IBM

GC23-0549-0

Virtual Machine/Extended Architecture™ System Product VM/XA™ SP Release 2.1

Support for the 3390 Direct Access Storage Device

Program Update Information

GC23-0549-0

Virtual Machine/Extended Architecture™
System Product VM/XA™ SP Release 2.1

Support for the 3390 Direct
Access Storage Device

Program Update Information

First Edition (December 1989)

This edition, GC23-0549-0, applies to Release 2.1 of the IBM 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 product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service which does not infringe any of IBM's intellectual property rights or other legally protectable rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

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 511, 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.

Preface

Purpose

This book provides reference material and task descriptions for the Virtual Machine/Extended Architecture System Product (VM/XA SP) support for the IBM 3390 Direct Access Storage (DASD).

Audience

This book is intended for system programmers, system administrators, IBM Service Representatives, and others interested in an overview of the VM/XA support for the IBM 3390 DASD.

This book assumes that you are familiar with VM/XA SP. To use this book effectively, you should first read, or have on hand, *IBM 3390 Direct Access Storage Introduction* and *Using IBM 3390 Direct Access Storage in a VM Environment.*

Organization

With the exception of Chapter 1, "Introduction" and Chapter 3, "Procedure for Switching 3390 Operating Modes," the sections in this book relate to changes in specific VM/XA SP books. This relationship is as follows:

Chapter	Publication
Chapter 2. Installation and Service	*Planning and Administration*, GC28-0378 *Installation and Service*, SC23-0364
Chapter 4. DASD Dump Restore (DDR)	*CP Programming Services*, SC23-0370 *CMS Command Reference*, SC23-0354
Chapter 5. Changed CMS Messages	*System Messages and Codes Reference*, SC23-0376
Chapter 6. CP Programming Services	*CP Programming Services*, SC23-0370
Chapter 7. Changed System Product Interpreter Commands	*System Product Interpreter Reference*, SC23-0374
Chapter 8. Changed CP and CMS Commands	*CMS Command Reference*, SC23-0354 *CP Command Reference*, SC23-0358
Appendix A. New and Changed Modules, Control Blocks, Macros, and Help Files	*CP Data Areas and Control Blocks*, LY27-8053 *CP Diagnosis Reference*, LY27-8054

Where to Find More Information

For more information about 3390 Direct Access Storage see the 3390 Storage Subsystem Library (SSL). To receive a copy of the SSL manuals, which are helpful for VM customers, order SBOF-3122. To receive the entire 3990 storage control library, order GBOF-0366.

Programming Interfaces

This book is intended to help you use the VM support for 3390. Unless specifically stated otherwise, the information in this book must not be used for programming purposes. However, this book provides the following type of information and it is explicitly defined where it occurs:

General-Use Programming Interface

General-use programming interfaces are provided to allow you to write programs that use the services of VM.

End of General-Use Programming Interface

Contents

Chapter 1. Introduction

The IBM 3390 is a high-speed disk storage device designed for online data storage in a wide range of data storage applications. It offers quick, reliable access to data, and is a cost-effective way to expand your online storage capabilities.

The 3390 provides improvements in performance and storage capacity, and a considerable savings in floor space when compared to previous IBM direct access storage products. The 3390 also provides improved reliability, availability, and serviceability through the use of a new head-disk assembly (HDA) and advanced circuit technology.

3390 Models

The 3390 is available in two models, Model 1 and Model 2, offering different data capacities per device. 3390 Model 2 units have a capacity of 1.89 gigabytes (GB)[1] per device (2226 cylinders), and 3390 Model 1 units have a capacity of 0.946 gigabytes (GB)[1] per device (1113 cylinders).

There are two HDAs in an A14, A24, B14, or B24 unit, four HDAs in an A18, A28, B18, or B28 unit, and six HDAs in a B1C or B2C unit. An HDA contains two, separately addressable devices, and a full 3390 string (consisting of one A-unit and two B-units) can contain up to 32 devices. A maximum of two 3390 strings can be attached to a 3990 storage control, for a total of 64 device addresses in the storage subsystem.

Nondisruptive Installation and Removal

When the 3390 is attached to a properly configured 3990 Storage Control, you can add or remove 3390 B-units to or from a 3390 string without disrupting operation or availability of existing devices. A second 3390 string can be attached to or removed from the 3990 Storage Control without disrupting the operation or availability of the other string.

3390 Operating Modes

3390 devices can be set to function in one of two modes: 3390 mode or 3380 track compatibility mode. In 3390 mode, the entire track capacity of the 3390 device is available for use. In 3380 track compatibility mode, devices are formatted so the tracks have the same capacity as 3380 tracks. This enables programs with device-dependent data to take advantage of the 3390's performance characteristics without going through a conversion process.

Regardless of the mode and device capacity, the number of cylinders per device is constant (1113 for Models A14, A18, B14, B18, and B1C and 2225 for Models A24, A28, B24, B28, and B2C).

3380 track compatibility mode allows programs with 3380 track dependencies to store and retrieve information from 3390 devices operating in 3380 track compatibility mode. Even if the 3390 device is operating in 3380 track compatibility mode, the performance of the 3390 device is better than a 3380 device, due to the 3390's higher data rate, faster seek times, and lower rotational delay (latency).

[1] GB equals 10^9 bytes.

The data capacity for 3390 models when all devices in the unit are in the same mode (either 3390 mode or 3380 track compatibility mode) is shown in Table 1. Figures shown in the table assume use of standard record zero (R0) and maximum record one (R1).

Selecting 3390 mode or 3380 track compatibility mode is done on a **device** level, using the Device Support Facilities INSTALL command. This means that within a single A-unit or B-unit, any individual device can be in either mode.

Table 1. 3390 Data Capacity: 3390 Mode and 3380 Track Compatibility Mode

Model	**Devices**	**Data Capacity 3390 Mode**	**Data Capacity 3380 Track Compatibility Mode**
A14, B14	4 devices (2 HDAs)	3.78 GB	3.17 GB
A24, B24	4 devices (2 HDAs)	7.56 GB	6.34 GB
A18, B18	8 devices (4 HDAs)	7.56 GB	6.34 GB
A28, B28	8 devices (4 HDAs)	15.13 GB	12.68 GB
B1C	12 devices (6 HDAs)	11.35 GB	9.51 GB
B2C	12 devices (6 HDAs)	22.70 GB	19.02 GB

Configuring for Capacity

Although an elaborate capacity planning study is not required to use 3390's, you should have a high-level understanding of space utilization in your current hardware configuration before installing the 3390. By understanding your current space needs, you can plan for orderly growth and reduce those performance problems that accompany insufficient space.

Table 2 summarizes the data capacity of the 3390 models and the various 3380 models in terms of blocks and approximate megabytes (MB)[2]. The figures given for 3390 are for 3390 mode. All figures assume IBM standard record zero (R0), and records without keys.

Table 2. Summary of VM Data Capacity for DASD Models (Blocks Without Keys)

DASD Model	Capacity per	512-byte Blocks	1024-byte Blocks	2048-byte Blocks	4096-byte Blocks
3390 A14, B14	Volume	818 055 blks 418 MB	550 935 blks 564 MB	350 595 blks 718 MB	200 340 blks 820 MB
	4-volume unit	3 272 220 blks 1 675 MB	2 203 740 blks 2 256 MB	1 402 380 blks 2 872 MB	801 360 blks 3 282 MB
3390 A18, B18	Volume	818 055 blks 418 MB	550 935 blks 564 MB	350 595 blks 718 MB	200 340 blks 820 MB
	8-volume unit	6 544 440 blks 3 350 MB	4 407 480 blks 4 513 MB	2 804 760 blks 5 744 MB	1 602 720 blks 6 564 MB
3390 B1C	Volume	818 055 blks 418 MB	550 935 blks 564 MB	350 595 blks 718 MB	200 340 blks 820 MB
	12-volume unit	9 816 660 blks 5 026 MB	6 611 220 blks 6 769 MB	4 207 140 blks 8 616 MB	2 404 080 blks 9 847 MB
3390 A24, B24	Volume	1 636 110 blks 837 MB	1 101 870 blks 1 128 MB	701 190 blks 1 436 MB	400 680 blks 1 641 MB
	4-volume unit	6 544 440 blks 3 350 MB	4 407 480 blks 4 513 MB	2 804 760 blks 5 744 MB	1 602 720 blks 6 564 MB
3390 A28, B28	Volume	1 636 110 blks 837 MB	1 101 870 blks 1 128 MB	701 190 blks 1 436 MB	400 680 blks 1 641 MB
	8-volume unit	13 088 880 blks 6 701 MB	8 814 960 blks 9 026 MB	5 609 520 blks 11 488 MB	3 205 440 blks 13 129 MB
3390 B2C	Volume	1 636 110 blks 837 MB	1 101 870 blks 1 128 MB	701 190 blks 1 436 MB	400 680 blks 1 641 MB
	12-volume unit	19 633 320 blks 10 052 MB	13 222 440 blks 13 539 MB	8 414 280 blks 17 232 MB	4 808 160 blks 19 694 MB
3380 AK4, BK4	Volume	1 831 950 blks 937 MB	1 234 575 blks 1 264 MB	716 850 blks 1 468 MB	398 250 blks 1 631 MB
	4-volume unit	7 327 800 blks 3 751 MB	4 938 300 blks 5 056 MB	2 867 400 blks 5 872 MB	1 593 000 blks 6 524 MB
3380 AE4, BE4	Volume	1 221 300 blks 625 MB	823 050 blks 842 MB	477 900 blks 978 MB	265 500 blks 1 087 MB
	4-volume unit	4 885 200 blks 2 501 MB	3 292 200 blks 3 371 MB	1 911 600 blks 3 914 MB	1 062 000 blks 4 349 MB
3380 A04, AA4, B04, AD4, BD4, AJ4, BJ4	Volume	610 650 blks 312 MB	411 525 blks 421 MB	238 950 blks 489 MB	132 750 blks 543 MB
	4-volume unit	2 442 600 blks 1 250 MB	1 646 100 blks 1 685 MB	955 800 blks 1 957 MB	531 000 blks 2 174 MB

[2] MB equals 10^6 bytes.

Chapter 2. Installation and Service

The following information will help you install the 3390 in your VM environment.

Defining 3390 to the System

Before you can use the 3390, you must identify its existence and location to VM. Do this before you install the 3390 units, so that you can test the configuration definition in advance.

Defining the 3390 to the operating system includes the following steps:

- Defining the real I/O configuration to Virtual Machine/Extended Architecture™ System Product (VM/XA™ (in HCPRIO)
- Defining the units to the processor by running the I/O Configuration Program (IOCP), when necessary.

These steps are described in more detail in the following sections.

Defining the Real I/O Configuration to VM/XA SP

The real I/O configuration file (called HCPRIO in VM/XA SP) consists of the RDEVICE macro, which describes the I/O devices attached to the real processor. Because VM uses this information to schedule I/O and to allocate resources, the macro entries in the real I/O configuration file must accurately represent the real I/O hardware configuration.

If you are adding new devices or changing hardware configurations, you must update the real I/O configuration file to include the new devices. Note that once the real I/O configuration file is updated to reflect the new devices, the corresponding device control blocks occupy space in real storage, regardless of whether the devices are actually installed. If you have severe real storage limitations, you may want to delay updating the real I/O configuration file until shortly before device installation.

For VM/XA SP systems, the RDEVICE macro is used to update the real I/O configuration file.

Note: The following sections are meant to be a summary of the most frequently used parameters. For complete information on parameter combinations and restrictions, see *VM/XA SP Planning and Administration.*

Coding the RDEVICE Macro for VM/XA SP

You must code an RDEVICE macro for each I/O device (or group of devices with contiguous device numbers) in the configuration. The relevant RDEVICE parameters are as follows:

DEVNO = (*rdevno,nnn*)
For VM/XA SP systems, *rdevno* is a 4-digit hexadecimal device number from 0000 to FFFF. The digits are assigned to represent a specific device or volume on a 3390 unit.

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

nnn is the number of RDEVs to be generated (one for each volume in the string). Use 32 for a full 3390 four-path string, and use 64 for two full 3390 four-path strings. For a 3390 string that is not full, specifying 32 gives you the flexibility to expand the string in the future without having to recode the RDEVICE macro.

