM™) feature is installed in the real machine. See also preferred virtual machine.

N

named saved system (NSS). A copy of an operating system that a user has named and retained in a system data file. The user can load the operating system by its name, which is more efficient than loading it by device number. See also discontinuous saved segment, member saved segment, saved segment, segment space, system data file.

NSS. Named saved system.

P

pageable virtual machine. Synonymous with virtual = virtual machine.

preferred virtual machine. A virtual machine that runs in the V=R area. CP gives this virtual machine preferred treatment in the areas of performance, processor assignment, and I/O interrupt handling. See also multiple preferred guests, virtual = fixed machine, virtual = real area, virtual = real machine.

Processor Resource/Systems Manager (PR/SM). A separately orderable feature available with 3090E processors that provides for logical partitioning of the real machine and support of multiple preferred guests. See also multiple preferred guests.

PR/SM. Processor Resource/Systems Manager.

R

real system operator. Any user who loads and runs VM/XA SP in the real machine. Contrast with virtual machine operator.

S

saved segment. One or more pages of storage that have been named and retained in a system data file. See also discontinuous saved segment, member saved segment, segment, segment space, system data file.

segment. In System/370 architecture, 64 kilobytes of storage. In 370-XA architecture, 1 megabyte of storage. See also saved segment.

segment space. A saved segment composed of up to 64 member saved segments accessed by a single

name. A segment space occupies one or more architecturally-defined segments; it begins and ends on segment boundaries. A user with access to a segment space has access to all of its members. See also discontinuous saved segment, member saved segment, saved segment, segment.

service virtual machine. A virtual machine that provides system services. These services include accounting, error recording, monitoring, and those provided by supported licensed programs.

SMSG function. A CP function that allows a virtual machine to send a special message to another virtual machine programmed to accept and process the message. See also special message.

SNA. Systems Network Architecture.

SNA/CCS terminal. Any terminal accessing VM/XA SP that is managed by a VTAM service machine.

special message. A data transmission, made up of instructions or commands, sent from one virtual machine to another via the SMSG function. A special message is processed by the receiving virtual machine and does not appear on the receiver's console. See also SMSG function.

spool file. A collection of data along with CCWs for processing on a unit record device. Contrast with system data file.

SVC 76. In VM/XA SP, a supervisor call instruction that records the error incidents encountered by certain operating systems running in virtual machines. When a virtual machine operating system issues an SVC 76, VM/XA SP translates the virtual storage and I/O device addresses to real addresses, records the information on the VM/XA SP error recording virtual machine, and returns control to the issuing virtual machine. This interface bypasses the virtual machine's own error recording routine, and avoids duplicate error recording.

System/370 mode. A virtual machine operating mode in which System/370 functions are simulated. Contrast with 370-XA mode.

system data file. A collection of data associated with a particular function. Types of system data files include saved segments, NSSs, UCR files, image libraries, message repository files, and system

trace files. Because a system data file contains no CCWs, it cannot be processed on a unit record device. Contrast with spool file.

system hold status. A spool file status that prevents a file from being printed, punched, or read until the real system operator releases it. Contrast with user hold status.

system trace file. A type of system data file that contains CP or virtual machine trace data.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

U

UCR file. User class restructure file.

unit record device. A reader, a printer, or a punch.

user class restructure file (UCR file). A type of system data file that contains information used to override the IBM-defined privilege class structure of CP commands, DIAGNOSE instruction codes, and certain CP system functions.

user directory. See directory.

user hold status. A spool file status that prevents a file from being printed, punched, or read until the file owner releases it. Contrast with system hold status.

V

Vector Facility (VF). A hardware feature that provides synchronous instruction processing for high-speed manipulation of fixed-point and floating-point data.

VF. Vector Facility.

V = F machine. Virtual = fixed machine.

virtual = fixed machine (V = F machine). A preferred virtual machine with a fixed, contiguous

area of host real storage that does not start at page 0. CP provides performance enhancements for this virtual machine. See also multiple preferred guests, preferred virtual machine, virtual = real area, virtual = real machine, virtual = virtual machine.

virtual machine. In VM/XA SP, a functional equivalent of either a System/370 computing system or a System/370-Extended Architecture computing system. Each virtual machine is controlled by an operating system. VM/XA SP controls the concurrent execution of multiple virtual machines on an actual System/370-Extended Architecture system.

Virtual Machine/Extended Architecture™ System Product (VM/XA SP). An operating system that allows multiple IBM System/370 and 370-XA operating systems to run simultaneously on a single 370-XA processor. The multiple systems may be used for production, testing, developing application programs, maintenance, and migration. VM/XA SP also provides a high-capacity interactive environment. There are four components: the control program (CP), the conversational monitor system (CMS), the dump viewing facility, and the group control system (GCS).

virtual machine operator. Any user who loads and runs an operating system in a virtual machine. Contrast with real system operator.

virtual = real area (V = R area). A fixed, contiguous section of real storage, starting at page 0, in which preferred virtual machines execute. CP does not page this storage. See also preferred virtual machine, virtual = fixed machine, virtual = real machine.

virtual = real machine (V = R machine). A preferred virtual machine with a fixed, contiguous area of host real storage that starts at page 0. CP provides performance enhancements and an automatic recovery facility for this virtual machine. See also multiple preferred guests, preferred virtual machine, virtual = real area, virtual = real machine recovery, virtual = virtual machine.

virtual = real machine recovery (V = R machine recovery). A CP function that allows the V = R machine to resume operation after most CP

Virtual Machine/Extended Architecture is a trademark of the International Business Machines Corporation.

abnormal terminations. When possible, the facility reestablishes the V = R machine environment, allowing the operating system running in that virtual machine to perform its own recovery processes. See also automatic software re-IPL.

virtual = virtual machine (V = V machine). A virtual machine that runs in the dynamic paging area. CP pages this virtual machine's guest real storage in and out of host real storage. See also dynamic paging area, virtual = fixed machine, virtual = real machine.

virtual supervisor state. A condition, controlled by a virtual machine's current PSW, during which the control program allows the virtual machine to issue input/output and other privileged instructions. When these instructions are not emulated, the control program intercepts these instructions and simulates their functions for the virtual machine.

virtual wait time. The period during which the control program suspends the processing of a program while a required resource is unavailable.

VM/XA SP. Virtual Machine/Extended Architecture System Product.

VTAM service machine. A collection of networking programs running in a virtual machine that, together with the CP console communication services (CCS) routines, provide full VM/XA SP console capabilities for SNA/CCS terminal users. A VTAM service machine contains either (1) VM/VTAM with VSCS running as an application under control of GCS, or (2) VM/VCNA running as a VTAM application under control of the VSE or VS1 operating system.

