

z/VM



# Migration Guide

*Version 3 Release 1.0*



z/VM



# Migration Guide

*Version 3 Release 1.0*

**Note:**

Before using this information and the product it supports, read the information under “Notices” on page 111.

**First Edition (February 2001)**

This edition applies to Version 3, Release 1, Modification 0 of IBM® z/VM (product number 5654-A17) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 1992, 2001. All rights reserved.**

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

# Contents

<b>Preface</b> . . . . .	<b>vii</b>
Who Should Read This Book . . . . .	vii
What You Should Know before Reading This Book . . . . .	vii
What This Book Contains . . . . .	vii
Related Information . . . . .	vii
How to Send Your Comments to IBM . . . . .	vii
<b>Chapter 1. Introduction</b> . . . . .	<b>1</b>
Migrating from Releases Prior to VM/ESA 2.1.0 . . . . .	1
Compatibility Terms Used in This Book . . . . .	2
<b>Chapter 2. System Changes</b> . . . . .	<b>3</b>
Installation and Service . . . . .	3
Electronic Delivery of Service [2.2.0] . . . . .	3
Local Modification of Replacement-Maintained Parts [2.2.0]. . . . .	3
S/390 <sup>®</sup> Service Update Facility [2.3.0] . . . . .	4
VMSES/E Enhancements [2.3.0] . . . . .	4
Tool for Removing PTFs [2.4.0] . . . . .	5
SPPF File Names [3.1.0]. . . . .	5
System Administration and Operation . . . . .	6
Year 2000 Support [2.2.0] . . . . .	6
CMS Productivity Aids Performance Enhancement [2.2.0]. . . . .	7
HELPIINST Saved Segment Replaced by HELPSEG and INSTSEG [2.2.0]. . . . .	7
Removable Media Services (RMS) Free Drive Support [2.2.0]. . . . .	7
SFS AUDIT Enhancement [2.2.0]. . . . .	7
Additional Year 2000 Support [2.3.0] . . . . .	7
ISO Date Used in Default CMS IPL Heading [2.3.0]. . . . .	8
Logical Device Limit Relief [2.3.0] . . . . .	8
Allocation of Real Storage for Segment Tables [2.3.0]. . . . .	8
TCP/IP Awareness [2.3.0] . . . . .	8
IPLing with the NODIRECT Option [2.3.0] . . . . .	10
New Processors Do Not Support 370-Mode Virtual Machines [2.3.0] . . . . .	10
Product Enablement Support [2.3.0] . . . . .	10
Dynamic CP Exits [2.4.0] . . . . .	10
CP Nucleus [3.1.0] . . . . .	11
CP Exit Modifications [3.1.0]. . . . .	11
High Level Assembler [3.1.0] . . . . .	11
RIO370 [3.1.0]. . . . .	11
64-Bit Support [3.1.0] . . . . .	11
Graphical User Interface (GUI) Facility Changes [3.1.0] . . . . .	12
VMLINK Improvements [3.1.0]. . . . .	12
Guest Support . . . . .	12
Multi-Path Lock Facility (MPLF) Support [2.2.0] . . . . .	13
Guest Coupling Simulation Support [2.3.0] . . . . .	13
Guest Support for QDIO Facility and OSA Express Gigabit Ethernet [2.4.0]. . . . .	14
Guest Cryptographic Support [2.4.0] . . . . .	14

Support for IEEE Floating Point [2.4.0] . . . . .	15
Parallel Access Volumes Support [3.1.0] . . . . .	15
Tape Support Enhancements [3.1.0] . . . . .	15
z/Architecture Support [3.1.0] . . . . .	16
Networking Support . . . . .	16
Java <sup>™</sup> and NetRexx <sup>™</sup> Support [2.3.0] . . . . .	16
MQSeries <sup>®</sup> Client for VM/ESA [2.3.0] . . . . .	16
Support for Hardware Architectures and Facilities . . . . .	16
S/390 Open Systems Adapter Support Facility (OSA/SF) [2.2.0]. . . . .	17
Integrated Cluster Bus Channel Support [2.4.0] . . . . .	17
Fibre Connection (FICON) Channel Support [2.4.0] . . . . .	17
Extended-TOD-Clock Support [2.4.0] . . . . .	17
Flash Copy Support [3.1.0] . . . . .	17
Application Development and Deployment. . . . .	17
REXX Sockets [2.2.0] . . . . .	18
Querying the CMS Level [2.2.0]. . . . .	18
Pseudo Timer Extended [2.2.0] . . . . .	18
Calling CSL Routines That Have Dates as Output [2.2.0] . . . . .	18
OS Simulation Support for Tape Library . . . . .	
Dataserfer [2.2.0] . . . . .	19
GCS Pathlength Reduced [2.2.0] . . . . .	19
Full MP Capability for VMCF [2.2.0] . . . . .	19
Language Environment <sup>®</sup> [2.3.0]. . . . .	19
Dynamic Link Libraries [2.3.0] . . . . .	19
Non-Relocatable Modules Loaded between 16-20MB [2.3.0] . . . . .	20
CMS Migration Utility Feature No Longer Available [2.3.0] . . . . .	20
Pipelines Code Bases Merged [2.3.0] . . . . .	20
OpenEdition <sup>®</sup> Enhancements [2.3.0] . . . . .	21
OS Simulation Enhancements [2.4.0] . . . . .	23
OpenEdition Enhancements [2.4.0]. . . . .	23
OpenEdition Renamed to OpenExtensions <sup>™</sup> [3.1.0] . . . . .	24
Open Files Limit Increased [3.1.0] . . . . .	24
BFS Root Not Case Sensitive [3.1.0] . . . . .	24
Binder/Loader Support [3.1.0] . . . . .	24
CMS OS Simulation Enhancement [3.1.0] . . . . .	24
Tape FICON and RAS Support [3.1.0]. . . . .	25
System Diagnosis . . . . .	25
FST and ADT Macro Changes to Support 4-digit Years [2.2.0] . . . . .	25
Viewing and Printing CP and CMS Control Blocks [2.3.0] . . . . .	26
Dump Viewing Facility No Longer Supports CP Dumps [3.1.0] . . . . .	26
VM Dump Tool [3.1.0]. . . . .	26
Other Changes . . . . .	26
Euro Support [2.4.0] . . . . .	27

<b>Chapter 3. Changes to External Interfaces</b> . . . . .	<b>29</b>
--	-----------

CP Changes . . . . .	29
System Configuration File Statements. . . . .	29
User Directory Control Statements. . . . .	30
CP Commands . . . . .	30
CP Utilities . . . . .	41
DIAGNOSE Codes . . . . .	41
CP Macros. . . . .	43
CP Messages . . . . .	43
CMS Changes . . . . .	44
General CMS Commands. . . . .	44
CMS File Pool Administration and Operator Commands . . . . .	49
OpenExtensions (OPENVM) Commands. . . . .	49
XEDIT Subcommands . . . . .	50
CMS Pipelines . . . . .	51
CMS Routines . . . . .	52
CMS Macros . . . . .	58
CMS Messages . . . . .	61
REXX/VM Changes . . . . .	61
GCS Changes. . . . .	62
TSAF Changes . . . . .	62
AVS Changes. . . . .	62
VMSES/E Changes. . . . .	62
\$PPF File Names . . . . .	62
VMSES/E Execs . . . . .	62
VMSES/E Messages . . . . .	64
Dump Viewing Facility Changes . . . . .	64

## Chapter 4. Migration Tasks . . . . . 65

Converting to System Configuration Files from HCPRIO and HCPSYS. . . . .	65
Difference with Multiple Systems Using Shared Data. . . . .	66
Converting HCPSYS Macros. . . . .	66
Converting HCPRIO Macros. . . . .	66
Migrating Spool Files and Saved Segments . . . . .	66
Using the Same Warm Start and Checkpoint Areas to Migrate All Spool Files at Once. . . . .	67
Using SPXTAPE to Migrate All or Some of Your Spool Files. . . . .	69
Building Saved Segments Individually . . . . .	69
Avoiding the Loss of Spool Files and System Data Files During Migration. . . . .	71
Sharing Data between Your Old System and Your New System . . . . .	72
Reserve/Release Considerations for VSE. . . . .	72
Examples of Reserve/Release . . . . .	72
Sharing Data among CMS Users on Multiple Systems . . . . .	77
Preparing for Cross-System Link . . . . .	77
Preparing for Cross-System Spool . . . . .	77
Migrating Your User Directory . . . . .	78
Object Directory Compatibility . . . . .	78
Getting Around Directory Differences . . . . .	78
User Directory Control Statement Changes . . . . .	78
Default Volume Label Changes . . . . .	78
Preparing to Migrate Your User Directory . . . . .	79
Using Two Source Directories . . . . .	79
Considerations for Using a Directory Maintenance Program . . . . .	79
Steps for Migrating Your Source User Directory . . . . .	80

What If I Cannot IPL Because of a Problem with the User Directory?. . . . .	81
Migrating Your SFS File Pool Servers. . . . .	81
Using Two System Images . . . . .	81
Using the 'Cut and Go' Strategy . . . . .	84
Directory Entry Considerations and Changes . . . . .	85
Converting an SFS File Pool Server Back to Your Old Release . . . . .	85
Establishing Connectivity between Your New and Old Systems . . . . .	86
Procedures . . . . .	87
How to Back Out . . . . .	88
Backing Out Because of System Problems . . . . .	88
Backing Out Individual Users Because of Problems with Application Programs. . . . .	89
Installing a Backlevel CMS . . . . .	89
Before You Install a Backlevel CMS . . . . .	89
Steps for Installing a Backlevel CMS . . . . .	91
Defining Various Levels of CMS . . . . .	95
Swapping Between the Backlevel CMS and z/VM 3.1.0 CMS . . . . .	97
Migrating REXX and EXEC 2 Application Programs . . . . .	98