DEVTYPE = *type*
type is the type of device.

DEVTYPE = 3390 is used for all models of the 3390. To define a 3390 you plan to use in 3380 track compatibility mode, you can specify either **DEVTYPE** = 3390 or **DEVTYPE** = 3380. Specifying **DEVTYPE** = 3390 gives you the flexibility to change modes in the future without having to recode the RDEVICE macro.

Note: Flexibility does not apply for the IPL device. If the IPL device is operating in 3390 mode, specify **DEVTYPE** = 3390 (in the RDEVICE macro), and **SYSTYPE** = 3390 (in the SYSRES macro, located in the HCPSYS module). If the IPL device is operating in 3380 track compatibility mode, you must specify **DEVTYPE** = 3380 and **SYSTYPE** = 3380, respectively.

For more information about HCPSYS, see *VM/XA SP Planning and Administration.*

SHARED = YES|NO
For VM/XA SP systems, SHARED indicates whether the device should be defined as shared among systems.

Sample RDEVICE macros are shown in Figure 2 on page 7 through Figure 4 on page 8.

Examples of Updating HCPRIO

The installation examples shown in this section assume the use of VM/XA SP2.

Defining the 3390 Four-Path String to HCPRIO

Figure 1 on page 7 shows a basic 3990-3390 configuration that includes one 3390 A-unit and two B-units. (The corresponding IOCP statements to define the 4-path string are shown in Figure 5 on page 9.)

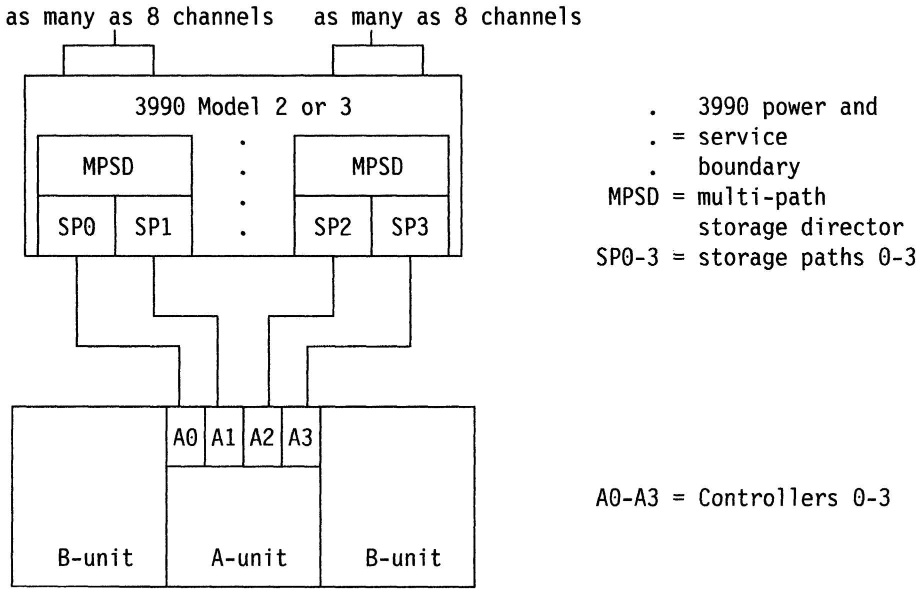

Figure 1. Single 3390 String Configuration Example

Figure 2 shows the RDEVICE statements used to define the configuration to VM/XA SP in the HCPRIO file. In this example, the second RDEVICE statement reserves an additional 32 addresses for a future nondisruptive DASD installation.

```
* define 32 DASD devices
         RDEVICE DEVNO=(0140,32),DEVTYPE=3390
* reserve addresses for 32 additional devices
         RDEVICE DEVNO=(0160,32),DEVTYPE=3390
```

Figure 2. Sample HCPRIO File Defining a 4-Path 3390 String

Defining a 3390 Four-Path and 3380 Two-Path String to HCPRIO

Figure 3 on page 8 shows an intermixed configuration containing:

- A 4-path string of 3390s
- Two, 2-path strings of 3380s (AD4/AE4 and AJ4/AK4 units)
- A 3990 Model 2 or 3 Storage Control.

(The corresponding IOCP statements to define this configuration are shown in Figure 6 on page 9.)

Notes:

1. The 3990 Storage Control must be set to DLSE mode by your IBM service representative.
2. 3390 Attachment Feature (6120) is required on 3990 Model 2 or 3.
3. Each 3380 AJ4 or AK4 string may contain as many as three BJ4 or BK4 units (any combination).
4. Each 3380 AD4 or AE4 string may contain as many as three BD4 or BE4 units (any combination).

5. The cabling shows the 3390 controllers A0-A3 sequentially connected to storage paths 0-3, as required.

6. 3380 AA4 strings are not supported in this configuration.

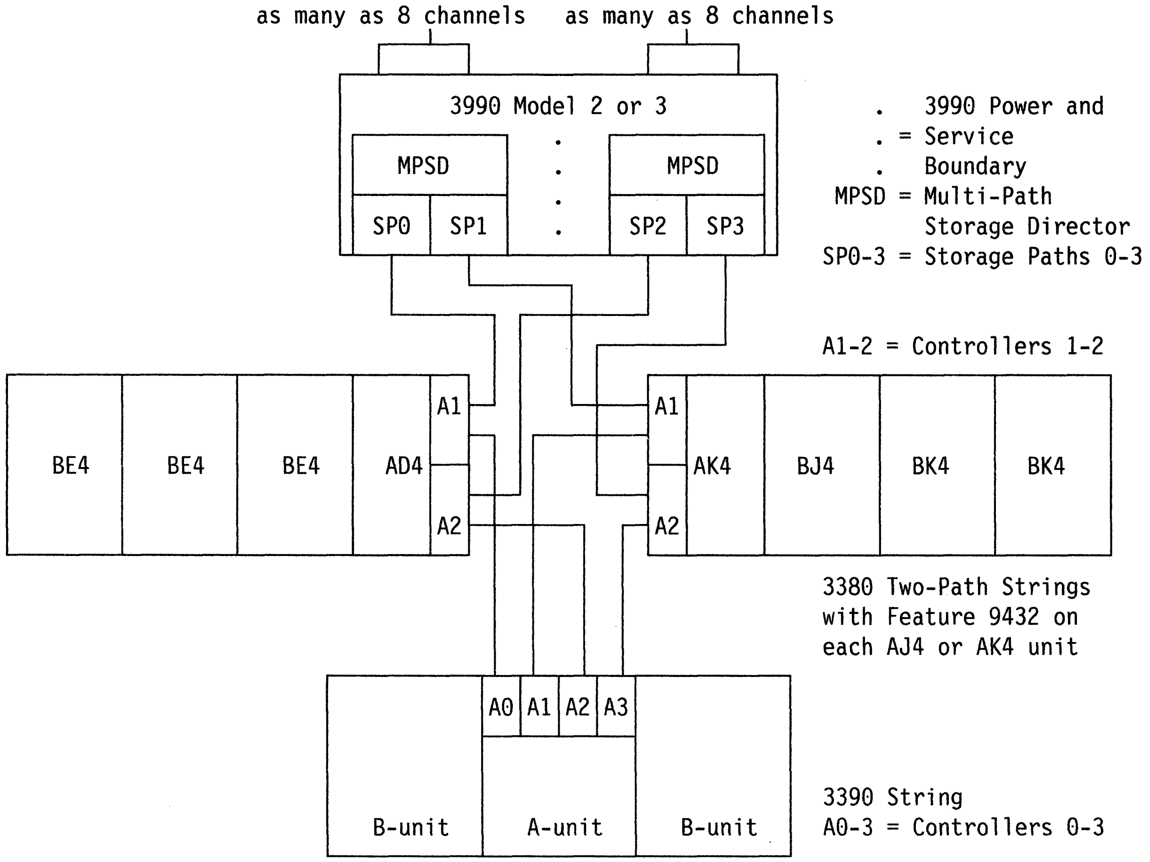

Figure 3. 3990 with 3390 Four-Path String and Two 3380 Two-Path Strings

Use the RDEVICE statements shown in Figure 4 to define the intermixed configuration to VM/XA SP in the HCPRIO file.

```
* define the two 3380 2-path strings
         RDEVICE DEVNO=(C80,16),DEVTYPE=3380
         RDEVICE DEVNO=(C90,16),DEVTYPE=3380
* define the 3390 4-path string
         RDEVICE DEVNO=(CA0,32),DEVTYPE=3390
```

Figure 4. Sample HCPRIO File Defining an Intermixed Four-Path and Two-Path String to VM/XA SP

For details on how to update HCPRIO, see *VM/XA SP Installation and Service.*

Defining the I/O Configuration to the Processor (VM/XA SP)

You must run the I/O Configuration Program (IOCP) to define the I/O configuration to the processor if you are adding new devices or changing hardware configurations.

Invoke the CMS version of IOCP to generate a new input/output configuration data set.

Defining the 3390 Four-Path String with IOCP

Figure 5 shows how the configuration defined to HCPRIO by the statements in Figure 2 on page 7 can be defined in the IOCP source file. An additional 32 addresses have been reserved for a future nondisruptive DASD install.

```
* define channel paths from each CPU
         CHPID PATH=((01),(07),(11),(17)),TYPE=BL
* define the first storage director of the 3990
         CNTLUNIT CUNUMBR=001,PATH=(01,11),PROTOCL=S4,SHARED=N,        -
               UNIT=3990,UNITADD=((40,64))
* define the second storage director of the 3990
         CNTLUNIT CUNUMBR=002,PATH=(07,17),PROTOCL=S4,SHARED=N,        -
               UNIT=3990,UNITADD=((40,64))
* define the installed DASD addresses
         IODEVICE ADDRESS=(140,32),CUNUMBR=(001,002),UNIT=3390
* reserve addresses for future non-disruptive installation
         IODEVICE ADDRESS=(160,32),CUNUMBR=(001,002),UNIT=3390
```

Figure 5. IOCP Generation for a 3390-3990 Configuration

Defining an Intermixed Four-Path and Two-Path String with IOCP

Figure 6 shows how the configuration defined to HCPRIO by the statements in Figure 4 on page 8 can be defined in the IOCP source file.

Note: The same IOCP statements used to define a 3390 with 3380 2-path intermix can be used to define a 3390 with 3380 4-path intermix.

```
*
         CHPID PATH=((01),(07),(21),(27)),TYPE=BL
         CNTLUNIT CUNUMBR=008,PATH=(01,21),PROTOCL=S4,SHARED=N,        -
               UNIT=3990,UNITADD=((80,64))
         CNTLUNIT CUNUMBR=009,PATH=(07,27),PROTOCL=S4,SHARED=N,        -
               UNIT=3990,UNITADD=((80,64))
         IODEVICE ADDRESS=(C80,16),CUNUMBR=(008,009),UNIT=3380
         IODEVICE ADDRESS=(C90,16),CUNUMBR=(008,009),UNIT=3380
         IODEVICE ADDRESS=(CA0,32),CUNUMBR=(008,009),UNIT=3390
*
```

Figure 6. IOCP Generation for Intermixed 3390 (Four-Path) and 3380 Two-Path String

Chapter 3. Procedure for Switching 3390 Operating Modes

Changing 3390 operating modes is a task that requires some consideration and scheduling. Consider the times when the least amount of activity occurs and choose the time that has the least impact on your users.

3390 Mode Switch Considerations

All mode switches require the affected device to be reformatted, so you should move the data off the device before beginning the procedure.

Depending on the procedure, you must either re-IPL or vary the affected device offline so that CP recognizes the new device type. Otherwise, the integrity of the control blocks associated with the device is jeopardized.

If the device is defined as a *3380* in the real I/O configuration file, it is necessary to update the RDEVICE macro and define the device as a *3390*. Although it is not a requirement, for consistency the device type can also be specified as 3390 in the I/O Configuration Dataset (IOCDS). Once the updates are made, you must shut down the VM system and re-IPL, using the new information in the real I/O configuration file and the IOCDS. That should be done at a convenient time before the actual mode switch is done.