V = R area. Virtual = real area.

V = R machine. Virtual = real machine.

V = R machine recovery. Virtual = real machine recovery.

V = V machine. Virtual = virtual machine.

Numerics

370 mode. Synonym for System/370 mode.

370-XA mode. A virtual machine operating mode in which System/370-Extended Architecture functions are simulated. Contrast with System/370 mode.

Bibliography

This bibliography gives the names and order numbers of microfiche and publications about VM/XA System Product.

VM/XA System Product Microfiche

You can order microfiche listings that contain code. The order numbers for the microfiche are:

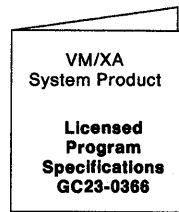
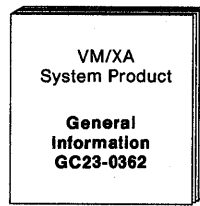
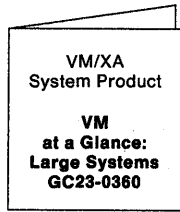
Order No.	Description
LYC7-0330	VM/XA System Product: CP listings
LYC7-0331	VM/XA System Product: CMS listings
LYC7-0332	VM/XA System Product: GCS listings
LYC7-0334	VM/XA System Product: dump viewing facility listings.

VM/XA System Product Publications

The publications are shown in Figure 2 on page 770. You can order any of them by their individual order numbers or you can order most of them as a group by using a single order number, SBOF-0260. SBOF-0260 provides:

- All unlicensed publications (order numbers that do not begin with LY)
- Enough three-ring binders to hold the publications
- Spine and cover inserts for the binders.

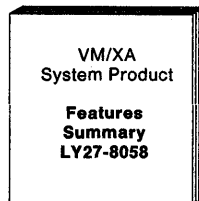
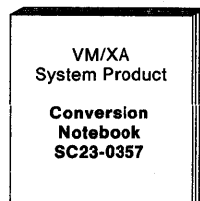
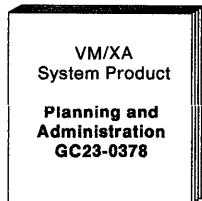
Evaluation



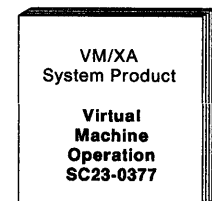
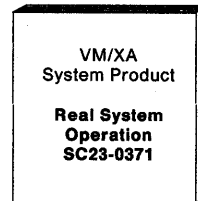
Installation



Planning and Administration



Operation



End Use

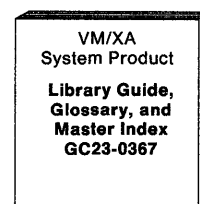
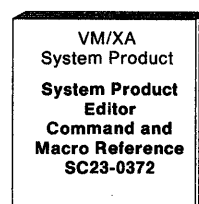
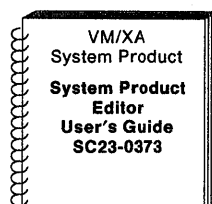
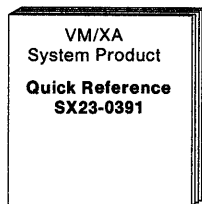
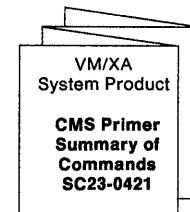
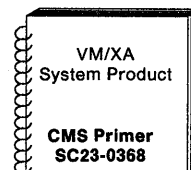
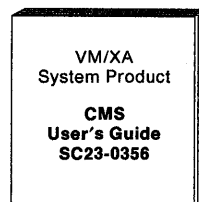
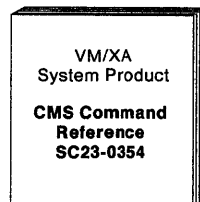
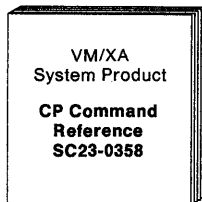
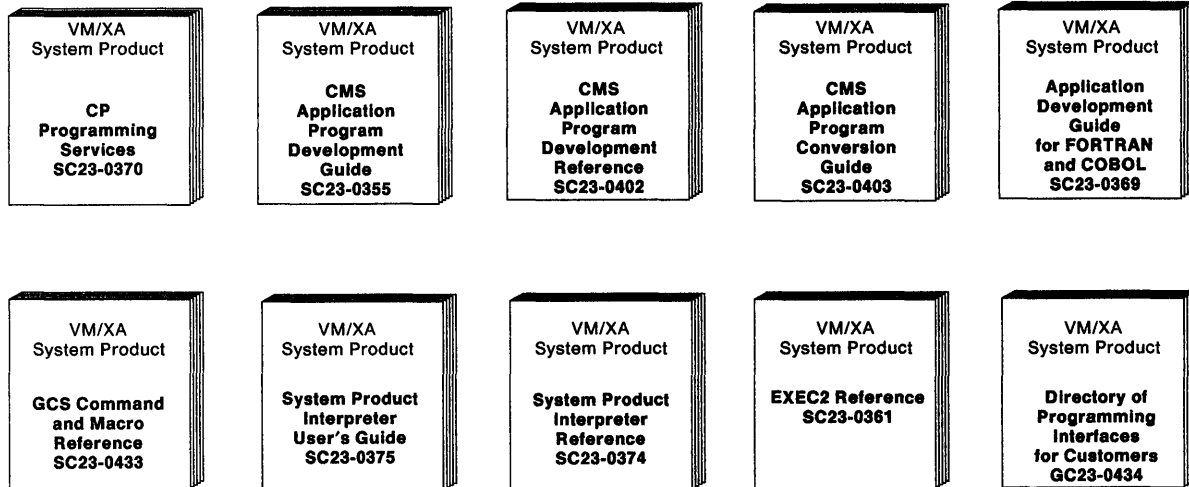
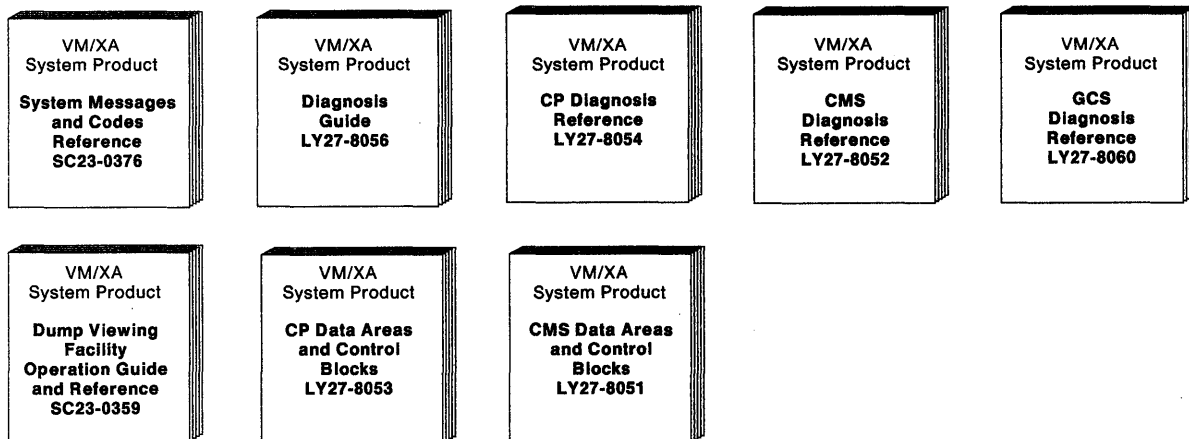