<b>Appendix A. CMS Pipelines Message Cross-Reference [2.3.0] . . . . .</b>	<b>99</b>
DMS to FPL Message Cross-Reference . . . . .	99
FPL to DMS Message Cross-Reference . . . . .	101

<b>Appendix B. Sample Utilities for CP Configurability . . . . .</b>	<b>105</b>
HCPTRIO . . . . .	106
HCPTSYS . . . . .	108

<b>Notices . . . . .</b>	<b>111</b>
Trademarks . . . . .	113

<b>Glossary . . . . .</b>	<b>115</b>
---------------------------	------------

<b>Bibliography . . . . .</b>	<b>117</b>
z/VM Base Publications . . . . .	117
Evaluation . . . . .	117
Installation and Service . . . . .	117
Planning and Administration . . . . .	117
Customization . . . . .	117
Operation . . . . .	117
Application Programming . . . . .	117
End Use . . . . .	118
Diagnosis. . . . .	118
Publications for Additional Facilities. . . . .	118
DFSMS/VM® . . . . .	118
OSA/SF . . . . .	118
Language Environment® . . . . .	118
Publications for Optional Features . . . . .	118
CMS Utilities Feature. . . . .	118
TCP/IP Feature for z/VM . . . . .	118
OpenEdition® DCE Feature for VM/ESA® . . . . .	118
LANRES/VM . . . . .	119
CD-ROM . . . . .	119

**Index . . . . . 121**





---

## Preface

This book provides information to help you migrate from VM/ESA<sup>®</sup> Version 2 (any release) to z/VM Version 3 Release 1.0 (z/VM 3.1.0).

For information about migrating from earlier VM releases, see “Migrating from Releases Prior to VM/ESA 2.1.0” on page 1.

For information about migrating your TCP/IP feature to the latest level, see *z/VM: TCP/IP Level 3A0 Planning and Customization*.

---

## Who Should Read This Book

This book is intended for system programmers, system analysts, and system support personnel who are responsible for planning and completing a system migration. Parts of this book may also be helpful to application programmers.

---

## What You Should Know before Reading This Book

This book assumes that you are familiar with VM and that you have a general understanding of z/VM 3.1.0 from reading the *z/VM: General Information* book.

---

## What This Book Contains

This book includes the following information:

- Introduction and definition of terms
- VM system changes and enhancements that you should be aware of before migrating
- VM external interfaces that have changed since VM/ESA 2.1.0
- Migration tasks that you might need to perform

This book is based, in part, on material from the *VM/ESA V2R4: Conversion Guide and Notebook*.

---

## Related Information

For more information about z/VM functions, see the books listed in the “Bibliography” on page 117.

---

## How to Send Your Comments to IBM

Your feedback is important in helping us to provide the most accurate and high-quality information. If you have comments about this book or any other VM documentation, send your comments to us using one of the following methods. Be sure to include the name of the book, the form number (including the suffix), and the page, section title, or topic you are commenting on.

- Visit the z/VM web site at:  
<http://www.ibm.com/servers/eserver/zseries/zvm>

There you will find the feedback page where you can enter and submit your comments.

- Send your comments by electronic mail to one of the following addresses:
  - Internet:**       pubrcf@vnet.ibm.com
  - IBMLink™:**     GDLVME(PUBRCF)
- Fill out the Readers' Comments form at the back of this book and return it using one of the following methods:
  - Mail it to the address printed on the form (no postage required in the USA).
  - Fax it to 1-607-752-2327.
  - Give it to an IBM representative.

---

## Chapter 1. Introduction

Migration is the transfer and adjustment of information required to upgrade from one VM release to a later VM release. Rather than starting from scratch when you upgrade from your old system, you probably transfer and adjust a lot of your old information to use on the new system. Information you may transfer includes:

- I/O configurations
- Memory configuration (saved segment definitions)
- Spool files
- User directory entries
- SFS file pool servers
- Application programs

**Note:** This book assumes that you have already determined your hardware requirements for the new system, including your DASD, real storage, and performance needs.

Before you begin to transfer any information, knowing the differences that you will encounter in the new system, compared to your old system, may make the migration easier. This book provides an overview of the changes to VM since VM/ESA 2.1.0. The information is presented two ways:

- “Chapter 2. System Changes” on page 3 provides a brief discussion of significant changes and enhancements to system functions. The chapter is organized by topic area according to where the change or enhancement will likely have the greatest effect, such as installation and service, guest support, application development, and so on.
- “Chapter 3. Changes to External Interfaces” on page 29 identifies changes to specific *existing* external interfaces (commands, routines, macros, and so on). Each change is identified as either upwardly compatible or incompatible. (For definitions of these terms, see “Compatibility Terms Used in This Book” on page 2.) The chapter is organized by component and by interface type.

In both chapters, the release in which each change occurred is indicated in brackets (like this, [2.3.0]), so you can select the items that affect you. Also included are references to other books where the changes are discussed in more detail.

The final chapter in the book provides guidance for some migration tasks you might have to do, such as migrating your spool files or migrating your user directory.

---

### Migrating from Releases Prior to VM/ESA 2.1.0

If you are migrating from VM/ESA 1.1.5 370 Feature, VM/ESA 1.2.0, VM/ESA 1.2.1, or VM/ESA 1.2.2, you need to obtain the following book (not included in the z/VM library):

- *VM/ESA V2R4: Conversion Guide and Notebook*, GC24-5839

Use that book to determine the differences between your old release and VM/ESA 2.4.0. Then use this book to determine the additional differences between VM/ESA 2.4.0 and the new release.

## Introduction

If you are migrating from an even older VM release, you need to obtain one of the following books (not included in the z/VM library):

- *VM/ESA V2R1: Conversion Guide and Notebook for VM/SP, VM/SP HPO, and VM/ESA (370 Feature)*, SC24-5754
- *VM/ESA V2R1: Conversion Guide and Notebook for VM/XA SP and VM/ESA*, SC24-5753

Use that book to determine the differences between your old release and VM/ESA 2.1.0. Then use this book to determine the additional differences between VM/ESA 2.1.0 and the new release.

---

## Compatibility Terms Used in This Book

This book uses certain terms to convey the degree of compatibility for the changes to external interfaces introduced by each new VM release. External interfaces are commands, routines, utilities, DIAGNOSE codes, macros, user directory control statements, configuration file statements, system services, and so on. The compatibility terms are:

### **Upwardly compatible**

The syntax, functions, or responses of the external interface have been changed, but not significantly. Invocations and applications using the external interface on the new system in the same manner as on the old system should continue to execute unchanged.

**Note:** New function, if not exploited, is also upwardly compatible.

### **Incompatible**

The syntax, functions, or responses of the external interface have been changed significantly. Some invocations and applications using the external interface on the new system in the same manner as on the old system may execute incorrectly or may not execute at all.

**Note:** Depending on how you use the interface, an incompatible change may actually be upwardly compatible for you.

---

## Chapter 2. System Changes

This chapter provides information about significant VM system changes (changed or new function or support) since VM/ESA 2.1.0. The release in which each change occurred is indicated. These changes may affect your migration to the new system, or they may affect how you administer or operate the new system, or how you run guests or applications.

For information about changes to external interfaces (commands, routines, and so on), see “Chapter 3. Changes to External Interfaces” on page 29.

This chapter contains the following major sections:

- “Installation and Service”
- “System Administration and Operation” on page 6
- “Guest Support” on page 12
- “Networking Support” on page 16
- “Support for Hardware Architectures and Facilities” on page 16
- “Application Development and Deployment” on page 17
- “System Diagnosis” on page 25
- “Other Changes” on page 26

---

### Installation and Service

The following changes may affect how you maintain the new system (compared to your old system) and how you service products that run on the new system.

**Note:** For information about changes to the tools and procedures for installing the VM base, see the *z/VM: Installation Guide*.

#### Electronic Delivery of Service [2.2.0]

The ITNVTSTR EXEC supports the delivery of IBM service electronically by satellite. This electronic delivery is provided by the IBM Advanced Digital Delivery System product (5799-XDG).

For more information, see:

- *z/VM: VMSES/E Introduction and Reference*
- *Advanced Digital Delivery User's Guide, SC23-3281*

#### Local Modification of Replacement-Maintained Parts [2.2.0]

The VMFREPL EXEC supports the local modification of replacement-maintained parts. VMFREPL can be used to:

- Copy the highest level of a part
- Copy a specified part
- Update a Version Vector Table
- Update a Select Data file
- Display the highest level of a part

For more information, see the *z/VM: VMSES/E Introduction and Reference*.

## System Changes — Installation and Service

### S/390<sup>®</sup> Service Update Facility [2.3.0]

The S/390 Service Update Facility (SUF) is an internet-based S/390 software service tool that makes ordering and receiving OS/390<sup>®</sup>, VM, and VSE software service quick and easy. SUF allows systems programmers to order both corrective and preventive service through the internet. VM service, both corrective and preventive, can be delivered through the internet or through standard physical media. Where available, Advanced Digital Delivery (satellite delivery) is also an option.