If you are switching from 3380 track compatibility mode to 3390 mode, and the device type is not defined as a *3390* in the real I/O configuration file, VM does not allow the device to be varied back online in step 7. The device is then unusable until after the next IPL.

Mode Switch Procedure

The following procedure is common to both VM operating environments.

1. Prevent all applications and users from using the device. For example, if the device is being used by the system for minidisks, all users must detach those minidisks, and the device must be detached from the system. If the device is simply being used by one guest, that guest must detach the device.
2. Move the data off the affected device. The procedure of switching modes destroys the data on the device and changes the track format.

   If you wish to restore the data to the same device after switching modes, you should not use service programs that require identical track formats (i.e. DASD Dump Restore (DDR)). For information on moving data, see *Using 3390 Direct Access Storage in a VM Environment.*
3. Vary the device offline to all processors that have access to it, except the one from which you perform the mode switch.
4. Attach the device to the virtual machine of the person who is performing the mode switch; typically, this person is the system programmer.
5. Use the *SETMODE* parameter of the ICKDSF INSTALL command to change the mode. For information about using ICKDSF for this operation, see *Using 3390 Direct Access Storage in a VM Environment.*

6. Reformat the affected device when all mode switches are made being sure that you have moved the data off the device before you begin. Use the CPFORMAT/CPFMTXA utility to reformat the device when the mode has been switched. Any CMS minidisks on the device must also be formatted using CMS FORMAT. See *VM/XA SP CP Command Reference* for a description of the CPFORMAT command and *VM/XA SP CMS Command Reference* for the CPFMTXA and FORMAT commands.

7. Detach the device from the virtual machine of the person who performed the mode switch.

8. If the mode switch is:

   - **From** 3380 track compatibility mode **to** 3390 mode, you must re-IPL the system.
   - **From** 3390 mode **to** 3380 track compatibility mode, vary the device offline and then back online.

   For information on reformatting the volume for a particular use (CP and CMS minidisks), and moving CP-owned volumes and CMS minidisks, see *Using 3390 Direct Access Storage in a VM Environment.*

   Keep in mind that the device type you use when restoring data is different from the one you used to back up the data in step 2.

9. Once the data is restored, attach the device again to the system or guest and allow applications to continue.

Chapter 4. DASD Dump Restore (DDR)

The following describes the DDR utility to be used with the 3390 DASD. Some of these changes are for support of the new device, 3390, but many other improvements are also implemented with this support.

Running the DDRXA Program

DDRXA runs as a stand-alone program, either on a real 370-XA processor or in a 370-XA virtual machine. DDRXA also runs as a CMS module. You may use it to:

- Dump data from a DASD to tape
- Restore dumped data from a tape back to a DASD
- Copy data from DASD to DASD or from tape to tape
- Print or display records from DASD or tape.

You tell the program what to do through DDRXA control statements. You may either enter the control statements from your console or supply them from the device on which you load the program (the IPL device).

Before You Begin

Before you run DDRXA, you must know its location. During system generation, your installation puts a copy of the program either on tape or in a file on DASD. To find out where the program is, contact the person at your installation who generates your VM/XA SP system.

Also, decide how you want to submit the DDRXA control statements. You may either enter the statements from your console, or supply them from the device on which you load the program (the IPL device). For more information, see "Supplying DDRXA Control Statements" on page 16.

Next, decide how you want to run the program. To run the program on a real processor, see the next section. To run it in a virtual machine, see "Running DDRXA in a 370-XA Virtual Machine" on page 14. To run it as a CMS module see "Running DDRXA as a CMS Module" on page 16.

Finally, before you restore or copy important data to a DASD volume, use the Device Support Facilities program to inspect the volume for defective tracks. For more information, see *VM/XA SP Real System Operation.*

Invocation

The DDRXA utility is changed to allow the use of the XF parameter, which specifies the Improved Data Recording Capability feature.

After DASD Dump/Restore is invoked, this utility prompts you for the INPUT and OUTPUT device information. Refer to page 19 for additional information on INPUT/OUTPUT statements.

Interactions

There are no new interactions introduced by this support. The existing interactions are changed to allow exploitation of the tape formats provided by the new tape devices.

Running DDRXA on a Real Processor

To run DDRXA on a real processor, you first must ensure that the program is loadable and then load it.

Making DDRXA Loadable on a Real Processor: If DDRXA resides on a tape, it is already loadable.

If the DDRXA program resides in a file on a DASD volume, do one of the following:

- Put a copy of the program on tape.
- Create a card deck of the program.

To put a copy of the program on tape, use the CMS FILEDEF and MOVEFILE commands. For information about how to use the FILEDEF and MOVEFILE commands, see *VM/XA SP Real System Operation.*

To create a card deck of the program, first enter the CMS PUNCH command with the NOHEADER option. Then punch the resulting spool file on a real card punch.

Loading DDRXA on a Real Processor: You can now load the program on the real processor:

1. If necessary, shut down the VM/XA SP system.
2. Make sure the real processor is ready to run in 370-XA mode.
3. If DDRXA resides on a tape, mount the tape on a 3420, 3422, 3430, or 3480 tape drive. If it resides in a card deck, load the deck into a real card reader.
4. Using your processor complex's system console, IPL the DDRXA program.

You are now ready to start running the program. See "Supplying DDRXA Control Statements" on page 16.

Running DDRXA in a 370-XA Virtual Machine

To run DDRXA in a virtual machine, you must first ensure that the program is loadable and then load it.

Making DDRXA Loadable in a Virtual Machine: If DDRXA resides on tape, it is already loadable.

If the DDRXA program resides in a file on a DASD volume, do one of the following:

- Dump a copy of the program to tape.
- Place a copy of the program in your virtual reader.

To dump the DDRXA program to tape, use the CMS FILEDEF and MOVEFILE commands. For information about how to use the FILEDEF and MOVEFILE commands, see *VM/XA SP Real System Operation.*

To place a copy of the DDRXA program in your virtual reader:

1. Make sure your virtual machine is simulating System/370 architecture. Enter:

   ```
   set machine 370
   ```

2. If DDRXA does not reside on the CMS system volume, link to the minidisk that contains the program.
3. Bring up CMS in your virtual machine. Enter:

   ```
   ipl 190
   ```

   or, if your installation has installed CMS in a named saved system, enter:

   ipl *cmsname*

   where *cmsname* is the name of your CMS system.
4. Route punch spool files to your virtual reader. Enter:

   ```
   spool punch *
   ```

5. Punch a copy of the DDRXA program and name the resulting punch file IPL DDRXA. Enter:

   ```
   punch ipl ddrxa * (noheader
   ```

You now have a copy of the DDRXA program in your virtual reader.

Loading DDRXA in a Virtual Machine: You are now ready to load the program in your virtual machine.

First, make sure your virtual machine is simulating 370-XA architecture. Enter:

```
set machine xa
```

Loading DDRXA from Tape: If DDRXA resides on a tape:

1. Mount the DDRXA tape on a 3420, 3422, 3430, or 3480 tape drive and attach the tape drive to your virtual machine as virtual device number 181.

   To attach the tape drive to your virtual machine, enter:

   attach *rdev* *** 181**

   where *rdev* is the real device number of the tape drive.
2. Load the DDRXA program into your virtual machine storage. Enter:

   ```
   ipl 181
   ```

You are now ready to start running the program. See "Supplying DDRXA Control Statements" on page 16.

Loading DDRXA from Your Virtual Reader: If DDRXA resides in your virtual reader:

1. Make the file ready to load, and prevent CP from purging it from your reader. Enter:

   change reader *spoolid* **nohold keep**

   where *spoolid* is the spool file identification number that CP assigned the spool file when it placed it in your reader.

2. Put the DDRXA file first in your reader queue. Enter:

   ```
   order reader spoolid
   ```

   where *spoolid* is the spool file identification number of the DDRXA reader file.

3. Make sure the reader can process the DDRXA file. Enter:

   ```
   spool vdev class *
   ```

   where *vdev* is the virtual device number of your virtual reader.

4. Load the DDRXA program into storage. Enter:

   ```
   ipl vdev
   ```

   where *vdev* is the virtual device number of your virtual reader.

You are now ready to start running the program.

Running DDRXA as a CMS Module

To run DDRXA as a CMS module, you must have the disk accessed which has the DDR MODULE on it.

To invoke the DDRXA program, enter:

DDR

In response the following prompt will appear:

```
VM/XA SYSTEM PRODUCT DASD DUMP/RESTORE PROGRAM
ENTER CARD READER ADDRESS OR CONTROL STATEMENTS
ENTER:
```

You can now begin to enter the DDRXA control statements required to perform the desired task.

Supplying DDRXA Control Statements

You may supply control statements to DDRXA in one of three ways.

- You may enter DDRXA control statements from your console after you invoke the program.
- You may supply them from the device on which you load the program (the IPL device). This option requires you to set up the control statements before you invoke the program.
- You may supply them from a CMS file. This option requires that you run DDRXA as a CMS module.

Entering DDRXA Control Statements from Your Console

Normally, you enter DDRXA control statements interactively from your console.

As soon as you load it into storage, DDRXA begins executing and sends you the following initial prompt:

```
VM/XA SYSTEM PRODUCT DASD DUMP/RESTORE PROGRAM
ENTER CARD READER ADDRESS OR CONTROL STATEMENTS
ENTER:
```

If you do not receive this prompt, press the ENTER key without any data (enter a null line). Entering a null line tells DDRXA your console's device number, so that it can send you the prompt. (DDRXA initially assumes that your console has device number 0009 or 001F.) You may then start entering control statements interactively.

Note: When you run DDRXA on a real processor, and you have a nonconsole device at either real device number 0009 or 001F, make sure the device is not operational. Otherwise, results are unpredictable.

In response to the initial prompt, enter the first statement for the first function. Entering statements for different functions is discussed in more detail in "DDRXA Control Statements" on page 18. However, when you enter the statements, remember that:

- DDRXA reads only the first 71 characters of data per statement.
- DDRXA ignores any data you enter after the last possible operand of a statement.

After you enter the first one, DDRXA prompts you for additional statements. When you have entered all the statements for the first function, enter a null line if needed. Null lines delimit groups of statements for DUMP, COPY, and RESTORE functions. You do not need to enter a null line for PRINT or TYPE functions.

DDRXA performs the first function and prompts you for another statement. Enter the statements for the next function in the same manner.

To end the program after DDRXA has completed the last function, enter a null line. The program then ends.

Supplying DDRXA Control Statements from the IPL Device

Normally, you enter DDRXA control statements interactively after you IPL the program. However, if DDRXA resides on a DASD volume, you can direct DDRXA to read the control statements from the end of the file you IPL. To do this, you must set up the program and control statements before you IPL, as follows:

1. Edit the DDRXA program and put the control statements at the end of the file.
2. Transfer the program to tape, a card deck, or your virtual reader as usual.

The control statements transfer along with the program. Then you can load the program on the real processor (using the tape or the card deck) or in your virtual machine (using the tape or your virtual reader) as discussed previously.

After you load the program and receive the initial prompt, press the ENTER key. DDRXA then reads all the control statements, completes the requested functions, and ends.

Supplying DDRXA Control Statements from a CMS File

First create a CMS file with the desired control statements in it. Then enter HCPDDR *Fn Ft Fm*. *Fn* is the file name, *Ft* is the file type, and *Fm* is the file mode of the file created to contain the DDRXA control statements. Specifying the file mode is optional.

Rules for Supplying DDRXA Control Statements from the IPL Device or a CMS file

When you prepare DDRXA control statements for entry from a tape or from a card reader, keep in mind the following rules:

- DDRXA reads only columns 1 to 71.
- DDRXA ignores any data you enter after the last possible operand of a statement.
- Each group of statements you include must define a single function: DUMP, RESTORE, COPY, PRINT, or TYPE.
- If you want DDRXA to perform the same function on different DASD extents, you must include a statement for each extent. You may not include more than 20 extent statements for the same function.