Figure 2 (Part 1 of 2). VM/XA System Product Publications

Application Programming



Diagnosis



Binders and Inserts

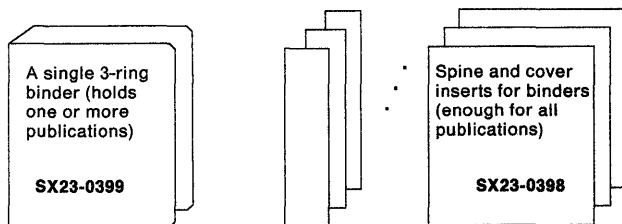


Figure 2 (Part 2 of 2). VM/XA System Product Publications

Index

A

- accessing the VM/XA SP system
 - using the LOGON command 244
- account number, displaying
- ACNT
 - command 20
- activating a protected application environment
 - using SET CONCEAL command 505
- activating CPTRACE entry recording of real machine events
 - using SET CPTRACE command 509
- activating or deactivating caching
 - using SET CACHE command 501
- activating the pseudo page fault facility
 - using SET PAGEX command 539
- activating TRACE sets
 - using TRACE GOTO/CALL/RETURN command 680
- allocating DASD volumes for CP uses
 - using the CPFORMAT command 63
- altering
 - the external attributes of a closed spool file or files 39
 - the user HOLD status of an active spool file 39
- altering a device's missing interrupt interval setting
 - using SET MITIME command 531
- altering processing parameters for CP recording facilities
 - using RECORDING command 472
- altering storage locations in real or virtual machine
 - using STORE command 613
- altering the contents of one or more vector registers
 - using STORE VECTOR command 622
- altering trace traps
 - using TRACE TRAP command 702
- assigning the RETRIEVE function to a PF key
 - using the SET PFnn RETRIEVE function 547
- associating information with the VM/XA SP spool files
 - using TAG command 627
- ATTACH
 - command 21
- attaching a partition of real Expanded Storage 21
 - using ATTACH command 21
- attaching DASD device to CP 21
 - using ATTACH command 21
- attaching real or logical devices 21
 - using ATTACH command 21
- attention interruption pending for your virtual console
 - using ATTN command 30
- ATTN
 - command 30
- AUTOLOG command
 - logging on another user's virtual machine 31

automatic logon 31

B

- BACKSPACE
 - command 35
- BACKWARD
 - command 35
- BEGIN
 - command 37
- bibliography 769

C

- canceling SPTAPE functions
 - using SPTAPE CANCEL command 594
- CHANGE command
 - changing attributes of spool files 39
- changes to this book 759
- changing
 - the configuration of your virtual machine 75
- changing a user's priority of access to system resources
 - using SET SHARE command 555
- changing color or highlighting definitions
 - using SCREEN command 489
- changing control options for your virtual machine
 - using SPOOL READER/PRINTER/PUNCH/CONSOLE command 577
- changing current language
 - using the SET CPLANGUAGE command 508
- changing the allocation map of a DASD volume 64
 - using the CPFORMAT command 64
- changing the current session account number
 - using SET ACCOUNT command 499
- changing the dispatching timeslice
 - using the SET SRM DSPSLICE command 559
- changing the identification of a virtual processor
 - using SET CPUID command 518
- changing the interactive bias system parameter
 - using SET SRM IABIAS command 559
- changing the number of users allowed in the dispatch list
 - using SET SRM DSPBUF command 559
- changing the order of closed spool files on a specified queue
 - using the ORDER command 271
- changing the storage resources to users
 - using the SET SRM STORBUF command 559
- changing the system's paging resources
 - using the SET SRM LDUBUF command 559
- changing the type of virtual processor configuration
 - using SET MACHINE command 528

- changing virtual device numbers
 - using REDEFINE command 474
- checking the system for an eventual warm start
 - using SHUTDOWN command 568
- choosing a virtual processor
 - using the CPU command 71
- classes, user privilege
- clearing all pending interrupts from a virtual device
 - using RESET command 479
- clearing storage
 - using the SYSTEM command 625
- CLOSE
 - command 48
- COMMANDS
 - command 58
- commenting on your console sheet
 - using * command 17
- commenting on your display screen
 - using * command 17
- connecting two virtual channel-to-channel adapters
 - using the COUPLE command 60
- connecting (logically) a display
 - using the DIAL command 120
- console
 - defining 77
- control of special messages
 - using SET SMSG command 558
- controlling break-ins
 - using the TERMINAL BREAKIN command 636
- controlling certain functions associated with the virtual console
 - using the TERMINAL command 630
- controlling CP output destination
 - using SET CPCONIO command 507
- controlling error message handling
 - using SET EMSG command 522
- controlling logical line editing functions
 - using SET LINEDIT command 525
- controlling response destination
 - using SET VMCONIO command 566
- controlling simulation of NOP CCWs in channel programs
 - using SET NOPDATA command 537
- controlling the console mode of your display
 - using the TERMINAL CONMODE command 640
- controlling the terminal attention environment
 - using TERMINAL MODE command 648
- controlling the use of text translation tables
 - using TERMINAL TEXT command 652
- controlling the virtual interval timer
 - using SET TIMER command 565
- controlling use of APL character translation tables
 - using TERMINAL APL command 633
- controlling whether certain informational messages are displayed
 - using SET IMSG command 523
- controlling whether messages are to be received at your display

- controlling whether messages are to be received at your display (*continued*)
 - using SET MSG command 536
- copying your 3270 display screen to a printer
 - using the SET PFnn COPY command 545
- counting successful trace events
 - using TRACE COUNT command 677
- COUPLE
 - command 60
- CP
 - command 62
- CP commands
 - executing from any command environment 18
 - issuing 62
- CPFORMAT
 - command 63
- CPTRAP command (see also TRSAVE, TRSOURCE, SET CPTRACE) 70
- CPU
 - command 71
- creating a virtual Vector Facility
 - using DEFINE VECTOR command 93
- creating account records for logged-on users
 - using ACNT command 20
- cylinders, displaying allocated, in use, and available 299