A common GUI interface is provided by the SUF Customer Application Server. This workstation server can support multiple systems programmers supporting multiple S/390 servers. OS/390, VM, and VSE servers can be attached to a single workstation server simultaneously.

For details regarding prerequisites, entitlement, and how to obtain SUF, refer to the following URL:

<http://www.s390.ibm.com/suf>

### VMSES/E Enhancements [2.3.0]

The following execs have been added:

#### **VMFENRPT**

Creates a report of the products that are enabled, disabled, and installed on your system.

#### **VMFSUFIN**

Installs service from RSU service envelope files, COR service envelope files, or both.

#### **VMFSUFTB**

Builds a table, sysid SYSSUF, that contains a list of all installed products and related data needed by the Service Update Facility to service each product.

#### **VMFINS DISABLE**

Changes a product to a disabled state.

#### **VMFINS ENABLE**

Changes a product to an enabled state.

The following Software Inventory tables have been added:

#### **VM SYSSUF (System-Level Service Update Facility table)**

Contains a list of all products that are installed on the system. For each product, it contains the data needed by the Service Update Facility to service the product.

#### **VM SYSREST (System-level Restart table)**

Contains records used to restart the VMFSUFIN EXEC, which is part of the Service Update Facility.

The following execs have been changed:

#### **ITNVTSTR**

The PROD and KEY operands have been added. The PROD operand identifies the products that were ordered for a RSU package. The KEY operand indicates that the RSU package is one of two packages that are to be installed together.

### VMFINS DELETE

The DISABLE operand has been added. The DISABLE operand sets up a product as disabled and deleted.

### VMFINS MIGRATE

The DISABLE and ENABLE operands have been added. The DISABLE operand sets up a product as disabled. The ENABLE operand sets up a product as enabled.

### VMFINS INSTALL

The DISABLE, ENABLE, NOSETUP, and SETUP operands have been added. The DISABLE operand sets up a product as disabled. The ENABLE operand sets up a product as enabled. The NOSETUP operand indicates that a new minidisk or directory access order is not set up. The SETUP operand indicates that a new minidisk or directory access order is set up. It is set up according to entries in the :MDA section of the product parameter file.

The following Software Inventory table has been changed:

### VM SYSAPPS

The :ESTAT tags has been added. The :ESTAT tag specifies the enablement status of a product on the system.

For more information about these changes, see the *z/VM: VMSES/E Introduction and Reference*.

## Tool for Removing PTFs [2.4.0]

The VMFREM EXEC removes individual PTFs by “un-applying” them from all service levels (apply disks) and optionally “un-receiving” them. To “un-apply” a PTF means to undo the function previously performed for that PTF by the VMFAPPLY command. To “un-receive” a PTF means to undo the function previously performed for that PTF by the VMFREC command.

VMFREM also removes complete service levels and optionally “un-receives” PTFs that are applied only to the removed levels. In addition, commit support is provided for individual PTFs that have been applied. For more information, see the *z/VM: VMSES/E Introduction and Reference*.

## \$PPF File Names [3.1.0]

The file name of the base \$PPF file for each component has changed:

Component	File name and file type
CP	3VMVMB10 \$PPF
CMS	3VMVMA10 \$PPF
REXX	3VMVMF10 \$PPF
GCS	3VMVML10 \$PPF
TSAF	3VMVMH10 \$PPF
AVS	3VMVMD10 \$PPF
VMSES/E	3VMVMK10 \$PPF
Dump Viewing Facility	3VMVMI10 \$PPF

The IBM-supplied override PPF, which contains overrides to the base \$PPF files for each component, has changed to ZVM \$PPF. The override PPF for VM/ESA was ESA \$PPF.

### System Administration and Operation

The following changes may affect how you administer or operate the new system (compared to your old system).

#### Year 2000 Support [2.2.0]

VM/ESA 2.2.0 provides support for the year 2000 and beyond. CP and CMS commands that accept dates as input or display dates as output have been enhanced to support dates with 4-digit years. In addition to commands, CMS Pipelines and application program interfaces such as REXX and CSL routines have been enhanced to support dates with 4-digit years. Existing dates consisting of 2-digit years are resolved with correct 4-digit year information. A new CSL routine, `DateTimeSubtract`, has been added to assist users in modifying their user-written programs to work with 4-digit date support. The TOD clock support is also enhanced to aid in IPLing a VM system in the year 2000 or later.

The capability to set a default date format for CP and CMS commands is provided on a system-wide basis and also on a user (virtual machine) basis. A new system configuration file statement, `SYSTEM_DATEFORMAT`, sets the system-wide default date format. A new directory control statement, `DATEFORMAT`, sets the default date format for a user (virtual machine). Two new CP commands, `SET DATEFORMAT` and `QUERY DATEFORMAT`, set and query the default date formats for the system and individual users. Default date formats for the `FILELIST` and `RDRLIST` commands can also be set with the `CMS DEFAULTS` command.

#### Range of Dates Supported for Files

Minidisk files and SFS files support dates only in the range of 01/01/1900 to 12/31/2099.

#### Files on a Back Level File Pool Server

4-digit years are not supported on a file pool server from a VM release prior to VM/ESA 2.2.0. The new system will interpret all dates associated with a file on a back level server as 19yy, where yy is the 2-digit year.

#### Remote Systems That Are Not Year 2000 Ready

If a remote system that is not Year 2000 ready accesses minidisks on the new system, all the 4-digit years for existing files will appear to be 19yy, even if they have been created after 1999.

If you copy a file from a remote system to a minidisk or SFS directory on the new system using the `OLDDATE` option, the 4-digit year for the file may not be correct if the remote system is not Year 2000 ready. Therefore, an incorrect date might be propagated onto your system.

If you use the new system to create a file on a minidisk on a remote system that is not Year 2000 ready, the new system will see the correct 4-digit year.

#### Profiles for CMS Productivity Aids

If you have a pre-VM/ESA 2.2.0 XEDIT profile for `FILELIST`, `RDRLIST`, `SENDFILE`, or `PEEK` on your A-disk or in your search order accessed ahead of the system disk, some PF key functions may not work correctly with the new `FULLDATE` and `ISODATE` options. To ensure that the new date functions operate correctly, you should erase your old profiles and create new ones. The recommended method for customizing the operation of the PF keys for these commands (as well as other productivity aids that use profiles) is to build a user profile that first calls the



## System Changes — Administration and Operation

profile from the system disk, followed by your customized changes. For more information, see Appendix A of the *z/VM: CMS Command Reference*.

### CMS Productivity Aids Performance Enhancement [2.2.0]

The CMS productivity aids FILELIST, RDRLIST, and PEEK have been rewritten from EXEC 2 to REXX and compiled. They have also been placed into the CMSINST logical saved segment, which has been moved above the 16MB line. (See “HELPINST Saved Segment Replaced by HELPSEG and INSTSEG [2.2.0]”.) This will improve the performance of these commands and also aid in the servicing of these parts.

### HELPINST Saved Segment Replaced by HELPSEG and INSTSEG [2.2.0]

The default CMS installation saved segment (CMSINST) and the HELP logical saved segment are no longer defined in a single physical saved segment called HELPINST. CMSINST is now defined in the INSTSEG physical saved segment, which has been moved above the 16MB line. HELP is now defined in the HELPSEG physical saved segment, which is still located below 16MB. This change provides more room for help files in the HELP saved segment.

More of the CMS productivity aids (execs and XEDIT macros that reside on the S-disk) have been added to CMSINST. For a list of the contents of CMSINST, see the CMSINST LSEG file.

To allow CMSINST to be moved above 16MB, execs included in CMSINST that were formerly written in EXEC or EXEC 2 have been rewritten in REXX. If you have user modifications to any of these rewritten execs, you must rewrite your modifications in REXX. You cannot add EXEC or EXEC 2 execs to CMSINST. If you have EXEC or EXEC 2 execs you want to provide in a logical saved segment, you must add them to a segment that is loaded below 16MB.

### Removable Media Services (RMS) Free Drive Support [2.2.0]

DIAGNOSE code X'254', Access Real Subsystem, allows the RMS virtual machine to issue nondrive dependent I/O to a 3494 or 3495 tape library subsystem, without requiring the DFSMS/VM<sup>®</sup> RMS machine to have a CP-attached tape drive. For more information, see *z/VM: CP Programming Services*.

### SFS AUDIT Enhancement [2.2.0]

You can now specify `AUDIT OFF CLOSE fn ft` or `AUDIT CLOSE fn ft` to create a file to capture a “snapshot” of the security audit trace file. This prevents overwriting the security audit trace file and allows you to get audit information while auditing remains on. For more information, see *z/VM: CMS File Pool Planning, Administration, and Operation*.

### Additional Year 2000 Support [2.3.0]

VM/ESA 2.3.0 provides extended support for year 2000 and beyond. A new CMS pipelines stage called DATECONVERT allows date conversion, validation, and windowing functions. It supports all of the REXX date formats, plus additional formats. It is a powerful front-end to the DateTimeSubtract CSL routine. There are two new data migration and testing aid commands: FIXCENT and FILESERV FIXCENT. These commands can be used to set the internal century information stored with minidisk and SFS files. Note that this is not needed as part of the general processing. There is a new conversion command that converts dates in the

## System Changes — Administration and Operation

user's NETLOG file called NETLCNVT as well as message display enhancements to the NETDATA command. When IPLing VM, the 'SET DATE' prompt has been enhanced to accept 4-digit year dates.