  For example, suppose you want to dump to tape cylinders 1 through 3, 5, 7 and 8, 10, 14, and 20 of a DASD. To do this, you must include six DUMP function statements, one for each noncontiguous set of cylinders (each extent).
- If you enter an incorrect statement, or if you enter the function statements in the wrong order, DDRXA sends you an error message. However, DDRXA continues to scan any remaining function statements for syntax errors.
- By default, DDRXA uses device number 00E as a printer. If you do not have a printer with device number 00E, use a SYSPRINT statement to define your printer. If the device specified on the SYSPRINT statement does not exist, DDRXA sends you an error message but checks any remaining statements for syntax errors.
- Start each group of statements that defines a DUMP, COPY, or RESTORE function with an INPUT or an OUTPUT statement.

  INPUT and OUTPUT statements delimit the different functions you want DDRXA to perform on this run. When DDRXA reads an INPUT or OUTPUT statement that follows one or more function statements, it performs the function described by the previous group of statements. It treats any function statements that follow as separate function requests.

DDRXA Control Statements

To tell DDRXA the I/O devices to use and the functions to perform, use DDRXA control statements. There are two types of statements:

- I/O definition statements (INPUT, OUTPUT, and SYSPRINT)
- Function statements (DUMP, COPY, RESTORE, PRINT, and TYPE).

This section describes the INPUT/OUTPUT statements′ syntax in detail. The syntax for the other statements is described in detail in *VM/XA SP Real System Operation.* The descriptions use the notational conventions found in *VM/XA SP CP Command Reference.* When a list of choices appears within braces ({ }), you must select one of the operands. When a list of choices appears within brackets ([ ]), you may either select one of the operands or omit the operand altogether.

INPUT/OUTPUT Statements

You must include an INPUT or OUTPUT statement to describe each tape drive and DASD volume you use. The format of the INPUT/OUTPUT statement is:

```
 { INput  }  { vdev } type [ volid   ] (options...)
 { OUTput }  { rdev }      [ SCRATCH ]
                           [ altape  ]

             Options:

             [ SKip nn ] [ MOde 6250 ] [ REWind ] [COmpact] [ EMSG  ]
             [ SKip  0 ] [ MOde 62   ] [ UNload ]           [ ESKIP ]
                         [ MOde 1600 ] [ LEave  ]
                         [ MOde 16   ]
                         [ MOde 800  ]
                         [ MOde 80   ]
                         [ MOde 38k  ]
                         [ MOde XF   ]
```

Note: The options are valid for tape drives only.

To reset the options for a tape device you defined on an INPUT/OUTPUT statement you previously entered, use this format:

```
 { INput  }  (options...)
 { OUTput }
```

where:

INput
specifies an input device.

OUTput
specifies an output device.

vdev
is the virtual device number of the device. Use this operand if you are running DDRXA in a virtual machine.

rdev
is the real device number of the device. Use this operand if you are running DDRXA on a real processor.

type

is one of the following device types:

2305-2
3330
3330-11
3340-35
3340-70
3350
3375
3380
3390
DASD
3420
3422
3430
3480
TAPE.

Note: DDRXA does not support the 7-track feature of any tape device. It also does not support the 3850 Mass Storage System.

The following table shows you how to specify some other types of input or output devices.

Type of Device	Specify as:
3333	3330
3340-70F	3340-70
3344	3340-70
3350 in 3330-1 compatibility mode	3330
3350 in 3330-11 compatibility mode	3330-11
3350 in native mode	3350
3390 in 3380 track compatibility mode	DASD or 3380
3390 in native mode	DASD or 3390

volid

is the volume identifier of a DASD device.

SCRATCH

specifies that DDRXA is not to verify the volume identifier of a DASD.

altape
is the real device number of an alternate tape drive.

If DDRXA requires more than one tape and you do not specify the *altape* operand, DDRXA displays the following prompt at the end of the tape:

```
END OF CYL xxxx HD xx, MOUNT NEXT TAPE
```

After you mount the next tape, DDRXA continues.

SKip
The SKIP option of the INPUT/OUTPUT control statement is **always** relative to the beginning of the tape. Tape is rewound anytime a tape is opened for either INPUT or OUTPUT. The SKIP count (*n*) determines how many files are skipped. The SKIP count can be from 0 to 255.

> **Caution: Starter Systems for VM/XA**
>
> Since the SKIP option is relative to the beginning of the tape, the starter system data is located using SKIP 2 on the INPUT/OUTPUT control statement (skipping two files: standalone ICKDSF and standalone IPL DDR).

MOde 6250
MOde 62
MOde 1600
MOde 16
MOde 800
MOde 80
MOde 38K
MOde XF
is the tape recording format. MODE values can be 800, 1600, or 6250 for 3420 devices, and 38K or XF for 3480 devices. This specifies the density of all output tapes that DDRXA opens for the first time and that are at the load point. DDRXA also sets all subsequent tapes to the specified density.

If you do not specify a MODE option, DDRXA does not reset the density. Therefore, the density setting remains as it was the last time you mounted a tape on this tape drive.

For a 3420, you may specify recording densities of 6250, 1600, or 800 bytes per inch. You may shorten these numbers to 62, 16, or 80, respectively.

For a 3422 or 3430, you may specify recording densities of 6250 or 1600 bytes per inch. You may shorten these numbers to 62 or 16, respectively.

For a 3480, you may specify 38K or XF.

The default for 3480 tapes is 38K.

XF is valid only for 3480 tape drives that support the Improved Data Recording Capability (IDRC) feature. If the user specifies both MODE XF and the COMPACT option, the software compaction option will always be ignored.

REWind
rewinds the tape at the end of a function.

UNload
rewinds and unloads the tape at the end of a function.

LEave
leaves the tape positioned at the end of the file at the end of a function.

Note: If you mount the wrong input tape, DDRXA displays an error message. Also, the tape rewinds and unloads, regardless of the REWIND, UNLOAD, or LEAVE setting.

COmpact
writes the output tape in compact format, using less space than output written in standard format. DDRXA accomplishes this by compressing strings of duplicate data into a smaller amount of space. This option is valid only on the OUTPUT control statement for the DUMP function. You may use tapes in compact format as input for the RESTORE, COPY, PRINT, and TYPE functions.

If the user specifies both MODE XF and the COMPACT option, the software compaction option will always be ignored.

EMSG
requests notification of permanent errors or missing records and allows the operator the option to bypass them if requested. This option is valid only for input devices.

ESKIP
any track in error or any missing record is bypassed and the job continues without operator notification. This option is valid only for input devices.

Usage Notes

1. DASD may be specified as the device type for 3375, 3380, or 3390 in the INPUT/OUTPUT control statement. TAPE may be specified as the device type for newer tape devices. If DASD or TAPE is specified, DDR determines the specific device type. If the device type cannot be determined (which is the case for some older devices), DDR requests input of the device type.
2. Two new options EMSG and ESKIP have been added to the INPUT control statement. These options allow notification/bypass of permanent errors or missing records on the input device.
3. The SKIP option of the INPUT/OUTPUT control statement has been changed to *always* be relative to the beginning of the tape.
4. Any time an operator response of YES, NO, or REREAD is requested, a response of Y, N, or R is allowed.
5. Whenever an operator depresses the ENTER key while running stand-alone (not under CMS), a message will be produced indicating the last track processed during a PRINT, DUMP, COPY, or RESTORE operation.
6. DDRXA can be run as a module under CMS.

   Build the DDRXA module by issuing UTILITY GEN DDR. After building, issue the command *DDR* to run the DASD Dump Restore program.

   Note: Like the IPL DDR program, the DDRXA module needs text decks for the following:

   HCPDDR
   HCPDNC
   HCPDDT
   HCPDDC
   HCPDNT

Chapter 5. Changed CP Messages

This chapter describes the messages and other externals that have changed with the support for 3390.

0403I *rdev* **{SCU|CACHE|DASD|MEDIA}** *sev***ALERT, MT =** *tttt-mm* **SER =** *mmaa-bbbbbbb* **REFCODE =** *cccc-cccc-cccc* **[ID =** *id***]** **[VOLSER =** *volser***]** **[CCHH = X'***cccc hhhh***'] [REPEATED]**

Explanation: The DASD Storage Subsystem has detected an abnormal operational condition within the DASD subsystem that requires service attention.

In this message, the variables are as follows:

rdev The device used to report the condition

sev ACUTE]SERIOUS]MODERATE SERVICE

tttt-mm The machine type and model

mmaa-bbbbbbb The serial number of the failing device

cccc-cccc-cccc The reference code

cccc hhhh The address of the failing track (media SIMs).

REPEATED For non-media SIMS (DASD hardware) only. It is shown when the message is a repeat presentation of a previously reported SIM.

System Action: System operation continues.

Operator Response: Record all of the information displayed in the message and contact your system support personnel.

Programmer Response: Examine the EREP records for additional details of the failure. Refer to the hardware System Reference Library (SRL) manuals for sense data formats and descriptions. If a hardware problem is indicated, contact the appropriate hardware service personnel and provide them with the information displayed in the message.

Module: ERP

0705D **I/O ERROR** *{rdev|vdev}* **IRB** *irb* **SNS** *sense* **[INPUT** *bbcchh***|OUTPUT** *bbcchh***] CCW** *ccw* **DO YOU WISH TO CONTINUE? RESPOND YES OR NO:**

Note: If EMSG option of the INPUT control statement is used, HCPDDR705E is changed to HCPDDR705D to solicit a response from the operator to continue.

Explanation: An unrecoverable I/O error has been detected. The variables have the following meanings:

rdev|vdev The device in error

irb The interrupt response block from the error

sense The sense bytes, in hexadecimal, describing the error

bbcchh The address (00, cylinder, and head), in hexadecimal, where the error occurred on the input or output cylinder

ccw The channel command word from the error

System Action: The system waits for a response.

User Response: Respond with YES, Y, NO, or N where these have the following results:

YES,Y The job continues. The track in error or missing record is bypassed.

NO,N The job step terminates. If the output device is tape, an attempt is made to write a trailer label closing the output device. A cylinder map is printed describing all valid data that was dumped, restored, or copied to the point of error.

Module: DDR

0651I DASD *rdev* DEVICE ADAPTOR NOT OPERATIONAL WITH STORAGE PATH – *ssid.p-cc-dd*

Explanation: An error recovery routine encountered a device adaptor that has been made unavailable to the system by the storage control. This happens when the storage control has determined that the device adaptor is in need of service. The variables in the message are as follows:

rdev	The failing device.
ssid.p-cc-dd	The subsystem identifier, storage path number, controller identifier and device unit address of the affected hardware components

System Action: System operation continues.

Operator Response: Contact your IBM service representative.

Programmer Response: Examine the EREP outboard record (OBR) for additional details as to the cause of the failure. Refer to the hardware publications for sense data format and descriptions. If a hardware problem exists, contact your IBM Support Center personnel to diagnose and correct the hardware problem.

Module: ERP

0705E I/O ERROR {*rdev*|*vdev*} **IRB** *irb* **SNS** *sense* **[INPUT** *bbcchh*|**OUTPUT** *bbcchh*] **CCW** *ccw*

Explanation: An unrecoverable I/O error has been detected. The variables have the following meanings:

rdev|*vdev*	The device in error
irb	The interrupt response block from the error
sense	The sense bytes, in hexadecimal, describing the error
bbcchh	The address (00, cylinder, and head), in hexadecimal, where the error occurred on the input or output cylinder
ccw	The channel command word from the error

System Action: The operation terminates. If the output device is tape, an attempt is made to write a trailer label closing the output device. A cylinder map is printed describing all valid data that was dumped, restored, or copied to the point of error.

User Response: Retry the operation. If the error persists, contact your IBM Support Center personnel for hardware support.

Module: DDR

0711D VOLID READ IS *volid1* **[NOT** *volid2*] **DO YOU WISH TO CONTINUE? RESPOND YES, NO OR REREAD:**

Explanation: The variables in this message are as follows:

volid1	The volume serial number of the DASD or tape as read from the device specified by the control statement
volid2	The volume serial number from the input or output control statement

System Action: The system waits for a response.

User Response: Respond YES, Y NO, N, REREAD, or R where these have the following results:

YES,Y	The operation continues.
NO,N	If the input is from cards, the program terminates after scanning the remaining statements for syntax. Otherwise, the next statement is solicited from the console.
REREAD,R	The label of the volume specified is read again.

Note: A new volume may have been mounted in the interim.