D

- deactivating a protected application environment
 - using SET CONCEAL command 505
- deactivating CPTRACE entry recording of real machine events
 - using SET CPTRACE command 509
- deactivating the pseudo page fault facility
 - using SET PAGEX command 539
- DEDICATE
 - command 73
- dedicating a real processor to the V=R user
 - using the DEDICATE command 73
- DEFINE command
 - CONSOLE operand 77
 - CPU operand 78
 - CTCA operand 80
 - GRAF operand 81
 - MDISK operand 82
 - spooling device operand 83
 - STORAGE operand 87
 - temporary disk operand 89
 - VECTOR operand 93
 - virtual device operand 91
- defining a function for a program function key on a 3270 display
 - using the SET PFnn command 542
- defining a minidisk
 - using DEFMDISK command 82
- defining a skeleton system data file for a DCSS
 - using the DEFSEG command 95

defining a skeleton system data file for an NSS 101
 using the DEFSYS command 101
 defining a spooling device 83
 defining a temporary disk
 using DEFINE (temporary disk) command 89
 defining a virtual channel-to-channel adapter
 using DEFINE CTCA command 80
 defining a virtual display station
 using DEFINE GRAF command 81
 defining a virtual system console
 using DEFINE CONSOLE command 77
 defining a virtual Vector Facility
 using DEFINE VECTOR command 93
 defining a 3800 printer
 using the DEFINE (spooling device) command 83
 defining logical tab characters on display devices
 using TERMINAL TABCHAR command 651
 defining storage size for virtual machine
 using DEFSTOR command 87
 defining the break key
 using the TERMINAL BRKKEY command 637
 defining the logical character delete symbol
 using TERMINAL CHARDEL command 639
 defining the logical escape character
 using TERMINAL ESCAPE command 643
 defining the logical line delete symbol
 using TERMINAL LINEDEL command 645
 defining the logical line end symbol
 using TERMINAL LINEND command 646
 defining virtual processors
 using the DEFINE CPU command 78
 defining 3101s or TTYs
 using the TERMINAL TYPE command 653
 DEFSEG command 95
 DEFSYS command 101
 deleting a log message
 using SET LOGMSG command 526
 deleting a specified line in a log message
 using SET LOGMSG command 526
 deleting all trace traps and all trace sets
 using TRACE END command 678
 deleting all traps in a named trace set
 using TRACE CLEAR command 676
 deleting message repository files
 using the PURGE NLS command 283
 deleting trace traps
 using TRACE TRAP command 702
 designating a user for system resources
 using SET QUICKDSP command 550
 designating a virtual processor
 using the CPU command 71
 designating the unit to receive a VM/XA SP system
 ABEND dump
 using SET DUMP command 519
 DETACH command
 CPU operand 107
 VECTOR operand 116
 VIRTUAL operand 112
 DETACH command (*continued*)
 XSTORE operand 118
 (real device) operand 108
 determining all devices on a path
 using the QUERY CHPID command 305
 determining your system status and machine
 configuration
 using the QUERY command 290
 DIAL command 120
 DISABLE command 122
 DISCONNECT command 124
 disconnecting your display station from the VM/XA SP
 system
 using the DISCONNECT command 124
 DISPLAY
 displaying contents of prefix register 139
 displaying contents of registers 141
 displaying first-level storage 135
 displaying guest storage 130
 displaying information from SCHIBs 144
 displaying PSW information 140
 displaying vector registers 146
 displaying a user's share of the system
 using QUERY SHARE command 401
 displaying account number
 using QUERY ACCOUNT command 298
 displaying all paths available to a specified device
 using QUERY PATHS command 377
 displaying allocation of cylinders
 using QUERY ALLOC command 299
 displaying caching status for 3880 subsystems
 using QUERY CACHE command 301
 displaying contents of first-level storage
 using DISPLAY HOST STORAGE command 135
 displaying contents of prefix register
 using DISPLAY PREFIX command 139
 displaying contents of registers
 using DISPLAY REGISTERS command 141
 displaying contents of storage at specified locations
 using DISPLAY GUEST STORAGE command 130
 displaying count of spool files
 using QUERY FILES command 337
 displaying count of system data files
 using QUERY SDF command 393
 displaying CP language identifiers
 using the QUERY CPLANGLIST command 310
 displaying current CP language
 using the QUERY CPLANGUAGE command 311
 displaying current limit of logged on users
 using QUERY MAXUSERS command 352
 displaying current online or offline status of paths to a
 device
 using QUERY PATHS command 377
 displaying current profile information
 using QUERY MONITOR command 356
 displaying current session time and date information
 using QUERY TIME command 413