For more information, see:

- *z/VM: CMS Pipelines Reference*
- *z/VM: CMS Command Reference*
- *z/VM: CMS File Pool Planning, Administration, and Operation*

### ISO Date Used in Default CMS IPL Heading [2.3.0]

The date field in the default CMS IPL heading has been changed to ISO format (*yyyy-mm-dd*). Each time you IPL CMS, a heading is displayed to identify the VM release level and time stamp of the CMS system being IPLed. A default heading is constructed when the CMS system is generated if the VERSION= parameter in the DEFNUC macro is specified without a value (which is the default).

### Logical Device Limit Relief [2.3.0]

Support has been added to CP to let you to change the maximum number of logical devices allowed on the system. The old limit of 4096 devices is now the default, but you can use the new CP SET MAXLDEV command to reset the limit up to 32768 devices. The new QUERY MAXLDEV command allows you to determine the current setting.

### Allocation of Real Storage for Segment Tables [2.3.0]

To represent the virtual storage for each primary address space (virtual machine) larger than 32MB and each nonprimary address space (data space), CP creates segment tables in real storage:

- For each virtual machine larger than 32MB but less than or equal to 1024MB, one real storage frame is allocated for the segment table.
- For each data space less than or equal to 1024MB, one real storage frame is allocated for the segment table.
- For each virtual machine or data space larger than 1024MB, two consecutive real storage frames are allocated for the segment table.

CP creates the segment table at the start of the storage frame. But in previous releases, the entire storage frame was allocated for the segment table regardless of the actual size of the table. This could have resulted in constraints on real storage availability if large numbers of virtual machines had arbitrarily large storage sizes, or were allowed to create many data spaces and/or very large data spaces, or loaded saved segments or saved systems defined at very high addresses outside the virtual machines.

On the new system, although real storage frames are still nominally allocated for the segment tables as indicated above, only the actual amount of real storage necessary to contain the segment tables is used. Any storage remaining beyond the end of a segment table may be used for CP free storage.

### TCP/IP Awareness [2.3.0]

Support has been added to the following CP functions to detect TCP/IP addresses and include those addresses in queries and responses:

- QUERY LDEVS command
- QUERY VIRTUAL CONSOLE command

## System Changes — Administration and Operation

- DIAGNOSE code X'7C' (Logical Device Support Facility)
- CP accounting records
- Access control interface (ACI)

For more information, see:

- *z/VM: Planning and Administration*
- *z/VM: CP Command and Utility Reference*
- *z/VM: CP Programming Services*

### Changes to Accounting Records for TCP/IP

The formats of the CP accounting records for journaling (Record types 04, 05, 06, and 08) have been changed to supply TCP/IP data, as indicated in Table 1.

Table 1. Changes to Accounting Records for TCP/IP Awareness

Record Type	Release	Columns	Contents
04	Old	57-70	Reserved
		71-78	LUNAME for SNA terminal
	New	57-62	Reserved
		63-70	Network qualifier for SNA terminal or host virtual machine name for TCP/IP terminal
05	Old	58-70	Reserved
		71-78	LUNAME for SNA terminal
	New	58-62	Reserved
		63-70	Network qualifier for SNA terminal or host virtual machine name for TCP/IP terminal
06	Old	61-70	Reserved
		71-78	LUNAME for SNA terminal
	New	61-62	Reserved
		63-70	Network qualifier for SNA terminal or host virtual machine name for TCP/IP terminal
08	Old	29-64	Reserved
		29-48	Reserved
	New	49-56	LUNAME for SNA terminal or IP address for TCP/IP terminal
		57-64	Network qualifier for SNA terminal or host virtual machine name for TCP/IP terminal

### Changes to the ACI for TCP/IP

In the access control interface (ACI) to an external security manager (ESM), the ACIPARMS control block has been changed. In the ACIPARMS parameter list for authorization checking on the LOGON command, a new ACILOGIP option can be set under ACILGOPT (if the ACILOGCL option is also set) to indicate a logical terminal with an IP address. The IP address is specified in ACITRMID.

## System Changes — Administration and Operation

### IPLing with the NODIRECT Option [2.3.0]

When you IPL your new system with the NODIRECT option, the OPERATOR user ID is logged on in ESA mode with 2047MB of virtual storage. In releases prior to VM/ESA 2.3.0, OPERATOR was logged on in 370 mode with 32MB of virtual storage.

### New Processors Do Not Support 370-Mode Virtual Machines [2.3.0]

New processors (such as the IBM S/390 Parallel Enterprise Server™ G4 and later) do not support 370-mode virtual machines. If you install your new system on such a processor, and a user attempts to define a 370-mode virtual machine, the following will occur:

- If a user issues the SET MACHINE 370 command, the command is rejected and message HCP1016E is issued.
- If a user with a MACHINE 370 statement in their directory logs on, the logon continues, but the virtual machine is defined as XA-mode. In addition, CP's 370 accommodation support (370ACCOM) is set ON and message HCP1512E is issued. Note that doing certain operations, such as resetting the system or defining storage, will set 370ACCOM OFF.

### Product Enablement Support [2.3.0]

You can use the CP SET PRODUCT command and the PRODUCT configuration statement to define products and features to your VM system and to determine whether the product or feature can run on that system. You can then use the CP QUERY PRODUCT command to display information about the products that are defined to the system. You can also use DIAGNOSE code X'27C' to request information about the enablement status of a single product or feature.

For more information, see:

- *z/VM: Planning and Administration*
- *z/VM: CP Command and Utility Reference*
- *z/VM: CP Programming Services*

### Dynamic CP Exits [2.4.0]

You can define CP exit points dynamically using the DEFINE EXIT command or system configuration file statement. A dynamic CP exit point behaves just like a formally-defined exit point, except that its ability to influence subsequent processing in the module containing the exit point is limited, and it does not carry over an IPL. Dynamic exits provide a convenient way to collect diagnostic or other information or to handle many situations in which the flow of control of a CP module does not need to be changed extensively.

The MODIFY EXIT command or system configuration file statement allows you to change the definition of an existing dynamic CP exit point or remove it from the system. The QUERY EXITS command has been enhanced to display additional information about a dynamic exit: exit location, characteristics, and parameter definitions.

For more information, see:

- *z/VM: CP Exit Customization*
- *z/VM: CP Command and Utility Reference*
- *z/VM: Planning and Administration*

### CP Nucleus [3.1.0]

An IPLable CP nucleus is no longer supported. Only a nucleus load module may be used to hold CP nucleus code.

z/VM now provides two CP images:

- 32-bit

You can IPL this CP image on an ESA/390 or z/Architecture processor.

- 64-bit

You can IPL this CP image only on an z/Architecture processor.

A dual-image CP is also provided with z/VM. If the dual-image CP is installed, it automatically determines at IPL if the processor is z/Architecture-capable. If it is, the 64-bit CP image is loaded; otherwise, the 32-bit CP image is loaded. This can be overridden on the IPL to load the 32-bit CP on a z/Architecture processor.

### CP Exit Modifications [3.1.0]

CP exits may need to be modified for correct operation. For more information, see *z/VM: CP Exit Customization*.

### High Level Assembler [3.1.0]

The IBM High Level Assembler V1R4.0 is required for:

- Installation:
  - Adding devices that cannot be sensed (updating HCPRIO ASSEMBLE)
  - Local modifications
- Servicing the CP Loader (HCPLDR)
- Creating the Stand-Alone Dump Utility (HCPSADMP EXEC)
- Assembling any CP modules
- Using CP exit routines
- Installing and servicing RTM/ESA

### RIO370 [3.1.0]

RIO370 is no longer supported.

### 64-Bit Support [3.1.0]

z/VM provides support for 64-bit addressing in:

- Storage configuration
- Display/dump/store of storage, registers, and PSWs
- Tracing

#### Storage Configuration

The CP DEFINE STORAGE command allows larger extents and discontinuous storage definitions. The display of the configurations has been changed.

Two new Directory Control Statements, STORAGE and MAXSTORAGE, allow you to define a default and a maximum virtual storage size for users.

The USER statement now allows a maximum storage of 16 exabytes.

#### Display/Dump/Store

All commands for displaying, dumping, or storing data or registers accommodate larger registers and storage areas.

## System Changes — Administration and Operation

### Tracing

Trace commands show new instructions, larger registers, PSWs, and storage areas. The new TRACE GG command allows you to trace changes in general-purpose registers for either a 64-bit or a 32-bit image of CP.

### Graphical User Interface (GUI) Facility Changes [3.1.0]

The following changes have been made to the GUI Facility:

- The GUI workstation agents, along with their Help files, are not shipped with z/VM. They are available with limited support from the VM Download Library at:  
<http://www.ibm.com/s390/vm/download>
- The CMSDESK application modules and message repository are also available as part of the GUI package from the VM Download Library.
- The CMSDESK command and nucleus routines remain in CMS. Documentation is available from the GUI download package. To use these functions, you must download the workstation agents and GUI modules and put them on a CMS minidisk or SFS directory.