Module: DDR

0714D RECORD *bbcchh* **NOT FOUND ON INPUT TAPE**
DO YOU WISH TO CONTINUE? RESPOND YES OR NO:

Note: If EMSG option of the INPUT control statement is used, message HCPDDR714E is changed to HCPDDR714D to solicit a response from the operator to continue.

Explanation: The record was not found on the tape. *bbcchh* refers to the address (bin, cylinder and head), in hexadecimal, of the missing track header record.

System Action: The system waits for a response.

User Response: Respond YES, Y, NO, or N where these have the following results:

YES,Y The job continues. The record that was not found is bypassed.

NO,N The job step is terminated. All data restored or copied to that point is valid. If the input is from cards the program is terminated after scanning the remaining statements for syntax. Otherwise, the next control statement is solicited from the console. Try using the COPY ALL or RESTORE ALL statement, or use the correct explicit cylinder operand.

Module: DDR

0714E RECORD *bbcchh* NOT FOUND ON INPUT TAPE

Explanation: The record was not found on the tape. *bbcchh* refers to the address (bin, cylinder and head), in hexadecimal, of the missing track header record.

System Action: The job step is terminated. All data restored or copied to that point is valid. If the input is from cards the program is terminated after scanning the remaining statements for syntax. Otherwise, the next control statement is solicited from the console.

User Response: Use the COPY ALL or RESTORE ALL statement, or use the correct explicit cylinder operand.

Module: DDR

0717D DATA DUMPED FROM *volid1* TO BE RESTORED TO *volid2*.
DO YOU WISH TO CONTINUE?
RESPOND YES, NO OR REREAD:

Explanation: The volume serial number from the dumped DASD volume (*volid1*) does not match the RESTORE to DASD volume serial number (*volid2*).

volid1 The DASD volume serial number of the input tape.

volid2 The volume serial number of the output DASD device that is to receive the data from *volid1*.

System Action: The system waits for a response.

User Response: Respond YES, Y, NO, N, REREAD, or R where these have the following results:

YES,Y The RESTORE function continues.

NO,N If the input is from cards, the program terminates after scanning the remaining statements for syntax; otherwise, the next statement is solicited from the console.

REREAD,R The input tape is backspaced to the start of the file, and the volume header label is reread.

Note: If the wrong input tape is mounted, replace the tape and respond REREAD.

Module: DDR

0719E INVALID FILENAME OR FILE NOT FOUND

Explanation: This message can appear only if the DDR utility is running under CMS. A filetype was not entered from the CMS command line, or the filename and filetype entered could not be found on the CMS disks currently accessed.

User Response: Either omit all operands on the CMS command line, defaulting to console input, or enter the proper filename, filetype, and, optionally, filemode, for the CMS file containing the input control statements.

Module: DDR

0720E ERROR IN *routine*

Explanation: *routine* is the name of the CMS routine in error from the first eight characters of the CMS parameter list. The CMS return code generated by the error is returned in the following manner:

- PRINTR—the CMS return code plus 100
- WAITRD—the CMS return code plus 200
- RDBUF—the CMS return code plus 300
- TYPE or TYPLIN—the CMS return code plus 400

System Action: If the input is from cards or a CMS file, the program terminates after scanning the remaining statements for syntax. Otherwise, the program is immediately terminated.

User Response: Correct the error as indicated by the return code, and resubmit the job.

Module: DDR

Chapter 6. CP Programming Services

This section describes:

- MRIODATD monitor record fields for 3390
- Usage changes for DIAGNOSE X'24'
- A new instruction DIAGNOSE X'0210'
- Directory and formatting changes for 3390

All of these are general-use programming interfaces. General-use programming interfaces are provided to allow the customer to write programs that use the services of VM.

Monitor Record

There are values for device type and model number which are new for 3390 in the monitor record—MRIODATD. The following small table describes these values. For information on the offsets or lengths of the fields see the entire MRIODATD record following the small table.

		Values	
Name	**Description**	**Track Compatibility Mode**	**3390 Mode**
IODATD_RDEVTYPE	Device Type	X'04'	X'82'
IODATD_RDEVDVID	Device type number in packed decimal.	3380	3390
IODATD_CALMODLNO	Device Model Number	X'96' (Model 1) X'8A' (Model 2)	X'02' (Model 1) X'06' (Model 2)

General-Use Programming Interface

```
NAME - MRIODATD
DESCRIPTIVE NAME - Monitor Event Record
                   Domain 6 - I/O Domain
                   Record 5 - Attach Device
DESCRIPTION - Indicates that a device has been attached to the system.
```

OFFSETS	TYPE	LENGTH	NAME	DESCRIPTION
0 (0)	Structure	44	IODATD	
0 (0)	Character	20	IODATD_MRHDR	Record header.
0 (0)	Unsigned	2	MRHDRLEN	
2 (2)	Unsigned	2	MRHDRZER	
4 (4)	Unsigned	1	MRHDRDM	
5 (5)	Unsigned	1	*	
6 (6)	Unsigned	2	MRHDRRC	
8 (8)	Character	8	MRHDRTOD	
16 (10)	Character	4	*	
20 (14)	Character		MRHDR_END	
20 (14)	Bitstring	1	IODATD_RDEVTYPE	Device type.
21 (15)	Bitstring	1	IODATD_RDEVCLAS	Device class.
22 (16)	Bitstring	2	IODATD_RDEVDVID	Device type number in packed decimal. Applicable only if IODATD_RDEVDVIV = 1. Otherwise, the device type number is not available.
24 (18)	Bitstring	1	IODATD_CALMODLNO	Device model number. See IODATD_RDEVDVIV flag for its source.
25 (19)	Bitstring	1	IODATD_RDEVLPM	Logical path mask.
26 (1A)	Unsigned	2	IODATD_RDEVDEV	Device number.
28 (1C)	Unsigned	4	IODATD_RDEVSID	Host subchannel ID.
32 (20)	Character	8	IODATD_RDEVCHPS	Eight channel path IDs for this device.
40 (28)	Unsigned	2	IODATD_RDEVCUID	Control unit ID in packed decimal. Applicable only if IODATD_RDEVCUIV is on.
42 (2A)	Unsigned	1	IODATD_RDEVCUMN	Control unit model number. Applicable only if IODATD_RDEVCUIV is on.
43 (2B)	Bitstring	1	IODATD_CALFLGAS	Flag byte.
1...			IODATD_RDEVDVIV	Flag pertaining to the source of device number in IODATD_CALMODLNO. 0 = provided by user through the MODEL operand of the RDEVICE macro. 1 = provided by the device at device initialization time.
.1..			IODATD_RDEVCUID	When on, control unit information is supplied in IODATD_RDEVCUID and IODATD_RDEVDUMN.
.111 1111			*	Reserved for IBM use.
44 (2C)	Character		IODATD_END	

Cross Reference

NAME	OFFSET	VALUE	LEVEL
IODATD	0		1
IODATD_CALFLGAS	2B		2
IODATD_CALMODLNO	18		2
IODATD_END	2C		2
IODATD_MRHDR	0		2
IODATD_RDEVCHPS	20		2
IODATD_RDEVCLAS	15		2
IODATD_RDEVCUID	28		2
IODATD_RDEVCUIV	2B	40	3
IODATD_RDEVCUMN	2A		2
IODATD_RDEVDEV	1A		2
IODATD_RDEVDVID	16		2
IODATD_RDEVDVIV	2B	80	3
IODATD_RDEVLPM	19		2
IODATD_RDEVSID	1C		2
IODATD_RDEVTYPE	14		2
MRHDR_END	14		3
MRHDRDM	4		3
MRHDRLEN	0		3
MRHDRRC	6		3
MRHDRTOD	8		3
MRHDRZER	2		3

End of General-Use Programming Interface

DIAGNOSE Code X'24' – Device Type and Features

Privilege Class: Any

Use DIAGNOSE code X'24' to request identifying information and status information about a particular virtual device.

Note: DIAGNOSE code X'24' has been replaced by DIAGNOSE code X'210', which should be used for new development.

Entry Values:

Rx

Must contain:

- The virtual device number of the device for which information is requested

  OR

- The value negative 1 (-1). Specify -1 when the device is a virtual console whose device address is unknown to your virtual machine.

Ry

Not used.

Exit Values:

Rx

Contains information about a virtual console. Rx contains this data only if you specified -1 as an entry value and if CP located the console device.

Ry

Contains information about the specified virtual device.

Ry + 1

Contains information about the real device that is associated with the specified virtual device. However, if Ry has been specified as register 15, CP returns only virtual device information; no information is returned in register Ry + 1.

Usage Notes:

1. A CMS application may determine which set of ASCII and APL tables is currently in use with DIAGNOSE code X'24'. The RDEVTMCD field returned in Rx is set to the following values:

Value	Meaning
X'10'	ASCII mode (TERMINAL APL OFF)
X'14'	ASCII/APL, SI state (TERMINAL APL ON, using standard ASCII
X'18'	ASCII/APL, SO state (TERMINAL APL ON, using ASCII/APL)

2. For a DIAGNOSE code X'24' to a SNA device connected through VCNA, the model number (RDEVMDL) information is correct; however, the device type (RDEVTYPE) may not be reliable.

3. This DIAGNOSE code returns the real device class (RDEVTYPC) and type (RDEVTYPE) fields as *class special* (X'02') and *device unsupported* (X'01') for 3390 in either native or 3380 track compatibility mode. For 3390 device information you should use DIAGNOSE X'0210'.

   The following tables summarize the type of information that will be returned to the Rx, Ry, and Ry + 1 registers after the instruction has executed.

   Not all of the bits that are significant in VM/SP HPO remain significant in VM/XA SP.

   For those bytes affected by these differences, the names of the bits that do remain significant are supplied, following the meaning of the byte, in the tables below.

Rx Register:

Byte 0	Byte 1	Byte 2	Byte 3
RDEVTMCD	00	Virtual Device Address	

Symbolic Name	Meaning
RDEVTMCD	Terminal code bits defining the type of console and the translate table the console is using

Ry Register:

Byte 0	Byte 1	Byte 2	Byte 3
VDEVTYPC	VDEVTYPE	VDEVSTAT	VDEVFLAG

Symbolic Name	Meaning
VDEVTYPC	Virtual device type class.
VDEVTYPE	Virtual device type.
VDEVSTAT	Virtual device status. (Note that in VM/XA SP only the VDEVDED, VDEVBUSY, and VDEVNRDY bits may have a nonzero value in this byte.)

Bit Name	Value	Meaning
VDEVDED	X'01'	Dedicated Device
VDEVBUSY	X'20'	Device Busy
VDEVNRDY	X'04'	Device Not Ready

VDEVFLAG	Virtual device flags (Note that in VM/XA SP only the VDEVRDO, VDEVTDSK, VDEVENAB, DEVDIAL, VDEVCCW1, and VDEVRSRL bits may have a nonzero value in this byte.)

Bit Name	Value	Meaning
VDEVRDO	X'80'	Read Only
VDEVTDSK	X'40'	TDISK
VDEVENAB	X'80'	Line Enabled
VDEVDIAL	X'40'	Dialed Line
VDEVCCW1	X'10'	Console/Spooling Processing First CCW
VDEVRSRL	X'02'	Reserve Release

Ry + 1 Register:

Byte 0	Byte 1	Byte 2	Byte 3
RDEVTYPC	RDEVTYPE	RDEVDVMD	RDEVFTR or RDEVLLEN

Symbolic Name **Meaning**

RDEVTYPC Real device type class (For 3390 a class of special is returned X'02'.)

RDEVTYPE Real device type (For 3390 a type of unsupported is returned X'01'.)

RDEVDVMD Real device model number.

For a virtual console, if the screen size does not match a Model 2, 3, 4, or 5, then the model number will be set to 2. To get the true screen size, use DIAGNOSE code X'8C'.

For 3390 DASD the model number does not apply.

For a 3380 DASD, the high-order four bits of byte 2 represent the high-order four bits of the control unit model number, and the low-order four bits of byte 2 represent the low-order four bits of the device model number.

RDEVFTR Real device feature code for a device other than a virtual console.

(Note that in VM/XA SP the FTREXTSN bit is never set in this byte for a CLASDASD device.)

For 3390 DASD the feature code does not apply.