- displaying current settings of dump options
 - using QUERY DUMP command 335
- displaying current terminal options
 - using QUERY TERMINAL command 410
- displaying current TRACE settings
 - using the QUERY TRACE command 414
- displaying current user access to a system DASD volume
 - using QUERY SYSTEM command 405
- displaying currently defined temporary disk space
 - using QUERY TDSK command 409
- displaying daily system log message
 - using QUERY LOG command 350
- displaying dispatch lists
 - using INDICATE QUEUES command 211
- displaying Expanded Storage Facility current assignment
 - using QUERY XSTORE command 467
- displaying Expanded Storage size
 - using QUERY VIRTUAL XSTORE command 464
- displaying extended color and highlighting values
 - using QUERY SCREEN command 392
- displaying file names stored on a tape
 - using SPTAPE SCAN command 601
- displaying IABIAS system parameter
 - using QUERY SRM command 402
- displaying information about CP system
 - using QUERY CP command 312
- displaying information about image library files
 - using the QUERY IMG command 345
- displaying information about links to virtual device
 - using QUERY LINKS command 348
- displaying information about logged-on users
 - using QUERY USERS command 431
- displaying information about message repository files
 - using the QUERY NLS command 366
- displaying information about NSS/DCSS files
 - using the QUERY NSS command 368
- displaying information about spool files
 - using QUERY READER/PRINTER/PUNCH command 383
- displaying information about terminal event records
 - using QUERY MONDATA command 355
- displaying information about trace files
 - using the QUERY TRFILES command 417
- displaying information about UCR files
 - using the QUERY UCR command 428
- displaying information from SCHIBs
 - using DISPLAY SCHIB command 144
- displaying list of logged-on users
 - using QUERY NAMES command 364
- displaying list of real processors and their system use
 - using QUERY PROCESSORS command 381
- displaying missing interrupt detector time interval settings
 - using QUERY MITIME command 353
- displaying operating load
 - using INDICATE LOAD command 205
- displaying page residency data
 - using INDICATE PAGING command 209
- displaying processor identification
 - using QUERY CPUID command 317
- displaying program function key assignments
 - using QUERY PFnn command 379
- displaying PSW information
 - using DISPLAY PSW command 140
- displaying real and virtual machine components
 - using the DISPLAY command 126
- displaying real storage allocations for V = R and V = F guests
 - using QUERY V = R command 465
- displaying resources used by a virtual machine
 - using INDICATE USER command 213
- displaying secondary users
 - using QUERY SECUSER command 394
- displaying setting of real system event tracing
 - using QUERY CPTRACE command 313
- displaying size of virtual storage
 - using QUERY VIRTUAL STORAGE command 460
- displaying status of a virtual device
 - using QUERY VIRTUAL DEVICE command 447
- displaying status of CP data collection (ACCOUNTING and EREP)
 - using QUERY RECORDING command 390
- displaying status of data from TRSOURCE traces
 - using the QUERY TRSOURCE command 423
- displaying status of interpretive execution assist
 - using QUERY IOASSIST command 347
- displaying status of locally-attached virtual display devices
 - using QUERY VIRTUAL GRAF command 449
- displaying status of logical devices
 - using QUERY LOGICAL command 349
- displaying status of real devices
 - using QUERY real device command 320
- displaying status of real storage
 - using the QUERY FRAMES command 340
- displaying status of SET command functions
 - using QUERY SET command 396
- displaying status of SOFT ABEND function
 - using QUERY ABEND command 297
- displaying status of TRSAVE function
 - using QUERY TRSAVE command 420
- displaying status of unit record devices
 - using QUERY VIRTUAL UR command 462
- displaying status of Vector Facilities and users
 - using QUERY VECTOR command 433
- displaying status of virtual channel-to-channel adapters
 - using QUERY VIRTUAL CTCA command 442
- displaying status of virtual DASD
 - using QUERY VIRTUAL DASD command 444
- displaying status of virtual printers
 - using QUERY VIRTUAL PRINTER command 451

- displaying status of virtual punches
 - using QUERY VIRTUAL PUNCH command 455
- displaying status of virtual readers
 - using QUERY VIRTUAL READER command 458
- displaying status of virtual tape devices
 - using QUERY VIRTUAL TAPES command 461
- displaying status of virtual Vector Facilities
 - using QUERY VIRTUAL VECTOR command 463
- displaying status of your virtual console
 - using VIRTUAL CONSOLE command 436
- displaying status of your virtual machine configuration
 - using QUERY VIRTUAL ALL command 435
- displaying status reserved DASD full pack mini-disks
 - using QUERY DASD RESERVE command 318
- displaying system resources at your console
 - using the INDICATE command 202
- displaying system-wide parameters used by the scheduler
 - using QUERY SRM command 402
- displaying tag information for a spool device or file
 - using QUERY TAG command 407
- displaying the address of a module identified by a symbolic label
 - using the LOCATE command 236
- displaying the address of CP control blocks
 - using the LOCATE command 236
- displaying the contents of Vector Facility registers
 - using the DISPLAY command 146
- displaying the size of real storage
 - using QUERY STORAGE command 404
- displaying the spool file limit
 - using QUERY MAXSPOOL command 351
- displaying the traceback table
 - using TRACE TABLE command 699
- displaying user's QUICKDSP attribute
 - using QUERY QUICKDSP command 382
- displaying user's authorized commands
 - using the COMMANDS command 58
 - using the QUERY COMMANDS command 308
- displaying user's authorized diagnose codes
 - using the COMMANDS command 58
 - using the QUERY COMMANDS command 308
- displaying userid and system identifier of user/display
 - using QUERY USERID/userid command 430
- displaying users in LOGON HOLD status
 - using QUERY HOLD command 343
- displaying users with output in system HOLD status
 - using QUERY HOLD command 343
- displaying Vector Facility register contents
 - using the DISPLAY command 126
- displaying Vector Facility users
 - using INDICATE VECTOR command 218
- displaying virtual machines in I/O wait state
 - using INDICATE I/O command 204
- displaying virtual machines in page wait status
 - using INDICATE PAGING command 209
- displaying virtual processors defined for your machine
 - using QUERY VIRTUAL CPUS command 440

- displaying warning messages issued from the WARNING command
 - using SET WNG command 567
- displaying whether passwords go on command line
 - using the QUERY PASSWORD command 375
- documentation 769
- DRAIN command
- DUMP
 - designating the unit to receive a VM/XA SP system
 - ABEND dump 519
 - displaying setting of dump options 335
 - formatted for the dump viewing facility 745
 - GUEST STORAGE operand 161
 - HOST STORAGE operand 166
 - including contents of virtual Vector Facilities containing data 745
 - using VMDUMP command 745
 - PREFIX operand 170
 - PSW operand 171
 - REGISTERS operand 173
 - SCHIB operand 176
 - SPTAPE DUMP command 595
 - VECTOR operand 178
- dumping areas of first-level storage
 - using DUMP HOST STORAGE command 166
- dumping contents of prefix register
 - using the DUMP PREFIX command 170
- dumping files for storage or migration
 - using SPTAPE DUMP command 595
- dumping information from subchannel information blocks
 - using the DUMP SCHIB command 176
- dumping locations in virtual machine storage
 - using the DUMP GUEST STORAGE command 161
- dumping program status words information
 - using DUMP PSW command 171
- dumping the contents of registers
 - using the DUMP REGISTERS command 173
- dumping Vector Facility information
 - using the DUMP VECTOR command 178
- dumping Vector Facility registers
 - using the DUMP VECTOR command 178

E

- ECHO command
- ENABLE command
- enabling disabled or nonenabled display devices
 - using the ENABLE command 183
- ending all system functions
 - using SHUTDOWN command 568
- event data collection 256
 - establishing profile for each type of data collection 256
- executing CP commands from any command environment
 - using #CP command 18

EXTERNAL command
simulating an external interrupt 184

F

FLUSH command
FORCE command
forcing single-spacing of an output file
using SPACE command 572
forcing soft abend codes to cause full system
termination
using SET ABEND command 498
forcing users off the system
using the FORCE command 188
formatting DASD volumes for CP uses
using the CPFORMAT command 63
FORWARD command
FREE command
LOGON operand 193
PRINTER, PUNCH option 194

H

HALT command
halting or purging a spool file
using the FLUSH command 186
hard-stop 612
highlighting the command redisplay line on console
using TERMINAL HILIGHT command 644
HOLD command
LOGON option 199
PRINTER, PUNCH options 200
holding current output on a specified real unit record
device
using the FLUSH command 186