### VMLINK Improvements [3.1.0]

The VMLINK command has been overhauled with both code and documentation changes. Many enhancements and some additions were made to increase the usability of this command:

- Changes were made to better validate some of the values in the VMLINK CONTROL file.
- The PUSH and POP options were changed to ensure the behaviors were consistent and could be documented.
- File mode support was made more complete and the documentation lists all the supported file mode definitions.
- Previously, when multiple INVOKE commands were processed and one or more failed, it was difficult to determine which ones failed because only one message and one return code were issued. VMLINK has been updated to return an error message and return code for each failing INVOKE command.
- Previously, the parameters passed to exits were automatically uppercased before the exit was called. These parameters will now remain in the case entered by the user, allowing mixed-case strings to be passed.
- The .EX and .PX VMLINK variables have been added to complete support for all possible NAMES file tags to be retrieved.
- Messages displayed when disks and directories are detached and released were changed for consistency.
- Message DMS2062I has been improved to include the name of the nickname which is being used when the QUERY option is being processed. This can be especially useful when more than one nickname is specified.
- The documentation has been completely reorganized. Guide information has been removed from the command description in the *z/VM: CMS Command Reference* and moved to the *z/VM: CMS User's Guide*.

---

## Guest Support

The following changes may affect how you run guests on the new system (compared to your old system).



## Multi-Path Lock Facility (MPLF) Support [2.2.0]

The 3990 Model 6 Multi-Path Lock Facility (MPLF) provides function to control processes and share data in a loosely coupled environment. VM now provides support to allow dedicated devices and full-pack minidisks to use this real hardware facility. The addition of this support allows Transaction Processing Facility (TPF) systems running as guests on VM to share data with native TPF systems.

MPLF controls locking through a set of channel commands implemented by the 3990 Model 6 DASD control unit. These commands result in a setting which indicates the requested operation is either compatible or incompatible with the current state of the lock. The control unit maintains the names and status of the locks in use and responds to requests to obtain or release a lock. The control unit also notifies a host when it permits lock ownership that was previously denied. A host can obtain a lock, release a lock, examine the status of active locks, and check the outcome of lock-related operations using the channel commands.

For more information, see the SET LKFACR and QUERY LKFACR commands in the *z/VM: CP Command and Utility Reference*.

## Guest Coupling Simulation Support [2.3.0]

VM guest coupling simulation provides for the simulation of one or more complete parallel sysplex systems within a single VM system image. This environment allows the testing and debugging of guest operating systems while running under VM. Guest coupling simulation is supported on the following processors (at the appropriate engineering-change levels):

- IBM S/390 Parallel Enterprise Server - Generation 3 and later
- IBM S/390 Multiprise® 2000 and later

VM guest coupling simulation support simulates a sysplex environment by using software equivalents of the real hardware and software requirements. The hardware is simulated by the message facility support in CP. Other virtual machines, referred to as coupled guests, are set up to run the guest operating systems (that is, MVS™ or OS/390). A special virtual machine called a Coupling Facilities (CF) service machine is defined and runs the coupling facility code. The coupled guest machines and the CF service machines are coupled together by a special message facility environment that passes information back and forth between the CF service machines and the coupled guests. The CF service machines manage data movement, scheduling, and locks, and maintain the status of the entire sysplex environment.

This facility also provides a CP command set to allow you to perform operations required to control the coupled guest environment:

- DEFINE MSGPROC
- DETACH MSGPROC
- QUERY VIRTUAL MSGDEVICES
- QUERY VIRTUAL MSGPROC
- RESTART MSGPROC
- SET MSGFACIL

These commands allow you to add and remove links to the CF service machines, request status from the CF service machines, and control the message facility

## System Changes — Guest Support

environment. Also, a new CP SET VTOD command has been added to allow Year 2000 testing of a parallel sysplex environment on VM.

For more information, see:

- *z/VM: Planning and Administration*
- *z/VM: CP Command and Utility Reference*
- *z/VM: Running Guest Operating Systems*
- *z/VM: Diagnosis Guide*

## Guest Support for QDIO Facility and OSA Express Gigabit Ethernet [2.4.0]

VM provides guest support for the Queued-Direct-I/O (QDIO) Facility on processors that support this new I/O architecture. The QDIO Facility allows a program to directly exchange data with an I/O device without performing traditional S/390 I/O instructions. To exchange data, both the I/O device and the program reference main storage directly through a set of data queues.

VM provides guest (dedicated device) support and dynamic I/O support for the new OSA Express Gigabit Ethernet (OSA-Express GbE). A new channel path ID has been defined for this device.

The following CP commands have been updated:

- DEFINE CHPID / PATH
- QUERY VIRTUAL OSA
- SET CPTRACE

For more information, see the *z/VM: CP Command and Utility Reference*.

## Guest Cryptographic Support [2.4.0]

This support provides guest virtual machine access to the S/390 CMOS Cryptographic Coprocessor by extending the existing VM guest cryptographic support for Bipolar processors. The new cryptographic support is upwardly compatible with the existing Bipolar support.

This support is intended primarily for use by OS/390 Integrated Cryptographic Service Facility (ICSF) applications running in an OS/390 guest of VM. ICSF is currently the only IBM application program interface to the S/390 CMOS cryptographic hardware. The following commands and statements were updated:

- CP DEFINE CRYPTO command
- CP QUERY CRYPTO command
- CP QUERY VIRTUAL CRYPTO command
- CP SET CRYPTO command
- CPU directory control statement
- CRYPTO directory control statement

For more information, see:

- *z/VM: CP Command and Utility Reference*
- *z/VM: Planning and Administration*



## Support for IEEE Floating Point [2.4.0]

New function has been added to support IEEE Floating Point hardware on servers that provide this capability. CP has been updated to allow multiple levels of guest operating systems to use basic floating point extensions, floating point support extensions, hexadecimal floating point extensions, and binary floating point.

This support includes preservation and restoration of 16 floating point registers (Additional Floating Point registers 1,3,5,7,8-15 plus existing floating point registers 0,2,4,6) and the Floating Point Control (FPC) register which is provided by the IEEE Floating Point hardware. The following were updated:

- CP DISPLAY Registers command
- CP STORE (Registers) command
- CP STORE STATUS command
- CP TRACE command
- Messages HCP6153E and HCP6154E (new)

For more information, see the *z/VM: CP Command and Utility Reference*.

**Note:** Applications that exploit the IEEE Floating Point hardware require the IBM High Level Assembler Version 1 Release 3.0.

## Parallel Access Volumes Support [3.1.0]

z/VM provides guest support for the Parallel Access Volumes feature. This feature allows the configuration of logical volumes (known as alias Parallel Access Volumes), where each logical volume (alias) has a unique device address but is actually an exposure of the existing real device (known as the base Parallel Access Volume). This allows the host to issue concurrent I/O requests to one real device, the base volume, through the different alias volumes.

Support for Parallel Access Volumes includes:

- The new CP QUERY PAV command, which displays information about the Parallel Access Volume devices on the system.
- Enhancements to the CP QUERY DASD DETAILS command to display additional information if the queried device is a Parallel Access Volume.
- A new CP Monitor Record, which has been added to Domain 6 (I/O) to record state change interrupts that indicate a change in the Parallel Access Volumes information:
  - Record 20 – MRIODSTC – State change

Other Parallel Access Volumes information will be recorded in the existing Device Configuration Data Record (Domain 1, Record 6) and the Vary On Device - Event Data Record (Domain 6, Record 1).

## Tape Support Enhancements [3.1.0]

z/VM provides additional guest support for the IBM 3494 Virtual Tape Server (VTS):

- Peer-to-Peer VTS provides flexible component placement to meet 7x24 processing requirements and provides a solution for remote backup and recovery.
- Import/Export allows physical tapes to be removed from and inserted into the VTS logical tape library.

## System Changes — Guest Support

z/VM supports guest use of IBM 3590 A60 tape controllers attached with FICON™ channels when such guests themselves support the 3590 A60 on native FICON channels.

### z/Architecture Support [3.1.0]

z/Architecture (64-bit) capabilities are supported for guest operating systems.

---

## Networking Support

The following changes may affect how the new system connects to other systems and applications (compared to your old system).

**Note:** This section does not describe changes to TCP/IP or discuss TCP/IP migration issues. For that information, see *z/VM: TCP/IP Level 3A0 Planning and Customization*.

### Java™ and NetRexx™ Support [2.3.0]

VM has the capability of developing, compiling, and running Java and NetRexx applications. You run the applications in the OpenExtensions Shell and Utilities environment. You can also write and compile Java applets, but you must run these applets on another platform, such as OS/2® Warp, Windows® 95, or Windows NT®.

### MQSeries® Client for VM/ESA [2.3.0]

MQSeries Client for VM/ESA enables applications using message queuing to communicate across different platforms using client-server technology.

MQSeries Client for VM/ESA executes in any CMS virtual machine and provides a simple user interface to the server environment. It is not a full function queue manager. Therefore it needs to be connected through TCP/IP or APPC to other queue managers.

The application Language Bindings supported are:

- IBM C for VM/ESA
- IBM VS Cobol II
- IBM PL/I Compiler
- REXX/VM
- IBM Assembler

For more information, see the *MQSeries: Application Programming Guide*, SC33-0807.

---

## Support for Hardware Architectures and Facilities

The following changes may affect how the new system exploits new hardware architecture and facilities (compared to your old system).

**Notes:**

1. This section does not discuss support for new processors or devices. For lists of supported processors and devices, see *z/VM: General Information*.
2. Support provided only for guests is described under “Guest Support” on page 12 .