RDEVLLEN Current device line length for a local virtual console

Responses

The following chart lists the condition codes CP can return for DIAGNOSE code X'24', the meaning of each condition code, and the registers where data is returned.

Condition Code	Status	Rx Will Contain[1]	Ry Will Contain[2]	Ry + 1 Will Contain
0	Successful completion	Virtual console information	Virtual device information	Real device information
1	Undefined			
2	Virtual device exists but is not associated with a real device	Virtual console information	Virtual device information	
3	Invalid device address, OR the virtual device does not exist			

[1] The Rx register contains information only when DIAGNOSE code X'24' specifies a virtual console whose address is unknown.

[2] If Ry is register 15, CP returns only virtual device information; no information is returned in register Ry + 1.

End of General-Use Programming Interface

Migration Notes

VM/SP HPO

VM/XA SP provides only certain bits from VDEVSTAT and VDEVFLAG.

VM/XA SF

None.

General-Use Programming Interface

DIAGNOSE X'0210' – Device Type and Features

Privilege Class: G

Use DIAGNOSE X'0210' to request identifying information and status information about a particular virtual device. Your virtual machine must specify the address of the virtual device block for which information is requested. The device information is set up as follows:

Entry Values:

Rx is the general purpose register that contains the real address of the Virtual/Real Device Characteristics Block. The Virtual Device Number and the length fields of the Virtual/Real Device Characteristics Block are required on input.

Virtual/Real Device Characteristics Block:
The Virtual/Real Device Characteristics Block is the control block that DIAGNOSE X'0210' uses in returning device information. It must be on a fullword boundary or a program-check specification exception will be recognized.

The Virtual/Real Device Characteristics Block has the following format:

	Word				
INPUT	0	Virtual Device Number (Input Only)		Block Length (Bytes) (Input Only)	
OUTPUT	1	VDEVTYPC	VDEVTYPE	VDEVSTAT	VDEVFLAG
	2	RDEVTYPC	RDEVTYPE	RDEVMDL	RDEVFTR or RDEVLLEN
	3	00000000 00000000 00000000 00000000 (Reserved and must be zeroes)			
	4	Control Unit Type Number		CU Model Number	Dev Type (Byte 0)
	5	Dev Type (Byte 1)	Device Model	Device/Control Unit Features (Bytes 0,1)	
	6	Device/Control Unit Features (Bytes 2,3)		Dev Class Code	Dev Type Code
	7	Read Device Characteristics Device Dependent Information (Bytes 12 - 63) // //			
	19				

Figure 7. Fields in the Virtual/Real Device Characteristics Block

Virtual Device Number

Bytes 0 and 1 of word 0, must contain the virtual device number of the device for which information is requested. This field is a required input parameter, and must be supplied by the issuer of the DIAGNOSE.

Block Length

Bytes 2 and 3 of word 0, must be the length, in bytes, of the Virtual/Real Device Characteristics Block. This fields determines how much device information is returned to the requestor. If the value is less than 8 then a program-check/specification exception is recognized. This field is a required input parameter, and must be supplied by the issuer of the DIAGNOSE.

VDEVTYPC

Byte 0 of word 1, is the Virtual Device Type Class. This byte contains a binary code that indicates the device class (DASD, Tape, Unit Record...) of the device. (For 3390 a device class of DASD (X'04') is returned.)

VDEVTYPE

Byte 1 of word 1, is the Virtual Device Type. This byte contains a binary code that specifies the device type (3380, 3800, 3480...) of the device. (For 3390 a device class of DASD (X'82') is returned.)

VDEVSTAT

Byte 2 of word 1, is the Virtual Device Status. Only the VDEVDED, VDEVBUSY, and VDEVNRDY bits in this byte may have a nonzero value.

Bit Name	Value	Meaning
VDEVDED	X'01'	Dedicated Device
VDEVBUSY	X'20'	Device Busy
VDEVNRDY	X'04'	Device Not Ready

VDEVFLAG

Byte 3 of word 1, is the Virtual Device Flag. Only the VDEVRDO, VDEVTDSK, VDEVRSRL, VDEVENAB, VDEVDIAL, and VDEVCCW1 bits in this byte may have a nonzero value.

Bit Name	Value	Meaning
VDEVRDO	X'80'	Read Only
VDEVTDSK	X'40'	TDISK
VDEVENAB	X'80'	Line Enabled
VDEVDIAL	X'40'	Dialed Line
VDEVCCW1	X'10'	Console/Spooling Processing First CCW
VDEVRSRL	X'02'	Reserve Release

RDEVTYPC

Byte 0 of word 2, is the Real Device Type Class. This byte contains a binary code that indicates the device class (DASD, Tape, Unit Record...) of the device.

RDEVTYPE

Byte 1 of word 2, is the Real Device Type. This byte contains a binary code that specifies the device type (3380, 3800, 3480...) of the device.

RDEVMDL

Byte 2 of word 2, is the Real Device Model.

For 3390 this will be what is specified on the RDEVICE macro or returned on the sense ID.

RDEVFTR

Byte 3 of word 2, is the Real Device Feature for non-display/graphic devices.

For 3390 use the feature information returned from READ DEVICE CHARACTERISTICS.

RDEVLLEN

Byte 3 of word 2, is the Real Device Line Length field for display/graphic devices.

Reserved

Word 3, is reserved for future IBM use and must contain zeroes on input. If this word is non-zero then a program-check/specification exception is recognized.

Control Unit Identifier

Bytes 0 and 1 of word 4, are the virtual device control unit type as returned by SENSE ID (bytes 1, 2) or READ DEVICE CHARACTERISTICS (bytes 0, 1) channel commands.

Control Unit Model Number

Byte 2 of word 4, is the virtual device control unit model as returned by SENSE ID (byte 3) or READ DEVICE CHARACTERISTICS (byte 2) channel commands.

Device Type Number

Byte 3 of word 4 and 0 of word 5, are the virtual device type number as returned by SENSE ID (bytes 4, 5) or READ DEVICE CHARACTERISTICS (bytes 3, 4) channel commands.

Device Model Number

Byte 1 of word 5, is the virtual device model number as returned by SENSE ID (byte 6) or READ DEVICE CHARACTERISTICS (byte 5) channel commands.

Device and Control Unit Features

Bytes 2 and 3 of word 5 and bytes 0 and 1 of Word 6, is the virtual device feature codes as returned by READ DEVICE CHARACTERISTICS (bytes 6-9) channel command, as it applies to your virtual device.

Device Class Code

Byte 2 of word 6, is the virtual device class code as returned by READ DEVICE CHARACTERISTICS (byte 10) channel command.

Device Type Code

Byte 3 of word 6, is the virtual device type code as returned by READ DEVICE CHARACTERISTICS (byte 11) channel command.

Device Dependent Information

Words 7 through 19, contains the virtual device dependent information as returned the READ DEVICE CHARACTERISTICS (bytes 12-63) channel command, as it applies to your virtual device.

Usage Notes

1. The device information from VDEVTYPC, VDEVTYPE, VDEVSTAT, VDEVFLAG, RDEVTYPC, RDEVTYPE, RDEVFTR, and RDEVLLEN is returned in a format consistent with DIAGNOSE X'24'. The RDEVMDL field is the real device model, as opposed to the combined control unit model and real device model which is returned by DIAGNOSE X'24'.
2. Refer to the hardware manuals for content descriptions of the SENSE ID or the READ DEVICE CHARACTERISTICS channel command results.
3. This DIAGNOSE will return as much information to the virtual machine as possible in the Virtual/Real Device Characteristics Block. If the length is 8 bytes only the virtual device information will be returned.
4. The Virtual/Real Device Characteristics Block must be on a fullword boundary or a program-check/specification exception will be recognized.
5. If the Virtual/Real Device Characteristics Block length field is less than 8, then a program-check/specification exception is recognized.

Responses

Upon completion of DIAGNOSE X'0210', the condition code is:

Condition Code	Status
0	Normal completion
1	CP paging error, no data returned
2	The virtual device exists, but is not associated with a real device
3	Invalid device address, or the virtual device does not exist

End of General-Use Programming Interface

DIRECTORY Control Statement

The DIRECTORY control statement defines to CP the device on which you have allocated space for a directory of that system. (This device must also be a CP-owned volume.) If your system is part of a multi-system complex, a DIRECTORY control statement is required for each logical system. The DIRECTORY control statements must be the first statements in your source directory or control DIRMPART file (cluster format directory).

```
DIRectory │ vdevno devtype volser ┌ ┌ altvdevno ┐ ┐ [nnnnnnn-xxxx sysafnid]
          │ vdevno devtype volser │ { altvdevno } │
          │                       │ └ EDIT      ┘ │
          │                       └               ┘
```

where:

vdevno
Is the virtual device number of the device that is to contain the directory.

devtype
Is the device type. Valid device types are 2305, 3330, 3340, 3350, 3375, 3380, and 3390. If multiple DIRECTORY control statements are used, the device types do **not** have to be the same.

volser
Is the volume serial number of the directory volume. Its length is 1 to 6 alphanumeric characters.

altvdevno|EDIT
Identifies another device, *altvdevno*, on which to write the directory if the primary *vdevno* is unavailable. The EDIT value is supported for compatibility with VM/SP HPO. The EDIT value specifies that the DIRECTORY control statement defines a special work volume to be used by the VM/SP HPO DIRECT command when it is issued with the EDIT option. There can be only one of these statements in a directory and it must be the last of the set of DIRECTORY control statements. DIRECTXA validates the syntax of this statement but ignores its contents.

nnnnnn-xxxx
Is the CPUID of the system to which the DIRECTORY control statement applies. Use the QUERY CPUID command to get the values for *nnnnnn* and *xxxx*, where *nnnnnn* is the processor identification number, and *xxxx* is the model number of the real machine. For more information about the QUERY CPUID command, see *VM/XA SP CP Command Reference*.

If your system is an *n*-way processor operating in single-image mode, you can specify the CPUID as **nnnnn-xxxx*. This allows you to use one DIRECTORY control statement to define all CPUIDs. See "Usage Notes" on page 39 for more information about defining CPUIDs with one DIRECTORY statement.

sysafnid
Is a 1- to 8-character alphanumeric value that identifies the system whose object directory is affected by the SYSAFFIN control statements and its associated definition control statements.

Usage Notes

1. CP dynamically updates the active VM/XA SP directory if your virtual machine has the proper privilege class and:

   - The *volser* specified is the one on which the directory was found during initialization, or
   - No directory has yet been found, and the *volser* specified is currently owned by CP (listed in the SYSCPVOL macro).

2. When sharing a single source directory among several systems, all DIRECTORY statements must be listed at the beginning of the source directory and before any other control statements.

3. The use of the * in the high-order position of the CPUID allows the definition of a single DIRECTORY statement to be used to account for all of the CPU IDs of an *n*-way processor operating in single image mode. However, if for example a 4-way is partitioned into two 2-way processors, then it would be necessary to create a DIRECTORY statement for each CPU that would specify all 6 digits of the CPUID. This is because the CPUID of the two separate systems would only differ in the high-order byte of the CPUID. See the examples below.