I

including data in MONITOR records
using SET MONDATA command. 535
increasing output file number of copies
using the REPEAT command 476
INDICATE
I/O operand 204
LOAD operand 205
PAGING operand 209
QUEUES operand 211
USER operand 213
VECTOR operand 218
indicating how users must enter commands that require
passwords
using the SET PASSWORD command 540
inserting TTY characters
using the TERMINAL CNTL command 642
intercepting machine checks
using TRACE MCH command 688
interpretive execution assist
displaying status of 347

IPL command 219
issuing CP commands
using CP command 62

L

labeling volumes for CP uses
using the CPFORMAT command 63
limiting autolog commands
using the QUERY CMDLIMIT command 307
limiting incorrect passwords
using the SET CMDLIMIT command 504
LINK command
usage notes 224
LOADBUF command
loading files from tape
using the SPTAPE LOAD command 598
loading spool files and system data files from tape
using SPTAPE command 586
loading the Forms Control Buffer with an image
using the LOADBUF command 228
loading the Universal Character Set with a print
chain/train image
using the LOADBUF command 228
LOADVFCB command
LOCATE command 236
LOCK command 238
locking in real storage pages of the pageable CP nucleus
using the LOCK command 238
locking in selected pages of the a user's virtual storage
using the LOCK command 238
logging on another user's virtual machine 31
using the XAUTOLOG command 751
LOGOFF command 241
LOGON command
logon, automatic 31

M

make a virtual spooling device appear to have the status
"not ready"
using NOTREADY command 270
making a device usable or unusable by the control
program or user
using VARY command 734
using VARY PATH command 737
using VARY VECTOR command 743
making a full pack mini-disk shareable or not shareable
using SET SHARED command 557
making an attention interruption pending for your
virtual console
using REQUEST command 478
making an installed Vector Facility available or
unavailable for use
using VARY VECTOR command 743
making an installed Vector Facility logically available or
unavailable for use
using VARY PATH command 737

making other devices available to your virtual machine
using the LINK command 223
merging trace traps into one trace set
using TRACE APPEND command 673

MESSAGE

ALL operand 250
OPERATOR operand 251
userid operand 251
* operand 251

MONITOR

activating event or sample data monitors 267
deactivating event or sample data monitors 267
EVENT operand 256
SAMPLE operand 262
START operand 267
starting the event data monitor 256
starting the sample data monitor 262
STOP operand 267
stopping the event data monitor 256
stopping the sample data monitor 262
monitoring events in a virtual machine
using the PER command (see also TRACE) 275

MSGNOH command

sending no-header messages to specified users 269

N

NOTREADY command

O

ORDER command

P

passwords

indicating how users must enter commands that
require passwords 540
using the QUERY PASSWORD command 375

PER command (see also TRACE)

placing all virtual processors into soft-stop
using STOP command 612

placing selected processors into hard-stop
using STOP command 612

placing user spool files into system HOLD status
using the HOLD command 198

placing your display device in the echo environment
using the ECHO command 182

preventing display stations from accessing the VM/XA
SP system

using the DISABLE command 122

preventing users from logging on to the system
using the HOLD command 198

privilege classes for users

publications 769

PURGE

IMG operand 281
NLS operand 283

PURGE (continued)

NSS operand 285
TRFILES operand 287
UCR operand 289

purging UCR files

using the PURGE UCR command 289

Q

QUERY command

ABEND operand 297
ACCOUNT operand 298
ALLOC operand 299
CACHE operand 301
CHPID operand 305
CMDLIMIT operand 307
COMMANDS operand 308
CPLANGLIST operand 310
CPLANGUAGE operand 311
CPLEVEL operand 312
CPTRACE operand 313
CPUID operand 317
DUMP operand 335
FILES operand 337
FRAMES operand 340
HOLD operand 343
IMG operand 345
IOASSIST operand 347
LINKS operand 348
LOGICAL operand 349
LOGMSG operand 350
MAXSPOOL operand 351
MAXUSERS operand 352
MITIME operand 353
MONDATA operand 355
MONITOR operand 356
NAMES operand 364
NLS operand 366
NSS operand 368
PASSWORD operand 375
PATHS operand 377
PER operand 378
PFnn operand 379
PRINTER operand 383
PROCESSORS operand 381
PUNCH operand 383
QUERY DASD RESERVE operand 318
QUICKDSP operand 382
READER operand 383
real device operand 320
RECORDING operand 390
SCREEN operand 392
SDF operand 393
SECUSER operand 394
SET operand 396
SHARE operand 401
SRM operand 402
STORAGE operand 404

QUERY command (*continued*)
 SYSTEM operand 405
 TAG operand 407
 TDSK operand 409
 TERMINAL operand 410
 TIME operand 413
 TRACE operand 414
 TRFILES operand 417
 TRSAVE operand 420
 TRSOURCE operand 423
 UCR operand 428
 USERID operand 430
 USERS operand 431
 VECTOR operand 433
 VIRTUAL ALL operand 435
 VIRTUAL CONSOLE operand 436
 VIRTUAL CPUS operand 440
 VIRTUAL CTCA operand 442
 VIRTUAL DASD operand 444
 VIRTUAL DEVICE operand 447
 VIRTUAL GRAF operand 449
 VIRTUAL PRINTER operand 451
 VIRTUAL PUNCH operand 455
 VIRTUAL READER operand 458
 VIRTUAL STORAGE operand 460
 VIRTUAL TAPES operand 461
 VIRTUAL UR operand 462
 VIRTUAL VECTOR operand 463
 VIRTUAL XSTORE operand 464
 V=R operand 465
 XSTORE USER operand 467
QUERY CPTRAP command (see **QUERY TRSAVE**,
QUERY TRSOURCE, **QUERY CPTRACE**) 316