## S/390 Open Systems Adapter Support Facility (OSA/SF) [2.2.0]

The IBM S/390 Open Systems Adapter (OSA) is an integrated hardware feature that allows the S/390 platform to provide industry-standard connectivity directly to clients on local area networks (LANs) and wide area networks (WANs). The Open Systems Adapter Support Facility (OSA/SF) is a host-based tool supplied with VM that allows you to customize an OSA's modes of operation. You can access OSA/SF by a CMS user ID, by a REXX call to the OSA/SF API, or through the OSA/SF Windows 95, Windows NT, or OS/2 graphical user interface (GUI).

For more information, see:

- *S/390: Planning for the S/390 Open Systems Adapter (OSA-1, OSA-2) Feature*
- *VM/ESA: Open Systems Adapter Support Facility User's Guide for OSA-2*

## Integrated Cluster Bus Channel Support [2.4.0]

VM provides support for integrated cluster bus channels for S/390 CMOS processors that support this channel type. The existing dynamic I/O configuration support has been updated to allow VM, when running in an LPAR controlling the dynamic I/O configuration changes, to define integrated cluster bus channels for an OS/390 LPAR on the same Central Electronics Complex (CEC).

## Fibre Connection (FICON) Channel Support [2.4.0]

VM provides support for FICON channels for processors that support this channel type. FICON channel technology can improve system performance and total aggregate system bandwidth. Each FICON channel provides the equivalent of eight ESCON<sup>®</sup> channels. VM support includes:

- Dynamic I/O configuration support for the new FICON and FICON-converter channel types
- Support for the fibre-channel-to-ESCON converter function on the 9032-5 switch
- Support for the FICON 9042-1 switch
- Support for the S/390 architecture changes in the SCHIB, ORB, and IRB
- A new CP monitor record, Extended Channel Path Measurement Data

## Extended-TOD-Clock Support [2.4.0]

The Extended-TOD-clock facility is a hardware facility available on certain processors which provides a 128-bit Time of Day (TOD) clock. VM supports the use of the Extended-TOD-clock facility from XA, ESA, and XC virtual machines.

## Flash Copy Support [3.1.0]

z/VM allows a native CP user to initiate a Flash Copy function (an instant copy of a disk or data set) of a source device to a target device on an IBM Enterprise Storage Server<sup>™</sup>. Customers will find this feature especially beneficial for large databases, which normally require a long time to copy. Flash Copy support includes the new CP FLASHCOPY command. For more information, see the *z/VM: CP Command and Utility Reference*.

---

## Application Development and Deployment

The following changes may affect how the new system supports application development and deployment (compared to your old system).

### REXX Sockets [2.2.0]

The REXX Sockets API allows you to write socket applications in REXX for the TCP/IP environment. This allow you to use REXX to implement and test TCP/IP applications. A new REXX/VM external function, SOCKET, is provided. The SOCKET function uses the TCP/IP IUCV API to access the TCP/IP internet socket interface. The REXX socket functions are similar to socket calls in the C programming language. For more information, see the *z/VM: REXX/VM Reference*.

### Querying the CMS Level [2.2.0]

The CMSPROG field of NUCON, returned in register 1 by the CMS command QUERY CMSLEVEL when the command is invoked internally by a program, has been frozen at X'0F', the value for CMS level 12 (VM/ESA 2.1.0). The CMSLEVEL assembler macro will not map CMS levels beyond CMS level 12. To determine the CMS level, use the DMSQEFL CSL routine or the DMSQEFL assembler macro.

### Pseudo Timer Extended [2.2.0]

DIAGNOSE code X'270' (Pseudo Time Extended) can be used to replace DIAGNOSE code X'0C' (Pseudo Timer). DIAGNOSE code X'0C' returns the time only in SHORTDATE format. DIAGNOSE code X'270' returns the same information as DIAGNOSE code X'0C' plus two additional fields containing the time in FULLDATE format and ISODATE format.

### Calling CSL Routines That Have Dates as Output [2.2.0]

The following CSL routines, which provide dates in their output, have been enhanced with a new parameter for specifying the format in which the dates are to be returned:

- DMSEXIDI
- DMSEXIFI
- DMSOPBLK
- DMSGETDA
- DMSGETDF
- DMSGETDS
- DMSGETDX

The date formats are:

<b>Keyword</b>	<b>Format</b>
<b>SHORTDATE</b>	<i>yy/mm/dd</i>
<b>FULLDATE</b>	<i>yyyy/mm/dd</i>
<b>ISODATE</b>	<i>yyyy-mm-dd</i>

If no keyword is specified, the default is SHORTDATE, which is the format compatible with prior VM releases. The output field in which the date is returned is either 8 or 10 characters in length, depending on the format requested.

In REXX, the date field returned is always 10 characters. If SHORTDATE is specified or allowed to default, the 8-character date is padded on the right with two blanks. An incompatibility is that old REXX programs which now get the date returned as an 8-character string will start getting the date returned as a 10-character string.

If you call one of these routines from a higher level language, the date field is not padded. For SHORTDATE, an 8-character field is returned. For FULLDATE or ISODATE, a 10-character field is returned. Therefore, if you specify FULLDATE or ISODATE, you must be sure to also specify a 10-character output field. Otherwise, you could get storage overlays.

### OS Simulation Support for Tape Library Dataserver [2.2.0]

CMS OS simulation has been enhanced to call DFSMS/VM Removable Media Services (RMS) CSL routines to mount and demount tapes on Tape Library Dataserver machines. The new LIBSRV option on the CMS FILEDEF command is used to indicate that a tape is under the control of a Tape Library Dataserver. OS simulation can also determine that a tape is under Dataserver control if the user has issued the DFSMS/VM MOUNT command to premount the tape before the CMS tape processing function is invoked.

When a tape is under Dataserver control:

- OS simulation calls the RMS FSMRMDMT (Demount) and FSMRMMNT (Mount) routines to get subsequent multivolume tapes mounted for the user through the native DMSTVS mounting service and the CMS native rewind and unload tape processing functions.
- The RUN (rewind and unload) function of the CMS TAPE and VMFPLC2 commands and the TAPECTL macro calls the RMS FSMRMDMT routine to demount the tape.

See the following books for more information:

- *z/VM: CMS Application Development Guide for Assembler*
- *z/VM: CMS Command Reference*
- *VM/ESA: DFSMS/VM Function Level 221 Removable Media Services User's Guide and Reference*

### GCS Pathlength Reduced [2.2.0]

GCS processor usage has been reduced, which improves the overall performance of GCS. This can also improve the performance of applications and program products that use GCS.

### Full MP Capability for VMCF [2.2.0]

The Virtual Machine Communications Facility (VMCF) has been changed so that it no longer has to run on the master processor. This can improve performance and capacity for systems that run applications (such as TCP/IP, OV/VM, and RACF®) that use VMCF.

### Language Environment® [2.3.0]

Language Environment is now included with VM. Language Environment provides a common run-time environment for programs generated with C for VM/ESA and other high-level languages.

### Dynamic Link Libraries [2.3.0]

Dynamic link library (DLL) support is available for applications running under VM and Language Environment.

A DLL is a collection of one or more functions or variables gathered in a load module and executable or accessible from a separate application load module. The

## System Changes — Application Development

term derives from the fact that the connection or link between the application that uses the DLL and the DLL functions or variables is made dynamically while the application is executing rather than statically when the application is built.

For more information, see:

- *z/VM: CMS Application Development Guide*
- *z/VM: OpenExtensions Callable Services Reference*
- *C for VM/ESA: Library Reference*

### Non-Relocatable Modules Loaded between 16-20MB [2.3.0]

Non-relocatable modules that are loaded in the virtual machine between 16-20MB must be regenerated. The CMS nucleus now spans from 15-20MB, which will prevent these modules from being loaded. You must either regenerate the non-relocatable modules at a different storage location or generate the modules as relocatable modules using the RLDSAVE option on the CMS LOAD command. See the *z/VM: CMS Command Reference* for information on generating modules using the LOAD and GENMOD commands.

### CMS Migration Utility Feature No Longer Available [2.3.0]

Beginning with VM/ESA 2.1.0, which became generally available in December 1995, CMS no longer runs in 370-mode virtual machines. To simplify the migration of older applications and to facilitate the running of 370-mode-only CMS applications in non-370-mode virtual machines (that is, in XA or XC mode) IBM provides the 370 Accommodation Facility. This function originally shipped with VM/ESA 1.2.1 (available July 1993) and has since been enhanced as the result of customer experience and input. 370 Accommodation handles the vast majority of inconsistencies between 370 mode and either XA or XC mode, eliminating the need to change these applications to exploit the new architectures.

IBM also provided the 370-capable level of CMS which shipped with VM/ESA 1.2.2 (known as CMS 11) as a no-charge feature for customers with specialized applications which 370 Accommodation could not support. Few VM customers have needed this CMS Migration Utility Feature and IBM has not updated it. Most importantly, the CMS Migration Utility Feature has not been made Year 2000 Ready. IBM recommends that customers who are using the CMS Migration Utility should modify their applications to run on a Year 2000-Ready level of CMS (VM/ESA 2.2.0 or later) and stop using the CMS Migration Utility as an application environment. Customers should not rely upon the CMS Migration Utility as a production environment. Also, IBM has announced the discontinuance of service for VM/ESA 1.2.2 (including CMS 11) and the CMS Migration Utility Feature as of April 30, 1999.