Examples

The following example shows a way of coding the DIRECTORY control statement:

1. To specify that:

   - The 3330 volume labeled XA0001, at virtual address 0123, is to contain the new directory.
   - The volume should be used at virtual address 0223 if an error is encountered while trying to access the primary device.

   code the DIRECTORY statement as follows:

   ```
   directory 0123 3330 xa0001 0223
   ```

2. To specify that:

   - A multiple system definition where a 3090 2-way processor is partitioned into two uni-processors, SYSTEMA (serial 012345) and SYSTEMB (serial 112345)
   - An EDIT work volume which is a 3380 with a virtual device address of 0243
   - A volume serial number of DRM19F

   code the DIRECTORY statements as follows:

   ```
   directory 0123 3380 xa0001 0223 012345-3090 systema
   directory 0223 3380 xa0002 0233 112345-3090 systemb
   directory 0243 3380 drm19f edit
   ```

3. For a 3090 4-way configured as a single-image processor, you could code the following so that it would not matter on which CPU the system was IPLed:

   ```
   directory 0b64 3380 vmaipl 015211-3090 vma
   directory 0b64 3380 vmaipl 115211-3090 vma
   directory 0b64 3380 vmaipl 215211-3090 vma
   directory 0b64 3380 vmaipl 315211-3090 vma
   directory 0243 3380 drm19f edit
   ```

or this:

```
directory 0b64 3380 vmaipl *15211-3090 vma
directory 0243 3380 drml9f edit
```

4. To specify a 3084 partitioned into two 2-way images, you would have to code 2 pairs of DIRECTORY control statements, one pair for each image, where each statement in the pair differs only in the high-order byte of the CPU identification number.

```
directory 0b64 3380 vmaipl 020893-3084 vma
directory 0b64 3380 vmaipl 220893-3084 vma
directory 0784 3350 vmbipl 120893-3084 vmb
directory 0784 3350 vmbipl 320893-3084 vmb
directory 0243 3375 drml9f edit
```

5. In the case where an *n*-way processor – such as a 3084 – is to be treated as a single image within a complex, you would code the DIRECTORY statement with an * in the high-order byte of the CPU ID. This ensures that DIRECTXA would not be sensitive to the individual CPU on which is was run.

```
directory 0b64 3380 vmaipl *20893-3084 vma
directory 0784 3350 vmbipl 012345-3083 vmb
directory 0243 3380 drml9f edit
```

MDISK Control Statement (Device)

The MDISK control statement defines minidisks for virtual machines. You can use one DASD to provide several minidisks or you can use one DASD to provide one minidisk. If you code the MDISK statement, it must follow any general statements you code in a user's directory entry.

Note: Neither CP nor the directory program checks to see if you have defined minidisks that overlap. It is your installation's responsibility to assure that it does not define minidisks that overlap.

```
Mdisk  { vdevno } devtype { cylr { cyls } volser [mode [pr  [pw  [pm  ]]]] }
       { vaddr  }         {      { END  }        [     [ALL [ALL [ALL ]]]] }
                          { T-DISK cyls                                  }
```

where:

{ vdevno }
{ vaddr }
is the virtual device number or virtual address of the minidisk. "vdevno" applies to 370/XA virtual machines, and "vaddr" applies to System/370 virtual machines.

devtype
is the device type of the real device on which the minidisk resides. Valid device types are: 2305, 3330, 3340, 3350, 3375, 3380, and 3390

{ cylr }
{ T-DISK }
cylr is the starting cylinder number on the real DASD you specify on the volser operand.

T-DISK means that the minidisk is temporary, that is, it's created from a preselected pool of temporary disk space when the virtual machine logs on and returned to the pool when the virtual machine logs off or the virtual device is detached. T-DISKS cannot have passwords.

$\left\{\begin{array}{l}\textbf{cyls}\\ \textbf{END}\end{array}\right\}$

is the number of cylinders allocated to the minidisk. END specifies that the MDISK should be defined with the remaining available cylinders. Table 3 on page 42 shows the number of cylinders available in each format for each device type on which you can put minidisks.

volser

is the volume serial number of the real DASD volume on which the minidisk resides.

mode

is the access mode for the minidisk. The first letter in the access mode is the primary access mode (read-only, write, or multiple-write); the second letter (optional) is the alternate access (read-only or write). The access modes are:

R

specifies that read-only access is requested. Access is denied if any other user has the disk in write status.

RR

specifies that read-only access is requested, even if another user has the disk in write status.

W

specifies that write access is requested. The disk is not defined if any other user has the disk in read or write status. If you omit the entire access mode from the MDISK statement, W is the default.

WR

specifies that write access is requested if no other user has the disk in read or write status, but that an alternate access of read-only is acceptable if others do have access to the disk.

M

specifies that multiple access is requested. This means that a write access is to be given to the disk unless another user already has write access to it, in which case access is denied.

MR

specifies that a write access is to be given to the disk unless another user already has write access to it. In this case, a read access is given to the user.

MW

specifies that a write access is to be given to the disk in any case.

V

specifies that CP is to use its virtual reserve/release support in the I/O operations for the minidisk. Append V to the right of the access mode. For example, MWV means the minidisk functions with write linkage using CP's virtual reserve/release.

$\left[\begin{array}{l}\textbf{pr}\\ \textbf{ALL}\end{array}\right]$

is the password that allows sharing the minidisk in read mode. By using the CP LINK command, users share minidisks. The password is 1 to 8 characters. If you specify ALL, a user can link in read mode to this minidisk without using a password.

[pw
ALL]

is the password that allows sharing the minidisk in write mode. By using the CP LINK command, users share minidisks. The password is 1 to 8 characters. If you specify ALL, a user can link in write mode to this minidisk without using a password.

[pm
ALL]

is the password that allows sharing the minidisk in multiple-write mode. By using the CP LINK command, users share minidisks. The password is 1 to 8 characters. If you specify ALL, a user can link in multiple write mode to this minidisk without using a password.

Table 3. Maximum Minidisk Sizes

Device Type	**Models**	**CMS/VSAM (Cylinders)**	**CMS 800-byte Format (Cylinders)**	**CMS 512, 1K, 2K, or 4K Format (Cylinders)**
3330	1 or 2	494	246	404
3330	11	808	246	808
3333	1	404	246	404
3333	11	808	246	808
3340 (35 Mb)	A2, B1, B2	348	348	348
3340 (70 Mb)	B2, B2F	696	682	696
3350 (native mode)	A2, A2F, B2, B2F, C2, C2F	555	115	555
3375	A1, B1, D1	959	182	959
3380	A04, AA4, B04, AD4, BD4, AJ4, BJ4, CJ2	885	121	885
3380	AE4, BE4	1770	121	1770
3380	AK4, BK4	2655	121	2655
3390 (native mode)	A14, A18, B14, B18, B1C	1113	104	1113
3390 (3380 emulation mode)	A14, A18, B14, B18, B1C	885	121	885
3390 (native mode)	A24, A28, B24, B28, B2C	2225	104	2225
3390 (3380 emulation mode)	A24, A28, B24, B28, B2C	1770	121	1770

Usage Notes

1. You must format temporary disk space before you use it. If you want to format the temporary disk for CP use, issue the CPFORMAT command. See *VM/XA SP CP Command Reference* for information on the CPFORMAT command. If you want to format the temporary disk for CMS use, issue the FORMAT command. See *VM/XA SP CMS Command Reference.* To format minidisk space for other operating systems, use the commands of the operating system that controls the minidisk.
2. If you have written sensitive data on temporary disk space, reformat the disk before you detach it from your virtual machine. This action removes any risk of security exposure.
3. If for some reason two or more volumes have the same label, CP defines the minidisk on the volume that is attached to the system. (You cannot attach two volumes with the same label to the system at the same time.)
4. To define a full-pack minidisk you must specify that the minidisk is to contain all of a DASD's primary cylinders. You can optionally specify any of the DASD's alternate cylinders.
5. The 3370 is not supported as a minidisk, but only as a dedicated device.
6. 3390 used for CP volumes will be supported in either 3380 emulation or native mode of operation, on a volume basis. The device type on the MDISK statement for all minidisks on a given volume must agree with the device type of the volume specified in HCPRIO. Control of 3380 emulation capability is restricted to guests with Class F privileges. The use of ICKDSF is the recommended method to change emulation modes.
7. Read Special home address and Write Special home address are only support for: dedicated and full-pack minidisk for guests with Class F privileges.

Examples

1. To define a minidisk that:

   - Has virtual device number 191,
   - Takes up 5 cylinders beginning at cylinder 100 of the 3350 DASD with volume serial MDDASD,
   - Is accessible only in read-only mode, and
   - Cannot be linked to in any mode (no passwords are specified),

   code the following statement in the virtual machine's directory entry:

   ```
   mdisk 191 3350 100 5 mddasd rr
   ```

2. To define a minidisk that:

   - Has virtual device number 291,
   - Takes up 10 cylinders beginning at cylinder 105 of the 3350 DASD with volume serial MDDASD,
   - Is accessible only in read or write mode,
   - Can be linked to by anyone in read mode (but no other mode),

   code the following statement in the virtual machine's directory entry:

   ```
   mdisk 291 3350 105 10 mddasd mr all
   ```

3. To define a minidisk that:

   - Has virtual device number 198,
   - Takes up 5 cylinders beginning at cylinder 100 of the 3350 DASD with volume serial MDDASD,
   - Is accessible only in multiple-write (MW) mode,
   - Can be shared using CP's virtual reserve/release, and
   - Has a read password of 12WE45

   code the following statement in the virtual machine's directory entry:

   ```
   mdisk 198 3350 100 5 mddasd mwv 12we45
   ```

4. To define 5 cylinders of temporary 3350 DASD space at virtual device number 391, code the following statement in the virtual machine's directory entry:

   ```
   mdisk 391 3350 t-disk 5
   ```

5. To define a 3350 DASD full-pack minidisk at virtual device number 199, code the following statement in the virtual machine's directory entry:

   ```
   mdisk 199 3350 000 555
   ```

Minidisk Restrictions

The following restrictions exist for minidisks:

1. In the following cases, VM/XA SP modifies the cylinder data in user storage at the completion of the channel program:

   Read home address (with the skip bit off)
   Read record 0 (with the skip bit off)
   Sense (with the skip bit off)
   Read track (with the skip bit off)
   Read device characteristics

   This is necessary because the addresses must be converted for minidisks. Therefore, the data buffer area may not be dynamically modified during the I/O operation in these cases.

   Note: For the read record 0 case, the above restriction will not apply to devices that do not provide alternate track recovery.

2. On a minidisk, if a CCW string uses multitrack-search on I/O operations, subsequent operations to that disk must have preceding seeks or continue to use multitrack operations. There is no restriction for dedicated disks.

3. OS/PCP, MFT, and MVT ISAM or OS/VS ISAM running V=R or V=F may be used with a minidisk only if the minidisk is located at the beginning of the physical disk (that is, at cylinder 0). There is no restriction for DOS ISAM or OS/VS ISAM running virtual=virtual.

4. If the user's channel program for a count-key-data minidisk does not perform a seek operation, then, to prevent accidental accessing, VM/XA inserts a positioning seek operation into the user's channel program. Thus, certain channel programs may generate a condition code (CC) of 0 on an SIO instead of the expected CC of 1, which is reflected to a virtual machine. The final status is reflected to the virtual machine as an interruption. The final states for some channel programs initiated via SIOF or SSCH may not indicate deferred CC 1.

5. A DASD channel program directed to a 3330, 3350, 3375, 3380, or a 3390 device may give results on dedicated DASDs that differ from results on minidisks having nonzero relocation factors if the channel program includes multiple-track operations and depends on a Search-ID-High or Search-ID-Equal-or-High to terminate the program. The record 0 count fields on these devices must contain the real cylinder number of the track on which they reside. Therefore, a Search-ID-High, for example, based on a low virtual cylinder number, may terminate prematurely if a real record 0 is encountered.

   Minidisks with nonzero relocation factors on 3330, 3350, 3375, 3380 3390 devices are not usable under OS and OS/VS systems. This is because the locate catalog management function employs a Search-ID-Equal-or-High CCW to find the end of the VTOC.

6. The IBCDASDI program cannot assign alternate tracks for 3330, 3350, 3375, 3380, or 3390 devices.

   Notes:

   a. Device support facilities may assign alternate tracks

   b. Alternate track assignment is permitted for full-pack minidisks only.

7. If the DASD channel programs directed to 3330, 3350, 3375, 3380, or 3390 devices include a Write-Record-Zero CCW or a Write-Home-Address CCW, the results depend on whether the device is dedicated or not dedicated. For a dedicated 3330, 3350, 3375, 3380, or 3390 a Write-Record-Zero/Write-Home-Address CCW is allowed, but the user must be aware that the track descriptor record may not be valid from one 3330, 3350, 3375, 3380, or 3390 to another. For a 3330, 3350, 3375, 3380, or 3390 that is not dedicated, a Write-Record-Zero/Write-Home-Address CCW is accepted only if the device is a full-pack minidisk. If the device is NOT defined as a full-pack minidisk, issuing a Write-Record-Zero/Write-Home-Address CCW on a non-dedicated 3330, 3350, 3375, 3380, or 3390 CP rejects the command.