R

READY command 471
RECORDING command 472
REDEFINE command 474
 redefining
 the configuration of your virtual machine 75
 redefining a virtual device number 91
 releasing a user from **LOGON HOLD** status
 using the **FREE LOGON** command 193
 releasing a user's spool files from system **HOLD** status
 using the **FREE PRINTER/PUNCH** command 194
 removing a real processor from a user's exclusive use
 using the **UNDEDICATE** command 731
 removing a set of users' spool files from system hold
 status
 using the **FREE** command 192
 removing a user from **LOGON HOLD** status
 using the **FREE** command 192
 removing a virtual Vector Facility from the virtual
 configuration
 using the **DETACH** command 105
 removing closed spool files from the system
 using the **PURGE** command 278

removing real devices from the CP system
 using the **DETACH** command 105
 removing real devices from virtual machine
 configuration
 using the **DETACH** command 105
 removing real Expanded Storage from a user's virtual
 machine
 using the **DETACH** command 105
 removing system trace files
 using the **PURGE TRFILES** command 287
 removing the Expanded Storage capability from your
 virtual machine configuration 118
 removing virtual devices from your virtual machine
 using the **DETACH** command 105
 removing virtual processor(s) from your virtual machine
 using the **DETACH** command 105
REPEAT command
 repositioning current spool output on a real printer or
 punch
 using the **FORWARD** command 190
 repositioning spool output on real printer or punch
 using **FORWARD** command 190
REQUEST command 478
 reserving Expanded Storage for CP use
 using **RETAIN XSTORE** command 481
 reset accumulated accounting data
 using **ACNT** command 20
RESET command 479
 restarting a spooling device after it has been drained
 using **START** command 603
 restarting the current output on a real punch or printer
 using the **BACKSPACE** command 35
 using the **BACKWARD** command 35
 resuming execution on a specific processor or processors
 using **BEGIN** command 37
 resuming execution on a virtual multiprocessor
 configuration
 using **BEGIN** command 37
RETAIN XSTORE command 481
 retrieving files from other virtual machines
 using the **TRANSFER** command 704
REWIND
 command 484
 rewinding a tape on a real tape drive
 using **REWIND** command 484

S

sample data collection 262
 establishing profile for each type of data
 collection 262
SAVESEG command 485
SAVESYS command 487
 saving a DCSS into a defined SDF
 using **SAVESEG** command 485
 saving an NSS into a defined SDF
 using **SAVESYS** command 487

- saving spool files and system data files on tape
 - using SPTAPE command 586
- saving trace tables on tape
 - using the TRSAVE command 710
- SCREEN
 - changing color or highlighting definitions 489
- selecting a virtual processor
 - using the CPU command 71
- selecting translation table
 - using the TERMINAL ASCIITBL command 634
 - using the TERMINAL ATTN command 635
- SEND command 491
- sending a special message
 - using SMSG command 571
- sending console input to a disconnected virtual machine
 - using SEND command 491
- sending files to other virtual machines
 - using the TRANSFER command 704
- sending high-priority messages to a specified user
 - using WARNING command 749
- sending messages to virtual consoles of other logged on users
 - using the MESSAGE command 248
- sending no-header messages to specified users
 - using the MSGNOH command 269
- SET command
 - ABEND operand 498
 - ACCOUNT operand 499
 - AUTOPOLL operand 500
 - CACHE operand 501
 - CCWTRAN operand 503
 - CMDLIMIT operand 504
 - CONCEAL operand 505
 - CPCONIO operand 507
 - CPLANGUAGE operand 508
 - CPTRACE operand 509
 - CPUID operand 518
 - DUMP operand 519
 - EMSG operand 522
 - IMSG operand 523
 - IOASSIST operand 524
 - LINEDIT operand 525
 - LOGMSG operand 526
 - MACHINE operand 528
 - MAXUSERS operand 529
 - MIH operand 530
 - MITIME operand 531
 - MODE (error recording) operand 534
 - MONDATA operand 535
 - MSG operand 536
 - NOPDATA operand 537
 - NOTRANS command operand 538
 - PAGEX operand 539
 - PASSWORD operand 540
 - PFnn COPY operand 545
 - PFnn operand 542
 - PFnn RETRIEVE operand 547
 - PFnn TAB operand 549

- SET command (*continued*)
 - QUICKDSP operand 550
 - RECORD operand 552
 - RESERVED operand 553
 - RUN operand 554
 - SHARE operand 555
 - SHARED operand 557
 - SMSG operand 558
 - SVC76 operand 564
 - TIMER operand 565
 - VMCONIO operand 566
 - WNG operand 567
- SET SRM command 559
- sets the prompting sequence to be used on a TTY device.
 - using the TERMINAL PROMPT command 649
- setting a device-end interruption pending for a virtual device
 - using READY command 471
- setting CCW translation
 - using the SET CCWTRAN command 503
- setting CP on the virtual machine to handle the SVC76 instruction
 - using SET SVC76 command 564
- setting error recording mode for system recovery machine checks
 - using SET MODE command 534
- setting maximum number of logon users
 - using SET MAXUSERS command 529
- setting recording mode for a device
 - using SET RECORD command 552
- setting start interpretive execution assist for a V=R guest
 - using SET IOASSIST command 524
- setting the reserved page performance option
 - using SET RESERVED command 553
- setting values for the TAB function
 - using SET PFnn TAB command 549
- setting your virtual machine to run after a command is issued
 - using SET RUN command 554
- SHUTDOWN command 568
- simulating an external interruption pending to your virtual machine
 - using the EXTERNAL command 184
- simulating an initial program load function for your virtual machine
 - using the IPL command 219
- simulating an I/O interrupt
 - using SET MIH command 530
- simulating the action of RESET and RESTART on a real console
 - using the SYSTEM command 625
- SLEEP command 569
- SMSG command 571
- soft-stop 612
- SPACE command 572

- specifying how CP detects autopolling changes
 - using SET AUTOPOLL command 500
- specifying scrolled lines on 3101
 - using the TERMINAL SCROLL command 650
- specifying the forms control buffer image
 - using the LOADVFCB command 233
- specifying the maximum line length for display screen output
 - using the TERMINAL LINESIZE command 647
- spool files
 - changing attributes of 39
- SPOOL READER/PRINTER/PUNCH/CONSOLE command 577
- SPTAPE command 586
 - CANCEL operand 594
 - DUMP operand 595
 - LOAD operand 598
 - SCAN operand 601
 - STOP operand 602
- SPTAPE DUMP command 595
- SPTAPE LOAD command 598
- START command 603
- STOP command 612
- stopping processing of SPTAPE dump
 - using SPTAPE STOP command 602
- stopping spooling operations on specified real unit record devices
 - using the DRAIN command 150
- stopping virtual machine for a specific amount of time
 - using SLEEP command 569
- storage
 - defining for virtual machine 87
- STORE command
 - GUEST STORAGE operand 615
 - HOST STORAGE operand 617
 - PSW/CAW/CSW operands 619
 - REGISTERS operand 620
 - STATUS operand 621
 - VECTOR operand 622
- storing data in first-level storage
 - using STORE HOST STORAGE command 617
- storing data in second or third-level storage
 - using STORE GUEST STORAGE command 615
- storing data in the current virtual machine PSW, CAW, or CSW
 - using STORE PSW/CAW/CSW command 619
- storing data into registers
 - using STORE REGISTERS command 620
- storing selected virtual machine data in low storage locations
 - using STORE STATUS command 621
- summary of changes to this book 759
- SYSTEM command 625