Consistent with IBM's policy of delivering only Year 2000-Ready products to customers in 1998 and beyond, the CMS Migration Utility Feature will no longer be available as part of VM. Customers who find that their applications cannot run under the current level of CMS should modify their applications if possible, or use the 370 Accommodation function. If problems persist, customers should call IBM Service for assistance.

### Pipelines Code Bases Merged [2.3.0]

The code bases for CMS Pipelines and CMS/TSO Pipelines have been merged:

- All pipelines that were written using previous levels of CMS Pipelines or CMS/TSO Pipelines should operate successfully with this new code base.



## System Changes — Application Development

- All internal EXECs, messages, and modules of CMS Pipelines have been renamed from a DMS to an FPL prefix. Message numbers and text have changed.

### Notes:

1. User-written applications that are sensitive to these changes may require alterations. Published externals (such as PIPGFTXT) have **not** been changed.
  2. For cross-references between the old (DMS) and new (FPL) Pipelines message numbers, see “Appendix A. CMS Pipelines Message Cross-Reference [2.3.0]” on page 99.
- All stages, commands and subcommands documented in the *CMS/TSO Pipelines: Author’s Edition* are now supported. Before the merge of the code bases, only the stages and subcommands documented in the *VM/ESA: CMS Pipelines Reference* were supported.
  - Some new function exists as a result of the code merge. Stages that are new or changed include:
    - AHELP
    - CONFIGURE
    - DATECONVERT
    - HOSTBYADDR
    - HOSTBYNAME
    - HOSTID
    - HOSTNAME
    - IP2SOCKA
    - SOCKA2IP
    - SPILL
    - TCPCLIENT
    - TCPDATA
    - TCPLISTEN
    - GETRANGE
    - SCANRANGE
    - SCANSTRING

Specific information about some enhancements can be found in PIPELINE NEWS, which is accessible from the internet at the following URL:

<http://pucc.princeton.edu/%7Epipeline>

- Any CMS Pipelines stages, commands, and subcommands that are not documented in the *z/VM: CMS Pipelines Reference* can be found in the *CMS/TSO Pipelines: Author’s Edition*, which is now included with the z/VM 3.1.0 library.

## OpenEdition<sup>®</sup> Enhancements [2.3.0]

Enhancements to OpenEdition include additional C/VM™ library routines, new callable services and mapping macros, new OPENVM commands for fork processing, and new shell commands for data compression.

### Additional C/VM Library Routines

Over 150 C library routines have been added to the C/VM run-time library. For more information about these routines, see the *C for VM/ESA: Library Reference*.

### New Callable Services and Macros

The following callable services have been added:

## System Changes — Application Development

Service	Purpose
DLL_delete (BPX1DEL)	Delete a previously-loaded program from storage
DLL_load (BPX1LOD)	Load a program into storage
fork (BPX1FRK)	Create a new process
msgctl (BPX1QCT)	Control message queues
msgget (BPX1QGT)	Create or find a message queue
msgrcv (BPX1QRC)	Receive a message from a message queue
msgsnd (BPX1QSN)	Send a message to a message queue
semctl (BPX1SCT)	Control semaphores
semget (BPX1SGT)	Create or find a set of semaphores
semop (BPX1SOP)	Perform semaphore operations atomically
shmat (BPX1MAT)	Attach a shared memory segment
shmctl (BPX1MCT)	Control shared memory segments
shmdt (BPX1MDT)	Detach a shared memory segment
shmget (BPX1MGT)	Create or find a shared memory segment
wait-extension (BPX1WTE)	Obtain status information about child processes
w_getipc (BPX1GET)	Query interprocess communications

The following mapping macros have been added:

Macro	Purpose
BPXYIPCP	Map interprocess communications permissions
BPXYIPCQ	Map the data structure used by w_getipc (BPX1GET)
BPXYMSG	Map interprocess communications message queues
BPXYSEM	Map interprocess communications semaphores
BPXYSHM	Map interprocess communications shared memory segments
BPXYSINF	Map the Siginfo_t structure used by wait-extension (BPX1WTE)

For additional information about these routines and macros, see the *z/VM: OpenExtensions Callable Services Reference*.

### OPENVM Commands for fork (BPX1FRK) Processing [2.3.0]

Because the VM implementation of the fork (BPX1FRK) service does not meet all POSIX.1 requirements, you must explicitly turn fork (BPX1FRK) processing ON with the CMS OPENVM SET FORK command before running your program. You can use the OPENVM QUERY FORK command to determine the current setting.

For information about restrictions in the VM implementation of fork (BPX1FRK), see the *z/VM: OpenExtensions Callable Services Reference*. For more information about OPENVM commands, see the *z/VM: OpenExtensions Command Reference*.

### Shell Commands for Data Compression [2.3.0]

The following shell commands have been added to the OpenEdition Shell and Utilities:

- **compress**



- **uncompress**
- **zcat**

The **compress** and **uncompress** commands use Lempel-Ziv compression techniques to compress and uncompress data in files or from the standard input. The **zcat** command calls **uncompress** to uncompress data from one or more files or from the standard input and writes it to the standard output.

For more information about these commands, see the *z/VM: OpenExtensions Command Reference*.

## OS Simulation Enhancements [2.4.0]

CMS OS Simulation support has been enhanced as follows:

- The CMS FILEDEF command has been changed to allow record length (LRECL) definitions up to 65535 bytes for variable spanned OS records and non-OS CMS files, and block size (BLOCK or BLKSIZE) definitions up to 65535 bytes for non-OS CMS files.
- The CMS LKED command has been changed to allow larger default work area sizes (SIZE option): 400K for *value1* and 100K for *value2*.
- The CMS MOVEFILE command has been changed to allow the processing of QSAM variable spanned records up to 65535 bytes in length under the Extended Logical Record Interface (XLRI). MOVEFILE can also process non-OS CMS files up to 65535 bytes.
- The CMS QUERY FILEDEF command has a new optional operand, ATTRIBUT, which allows you to display the RECFM, LRECL, and block size attributes associated with the current FILEDEF.
- OS Simulation DCB macro processing has been changed to allow it to describe and pass both LRI and XLRI conventions for QSAM variable spanned long records, up to 65535 bytes in length, for subsequent OPEN, CLOSE, GET, or PUT processing. DCB can now also describe non-OS CMS files up to 65535 bytes in length.
- The OS Simulation OPEN, CLOSE, GET, and PUT macros, and the SVC 19 and SVC 20 supervisor calls, have been changed to allow the processing of OS formatted variable spanned QSAM records and non-OS CMS files up to 65535 bytes in length.
- The OS Simulation tape processing routines, such as DMSTVS, will issue a new message, DMS2139I, if SENSE data from a tape mount indicates that the mounted tape cartridge may be incorrect for the tape device in use. These tape processing routines may be invoked by MOVEFILE processing and by the following OS Simulation macros: OPEN, CLOSE, GET, PUT, READ, WRITE, and FEOV.

For more information, see:

- *z/VM: CMS Application Development Guide for Assembler*
- *z/VM: CMS Command Reference*

## OpenEdition Enhancements [2.4.0]

### New realpath (BPX1RPH) Callable Service

The realpath (BPX1RPH) service finds the absolute path name for a specified relative path name. Any dot (.) or dot dot (..) components, symbolic links, or mount external links included in the relative path name input are resolved in the absolute path name output.

## System Changes — Application Development

### **New setopen (BPX1VM6) Callable Service**

The setopen (BPX1VM6) service sets certain flags specific to the OpenEdition platform without creating a new POSIX process in the virtual machine. The function codes used in the setopen service are mapped by the BPXYVM6 macro.

### **OpenEdition Renamed to OpenExtensions™ [3.1.0]**

In z/VM, support for POSIX and XPG standards is called OpenExtensions. In VM/ESA, this support was called OpenEdition for VM/ESA. No modifications are necessary to run OpenEdition applications on z/VM. OpenExtensions Services and OpenExtensions Shell and Utilities are now included in CMS.

### **Open Files Limit Increased [3.1.0]**

Under OpenExtensions, the limit on open files has been increased from 1024 to 64K (65536).

### **BFS Root Not Case Sensitive [3.1.0]**

The fully qualified VM byte file system root (*././VMBFS:filepoolid:filespaceid*) is no longer case sensitive.

### **Binder/Loader Support [3.1.0]**

CMS hosts the DFSMS/VM® Program Management binder and provides a native implementation of the corresponding program object loader functions. Included with this support is a new CMS command, BIND, to invoke the services of the binder to bind and store an executable file containing a program object on DASD (minidisk, SFS directory, or BFS directory). An executable file produced by the binder may be used on a CMS system containing the program object loader in exactly the same way as a conventional MODULE file generated by the CMS GENMOD command.

Also included in this support is an enhancement to the **c89** command to use the new BIND command in place of the combination of the Language Environment prelinker and the old LOAD, INCLUDE, and GENMOD commands. In addition, the complete binder API is available to application programs that need to invoke the services of the binder dynamically.

CMS support of the binder and loader allows ready transportability of executable files between z/VM and OS/390 or z/OS systems and provides the infrastructure to permit CMS to exploit new and emerging programming language technology being developed for z/OS.

For more information, see *z/VM: Program Management Binder for CMS*.

### **CMS OS Simulation Enhancement [3.1.0]**

The new SET TAPENEVR command allows you to control CMS OS Simulation tape label date checking for 'Unexpired Files'. The new QUERY TAPENEVR command allows you to display the current setting.