8. When performing DASD I/O, if the record field of a Search-ID-Argument is zero when a virtual SIO, SIOF, or SSCH is issued, but the Search-ID-Argument is dynamically read by the channel program before the Search-ID CCW is executed, the real Search-ID uses the relocated search argument instead of the argument that was read dynamically. To avoid this problem, the record field of a Search-ID argument should not be set to binary zero if the search argument is to be dynamically read or if the search ID on record 0 is not intended.

9. Diagnostic-Read-Home-Address and Diagnostic-Write-Home-Address commands are supported only for dedicated and full-pack 3375, 3380 and 3390 minidisks.

   Use of Diagnostic-Write-Home-Address is restricted to class F users.

10. Refer to *Device Support Facilities* for procedures to format all supported DASD for use in an OS/VS operating system running in a virtual machine.

11. If a V = R or V = F device is placed on the same control unit as the SYSRES device or a system dump device, the devices contend for outstanding sense information from the control unit. This could prevent proper system operation. To avoid this potential problem, IBM recommends that no V = R nor V = F dedicated devices nor V = R nor V = F full-pack minidisks share a control unit with the SYSRES device or with a system dump device.

12. 3390 used for CP volumes will be supported in either emulation or native mode of operation, on a volume basis. The device type on the MDISK statement for all minidisks on a given volume must agree with the device type of the volume specified in HCPRIO. Control of 3380 emulation capability is restricted to guests with Class F privileges. The use of ICKDCF is the recommended method to change emulation mode.

Reformatting

CP space on a 3390 device is formatted differently depending on what mode the device is in. Therefore, reformatting is necessary when changing the mode of the device. You will need to be certain that the Format Utility (CPFORMAT/CPFMTXA) is invoked after the mode of the 3390 device has been changed. Any CMS minidisks defined on the device will also need to be reformatted using CMS FORMAT.

Chapter 7. Changed System Product Interpreter Commands

The DIAG function and DIAGRC function have been updated to support DIAGNOSE X'0210'. These functions communicate through CP via a dummy DIAGNOSE instruction and are described in the discussion on the DIAGNOSE Instruction in the *VM/XA SP System Product Interpreter Reference*, SC23-0374.

The format is:

```
DIAG(210,devaddr)

DIAGRC(210,devaddr)
```

Where:

devaddr is the virtual device number of the device for which information is requested.

The value returned is a 13-byte string of virtual and real device information. The contents of the string are listed below.

Position	Contents
1 to 4	Virtual device information (word 1 from the VRDCB control block)
5 to 8	Real device information (word 2 from the VRDCB control block)
9 to 12	Device number
13	Condition code

The value returned by DIAGRC is identical to the DIAG function, except that the data is prefixed as shown below.

Position	Contents
1 to 9	Return code from CP
10	A blank
11	Condition code from CP
12 to 16	Five blanks

Chapter 8. Changed CP and CMS Commands

You can now specify *3390* on the following commands where the term *3380* could be used previously.

DEFINE (temporary disk)

Privilege Class: G

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

```
DEFine   ┌ T2305   ┐
         │ T2314   │
         │ T2319   │
         │ T3310   │
         │ T3330   │
         ┤ T3340   ├ [AS] vdev [CYL] nnnn
         │ T3350   │
         │ T3370   │
         │ T3375   │
         │ T3380   │
         │ T3390   │
         └ TFB-512 ┘
```

where:

T2305
T3330
T3340
T3350
T3375
T3380
T3390

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.

Responses

```
DASD vdev DEFINED
```

confirms the definition of the temporary disk. This response does 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.

MONITOR

Use the MONITOR command to control the selection, collection, and reporting of data from the system.

Complete Format for MONITOR

```
MONitor   EVent  [          ]  { I/O  {            { rdev           } }  }
                 [          ]  {      {   DEVice   { rdev rdev...   } }  }
                 [          ]  {      {            { rdev-rdev...   } }  }
                 [          ]  {      {   CLass    class-list         }  }
                 [          ]  {      {   TYpe     type-list          }  }
                 [          ]  {      {   VOLume   { volid-list }     }  }
                 [          ]  {      {            { CPVOL      }     }  }
                 [          ]  {      {   ALL                         }  }
                 [ DISable  ]  {                                         }
                 [          ]  { USER { USERID   userid-list }           }
                 [          ]  {      { ALL                  }           }
                 [          ]  {                                         }
                 [          ]  { PROCessor                               }
                 [          ]  {                                         }
                 [          ]  { SCHeduler { USERID userid-list }        }
                 [          ]  {           { ALL                }        }
                 [          ]  {                                         }
                 [          ]  { STORage                                 }
                 [ ENable   ]  {                                         }
                 [          ]  { SEEKS {            { rdev           } } }
                 [          ]  {       {   DEVice   { rdev rdev...   } } }
                 [          ]  {       {            { rdev-rdev...   } } }
                 [          ]  {       {   TYpe     type-list          } }
                 [          ]  {       {   VOLume   { volid-list }     } }
                 [          ]  {       {            { CPVOL      }     } }
                 [          ]  {       {   ALL                         } }
                 [          ]  {                                         }
                 [          ]  { ALL                                     }

                 {BLock m}

                 { STArt [BLock m] [PARTition n] }
                 { STOP                          }
```

```
MONitor   SAMPle  [         ]  { I/O  {          { rdev         } } }
                  [         ]  {      {   DEVice { 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                                 }

                  { INTerval n [ MINutes ] }
                  {            [ SEConds ] }

                  { RATE { n [ SEConds] } }
                  {      { STOP         } }

                  { STArt }
                  { STOP  }

          { STArt } [BLock m] [PARTition n]
          { STOP  }
```

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 through 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 are activated upon completion of the EVENT command, and new sample domains are activated when the next interval elapses.
3. For a detailed description of data collected, refer to the monitor records in *VM/XA SP CP Programming Services.*

4. The valid classes and types for use with SEEKS and I/O domains are listed in the following table:

CLASS	TYPE	Other TYPES Included	Valid For SEEKS	Valid For I/O
TAPE	3420			X
	3422			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
	3390		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

[1] GPRT represents all graphic display printers; GTERM represents all display terminals. For a list of graphic devices supported, see *VM/XA SP General Information.*

Table 4. Valid Classes and Types for SEEKS and I/O Domains

Migration Note

VM/SP HPO: VM/XA SP requires different commands, produces different data records, and uses different collection mechanisms.

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 ┐ ┌ IPL   ┐ ┌ CP  ┐
           └ rdev ┘ └ NOIPL ┘ │ ALL │
                              └ V=R ┘
```

where:

DUMP
indicates that you are designating the unit to receive CP ABEND dumps.

DASD
specifies that the system dump unit is a disk. The DASD dump space is released when the dump setting is changed. Spooling cylinders are 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
- 3390
- 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.
3. Your installation's system security can be compromised if you use the SET DUMP ALL command, because user pages are 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 is not 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 are set to CP and IPL.
7. When the system is initialized, the dump device is set to DASD when 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

```
{ type rdev } DUMP UNIT { CP  } { IPL   } [CYL nnn]
{ PRINTER   }           { ALL } { NOIPL }
                        { V=R }
```

where:

type rdev
is the device type and the real device number of your dump device. *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 automatically restarts 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.

Change in CMS Command Responses

The response to the QUERY DISK command contains soquel where 3390 DASD are accessed. If an I/O error is encountered during the operation of the RESERVE or FINIS commands, 32 sense bytes will be presented for all devices which produce 32 sense bytes.

Appendix A. New and Changed Modules, Macros, Control Blocks, and HELP Files

This appendix lists the modules, macros, control blocks, and help files that are created or changed by support for the IBM 3390 DASD device.

New and Changed Modules

An asterisk (*) indicates object modules.

3390 DASD

HCPADF
HCPAPS
HCPBLOK
HCPBSL
HCPCBI
HCPCCF
HCPCCU
HCPCCW *
HCPCDL *
HCPCPF
HCPCPN
HCPCSW
HCPCTB
HCPDCT
HCPDDI
HCPDDP *
HCPDDR
HCPDEF
HCPDEN
HCPDGB
HCPDGN
HCPDIR
HCPDMA
HCPDMP
HCPDMQ
HCPDMS
HCPDMW
HCPDSE
HCPDTB
HCPDTE
HCPDUC
HCPDUP *
HCPDVT
HCPERA
HCPERC
HCPERM
HCPERP
HCPFAA
HCPFAB
HCPFAC
HCPFAD
HCPFAE
HCPFAF
HCPFAG
HCPFAH
HCPFAI
HCPFAL
HCPFAM
HCPFAN
HCPFAR
HCPFAW
HCPFBD *
HCPFON
HCPFOR
HCPFOW
HCPGDS
HCPGEN
HCPGIO
HCPIDA
HCPIGR
HCPIIO
HCPIIS
HCPIMS
HCPINS *
HCPIOC
HCPIOF
HCPIOG
HCPIOS
HCPISH
HCPISL
HCPISP
HCPITP
HCPLDL
HCPLKS *
HCPLOD
HCPLSO
HCPMDP *
HCPMD1 *
HCPMD2 *
HCPMD4 *
HCPMES
HCPMNE
HCPMNJ
HCPMNT
HCPMNY
HCPMOT
HCPPEM
HCPPEN
HCPPGD
HCPPTB
HCPPTI
HCPPUC
HCPQSY
HCPRDA
HCPRDB
HCPRDI
HCPRIO
HCPRPA
HCPRRM
HCPRTR
HCPSAD
HCPSER
HCPSGH *
HCPSIM
HCPSYS
HCPTDD *
HCPTDK
HCPTMD *
HCPTPP *
HCPTRC
HCPTTB
HCPUDR
HCPUNS
HCPUNT *
HCPVCN
HCPVCT
HCPVCY
HCPVDB
HCPVER
HCPVMI
HCPVOS
HCPVRE
HCPVRJ
HCPXAB

New and Changed CMS Modules

DMSACM
DMSASN
DMSDEV
DMSDID
DMSDIO
DMSDIP
DMSFNS
DMSFOR
DMSGVE
DMSIND
DMSINI
DMSINS
DMSLDS
DMSQRS
DMSROS
DMSRSF
DMSRSV

New and Changed Macros

3390 DASD

BLDCPRAC
BLDCPRAE
BLDCPRLC
BLDCPRLE
BLDCPWAC
BLDCPWAE
BLDCPWLC
BLDCPWLE
HCPCCWAC
HCPCCWGS
HCPDEN$
HCPDMS$
HCPDMW$
HCPDUC$
HCPELORC
HCPLOREC
HCPMDLAT
HCPMDWRK
HCPREP$
HCPSDCMP
IDOFFSET
RDEVICE
SYSRES

New and Changed Control Blocks

3390 DASD

DEVTYPES
HCPALOC
HCPBCPRG
HCPCPTCA
HCPCWOEQ
HCPDVTYP
HCPFMABK
HCPFMLRC
HCPFMNUC
HCPFMREC
HCPIORBK
HCPMXRBK
HCPRCWBK
HCPSNSEQ
HCPVRDCB
MDRREC
OBRREC

New and Changed HELP Files

HCP0705E HELPMSG
HCP0711D HELPMSG
HCP0714D HELPMSG
HCP0719E HELPMSG
DEFINE HELPCP
DDR HELPCMS
DUMP HELPCPSE
HCP0705D HELPMSG
HCP0714E HELPMSG
HCP0717D HELPMSG
HCP0720E HELPMSG
MONITOR HELPCP
FORMAT HELPCMS

New and Changed $EXEC

HCPSADMP

Index

Virtual Machine/
Extended Architecture
System Product
Release 2.1

SYSTEM USABILITY COMMENTS

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?	□	□
• Is it easy to use and understand?	□	□
• Are the commands/messages easy to understand and use?	□	□
• Are the HELP facilities appropriate?	□	□

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 □
Installation □
Customization □
Diagnosis □
Other □

Planning □
Administration □
Operations □
End Use □

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 G35
P.O. Box 6, Route 17C Glendale Drive
Endicott, New York 13760

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 ______________________

IBM®

PRINTED IN U.S.A.

Program Number
5664-308

File Number
S370-34

GC23-0549-0

Printed in U.S.A.