T

TAG command 627

- TERMINAL command
 - APL operand 633
 - ASCIITBL operand 634
 - ATTN operand 635
 - BREAKIN operand 636
 - BRKKEY operand 637
 - CHARDEL operand 639
 - CNTL operand 642
 - CONMODE operand 640
 - ESCAPE operand 643
 - HIGHLIGHT operand 644
 - LINEDEL operand 645
 - LINEND operand 646
 - LINESIZE operand 647
 - MODE operand 648
 - PROMPT operand 649
 - SCROLL operand 650
 - TABCHAR operand 651
 - TEXT operand 652
 - TYPE operand 653
- terminating execution of an active channel program on a real device
 - using the HALT command 196
- terminating spooling activity on a console device
 - using the CLOSE command 48
- terminating spooling activity on a virtual spooled unit record
 - using the CLOSE command 48
- terminating virtual machine execution
 - using the LOGOFF command 241
- TRACE CALL
 - command 680
- TRACE command 654
 - ALL operand 672
 - APPEND operand 673
 - BRANCH operand 674
 - CLEAR operand 676
 - COUNT operand 677
 - END operand 678
 - EXTERNAL/PROGRAM operand 679
 - GOTO operand 680
 - GPRs operand 682
 - INSTRUCTION operand 684
 - I/O operand 686
 - MCH operand 688
 - mnemonic1 operand 689
 - mnemonic2 operand 691
 - options 659
 - RIO/RSCH/SIO/SIOF/SSCH operands 693
 - STORE operand 695
 - SVC/DIAGNOSE/MC operands 697
 - TPI operand 700
 - TRAP operand 702
 - using TRACE command 654
- TRACE RETURN
 - command 680
- TRACE TABLE
 - command 699

tracing alterations to a virtual machine's general purpose registers
 using TRACE GPRs command 682

tracing alterations to virtual machine storage
 using TRACE STORE command 695

tracing external or program interruptions
 using TRACE EXTERNAL/PROGRAM 679

tracing input/output instructions and/or interruptions for devices
 using TRACE I/O command 686

tracing input/output instructions within a device range
 using TRACE mnemonic2 command 691

tracing instructions in a virtual machine
 using TRACE INSTRUCTION command 684

tracing interruptions and instructions
 using TRACE ALL command 672

tracing real machine events
 using SET CPTRACE command 509

tracing RIO/RSCH/SIO/SIOF/SSCH instructions for I/O devices
 using TRACE RIO/RSCH/SIO/SIOF/SSCH command 693

tracing successful branching instructions
 using TRACE BRANCH command 674

tracing supervisor calls, DIAGNOSE instructions or monitor calls
 using TRACE SVC/DIAGNOSE/MC command 697

tracing Test-Pending-Interrupt instructions
 using TRACE TPI command 700

tracing the execution of assembler language instructions
 using TRACE mnemonic1 command 689

TRANSFER command

TRSAVE command

TRSOURCE command 715

U

UNDEDICATE command 731

UNLOCK command

unlocking page frames
 using the UNLOCK command 732

V

VARY command

VARY PATH command

VARY VECTOR command

varying a path offline
 using VARY OFFLINE PATH command 737

varying a path online
 using VARY ONLINE PATH command 737

virtual machine storage
 defining for virtual machine 87

VMDUMP
 command 745
 formatted for the dump viewing facility
 using VMDUMP command 745

W

WARNING
 command 749

X

XAUTOLOG command 751

Special Characters

* (ASTERISK COMMAND)
 command 17

#CP command

Please use this form to communicate your comments about the usability of the VM system, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the Product Usability Department for appropriate review and action, if any. Comments may be written in your own language; English is not required.

System Information

If you answer **No**, please explain.

- | | Yes | No |
|---|--------------------------|--------------------------|
| • Does the VM system meet your needs? | <input type="checkbox"/> | <input type="checkbox"/> |
| • Is it easy to use and understand? | <input type="checkbox"/> | <input type="checkbox"/> |
| • Are the commands/messages easy to understand and use? | <input type="checkbox"/> | <input type="checkbox"/> |
| • Are the HELP facilities appropriate? | <input type="checkbox"/> | <input type="checkbox"/> |

Customer Information

- What is your occupation? _____
- How long have you been in this occupation? _____
- How long have you been using VM? _____
- Indicate the tasks your job involves:

Evaluation	<input type="checkbox"/>	Planning	<input type="checkbox"/>
Installation	<input type="checkbox"/>	Administration	<input type="checkbox"/>
Customization	<input type="checkbox"/>	Operations	<input type="checkbox"/>
Diagnosis	<input type="checkbox"/>	End Use	<input type="checkbox"/>
Other	<input type="checkbox"/>		

Your Comments:

We appreciate your comments.

If you would like a reply, please supply your name and address on the reverse side of this form. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automatic mail-sorting equipment. Please use pressure-sensitive or other gummed tape to seal this form.

System Usability Comments

Fold and Tape

Please Do Not Staple

Fold and Tape



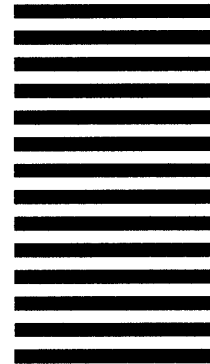
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

**International Business Machines Corporation
Department 47U MS 914
Neighborhood Road
Kingston, New York 12401**



Fold and Tape

Please Do Not Staple

Fold and Tape

If you would like a reply, *please print:*

Your Name _____
Company Name _____ Department _____
Street Address _____
City _____
State _____ Zip Code _____
IBM Branch Office serving you _____



PRINTED IN U.S.A.

CP Command Reference

Order No. SC23-0358-1

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you. Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

How did you use this publication?

- As an introduction As a text (student)
 As a reference manual As a text (instructor)
 For another purpose (explain) _____

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number:

Comment:

What is your occupation? _____

Newsletter number of latest Technical Newsletter (if any) concerning this publication: _____

If you wish a reply, give your name and address: _____

IBM branch office serving you _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automatic mail-sorting equipment. Please use pressure-sensitive or other gummed tape to seal this form.

Reader's Comment Form

Fold and Tape

Please Do Not Staple

Fold and Tape



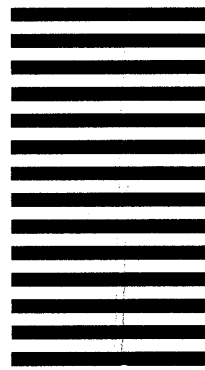
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 52Q MS 458
Neighborhood Road
Kingston, New York 12401



Fold and Tape

Please Do Not Staple

Fold and Tape





Program Number
5664-308

File Number
S370-36

SC23-0358-1



Printed in U.S.A.