For more information, see:

- *z/VM: CMS Application Development Guide for Assembler*
- *z/VM: CMS Command Reference*

## Tape FICON and RAS Support [3.1.0]

Extended function in DIAGNOSE code X'210' allows application programs to find out the underlying real hardware that is associated with tape devices operating in emulation mode. For more information, see *z/VM: CP Programming Services*.

## System Diagnosis

The following changes may affect how you diagnose problems on the new system (compared to your old system).

## FST and ADT Macro Changes to Support 4-digit Years [2.2.0]

### FST (File Status Table)

When you access a disk or SFS directory, a file directory is stored in your virtual machine. The entries in the file directory for each CMS file are called the File Status Table (FST). The FST describes the attributes of the file. One of the attributes of a file is date/time of last update. This is currently stored in 6 bytes (*yy mm dd hh mm ss*), where each byte holds two decimal digits. This is the date and time that the accessed file was last updated.

In VM/ESA 2.2.0, support was added for 4-digit years by adding a century flag FSTCNTRY (X'08') in the FST flag byte (FSTFLAGS) for both FST forms (see table below). This is bit 4, which identifies the century in which the file was last written or updated. If bit 4 is off, the year is in the 1900s. If bit 4 is on, the year is in the 2000s. Therefore, the range of years supported is from 1900 to 2099.

If you are interested in the existing flag settings for the FSTFLAGS field, see the usage notes for the FSSTATE macro in the *z/VM: CMS Macros and Functions Reference*.

If you are using the FST to retrieve the date and time of last update in your application, you can add support to use an appropriate CSL routine (for example, DMSGETDI, DMSEXIST, or DMSERP) or you can use the FSSTATE macro. For information about CSL routines, see the *z/VM: CMS Callable Services Reference*. For information about FSSTATE, see the *z/VM: CMS Macros and Functions Reference*.

Table 2. Base versus Extended FST Format. Note that not all field names are shown.

Hex Disp	Dec Disp	Size	Base	Ext	Field Name	Field Description
0	0	16	B	E	FSTDFNFT	File name File type
		8	B	E	FSTFNAME	File name
		8	B	E	FSTFTYPE	File type
10	16	2	B		FSTDATEW	Date ( <i>mm yy</i> ) last written
1F	31	1	B	E	FSTFLAGS	FST flag byte
		Bit 4	B	E	FSTCNTRY	X'08' Century last written (0 - 19 <i>nn</i> , 1 - 20 <i>nn</i> )
26	38	2	B		FSTYEARW	Year ( <i>yy</i> ) last written
36	54	6		E	FSTADATI	Alternate date/time ( <i>yy mm dd hh mm ss</i> )
4E	78	1		E	FSTFB3	FST flag byte 3
		Bit 4		E	FSTCDOLR	X'08' Century date of last reference (0 - 19 <i>nn</i> , 1 - 20 <i>nn</i> )

## System Changes — Diagnosis

Table 2. Base versus Extended FST Format (continued). Note that not all field names are shown.

Hex Disp	Dec Disp	Size	Base	Ext	Field Name	Field Description
54	84	3		E	FSTDOLR	Date of last reference
<b>Note:</b> FSTFB3, FSTCDOLR, and FSTDOLR apply only to an SFS FST.						

Also, for an SFS FST, a new flag FSTCDOLR (X'08') is added in the FSTFB3 flag byte 3. This is the flag that indicates the Century for Date of Last Reference (0 indicates the year is in the 1900s, 1 indicates the year is in the 2000s) that corresponds to FSTDOLR.

### ADT (Active Disk Table) - Disk Label

The ADTSECT maps information in the active disk table (ADT). It also contains information about the disk label. One of the fields contained in the disk label is ADTDCRED, which is the creation date and time of the minidisk. Following the ADTDCRED field is a new flag byte for the volume label called ADTFLGL, which contains the new ADTCNTRY flag. The ADTCNTRY flag is X'01', and corresponds to the ADTDCRED field. If the value of this flag is 0, it indicates that the creation year is in the 1900s. If the value is 1, it indicates the creation year is in the 2000s.

## Viewing and Printing CP and CMS Control Blocks [2.3.0]

CP and CMS control block documentation is now available on the VM home page:

<http://www.ibm.com/servers/eserver/zseries/zvm/pubs>

You can view or print an entire control block or select general sections.

In addition, VM provides an unsupported tool, DACBGEN EXEC, that you can use against CP and CMS control blocks on your system to get the most current data. You can also use DACBGEN against user blocks if they follow the prescribed formula. You can tailor DACBGEN to your own environment.

## Dump Viewing Facility No Longer Supports CP Dumps [3.1.0]

The Dump Viewing Facility no longer supports CP dumps; it supports only VM (virtual machine) dumps. The VM Dump Tool is the dump viewing program to use for z/VM CP dumps.

## VM Dump Tool [3.1.0]

The VM Dump Tool assists in analyzing dump data from a dump file created by the DUMpload utility. The VM Dump Tool provides a variety of subcommands and macros that allow you to display, locate, and format dump data interactively. This tool can process CP stand-alone dumps, CP ABEND dumps, and virtual machine dumps of a CP system.

For more information, see *z/VM: VM Dump Tool*.

---

## Other Changes

The following changes may also affect how you run the new system (compared to your old system).

## Euro Support [2.4.0]

Support has been added to the following:

- Almost 200 new translation tables have been added to VM TCP/IP.
- A new CSL routine, DTCXLATE, provides an application programming interface (API) for programs to use the translation information contained in the TCP/IP translation tables.
- OPENVM GETBFS and PUTBFS commands were updated, increasing the number of code pages that could be specified.
- The FTP server and VM/ESA 2.3.0 TCP/IP FL310 NFS feature have been updated to allow specification of particular translation tables.
- Language Environment V1R6 and V1R8
- LANRES
- LFS
- The TCP/IP LPR client and LPD server applications have been enhanced to provide users with the ability to specify alternate translation tables.
- RSCS LPR/LPD
- SMTP

For more information, see the euro link on the VM home page:

<http://www.vm.ibm.com/euro>

## System Changes — Other

---

## Chapter 3. Changes to External Interfaces

This chapter identifies the external interfaces that have changed since VM/ESA 2.1.0. The release in which each change occurred is indicated, and each change is identified as either **upwardly compatible** or **incompatible**. Use the information provided in this chapter to determine if you need to make any changes in the way you use these interfaces. See “Compatibility Terms Used in This Book” on page 2 for the meaning of the compatibility terms.

### Notes:

1. For changes that occur every release, such as responses from commands that query the level of the system, only the latest release is indicated.
2. Changes from post-release APARs are identified as changes in the successive release.

This chapter contains the following major sections:

- CP Changes
- “CMS Changes” on page 44
- “REXX/VM Changes” on page 61
- “GCS Changes” on page 62
- “TSAF Changes” on page 62
- “AVS Changes” on page 62
- “VMSES/E Changes” on page 62
- “Dump Viewing Facility Changes” on page 64

---

### CP Changes

This section identifies the changes to CP external interfaces. It contains the following subsections:

- System Configuration File Statements
- “User Directory Control Statements” on page 30
- “CP Commands” on page 30
- “CP Utilities” on page 41
- “DIAGNOSE Codes” on page 41
- “CP Macros” on page 43
- “CP Messages” on page 43

### System Configuration File Statements

Table 3 lists the system configuration file statements that have changed. For additional information, see *z/VM: Planning and Administration*.

Table 3. Changes to System Configuration File Statements

Statement	Changes
CHARACTER_DEFAULTS	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"><li>• [2.3.0] Letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'OF' (shift in) cannot be defined as the system default line edit symbols (line-delete, character-delete, escape, line-end, and tab).</li></ul>

## Interface Changes – CP

Table 3. Changes to System Configuration File Statements (continued)

Statement	Changes
CPXLOAD	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [3.1.0] New filetype options: &amp;TEXT, &amp;TXTLIB.</li></ul>
FEATURES	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [3.1.0] New options for automatic start and restart, IPL message control, and forced disconnect timeout.</li></ul>
SYSTEM_RESIDENCE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [3.1.0] New VOLID options: &amp;SYSRES, &amp;SYSPARM.</li></ul>
USER_DEFAULTS	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [3.1.0] New parameter: MESSAGE_LEVEL.</li></ul>

## User Directory Control Statements

Table 4 lists the user directory control statements that have changed. For additional information, see *z/VM: Planning and Administration*.

Table 4. Changes to User Directory Control Statements

Statement	Changes
CRYPTO	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.4.0] New operand: MODIFY.</li></ul>
MDISK	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [3.1.0] New volume identification option: &amp;SYSRES.</li></ul>
OPTION	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.2.0] Current LKFAC operand authorizes full-pack minidisks and devices for real MPLF use. For dedicated devices, MPLF channel commands may now succeed where they used to fail. For full-pack minidisks, the change is transparent until you issue the new SET LKFACR command.</li><li>• [2.3.0] Specifying the TODENABLE operand allows a user to change the virtual machine TOD clock with the new CP SET VTOD command.</li><li>• [2.3.0] New operands: CFVM, CFUSER.</li><li>• [2.4.0] New operand: DIAG88.</li></ul>
SPECIAL	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"><li>• [3.1.0] Changed message: HCP2801E.</li></ul> <b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.3.0] New operand: MSGPROC.</li></ul>
USER	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"><li>• [2.3.0] Letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'OF' (shift in) cannot be defined as logical line edit symbols (line-end, line-delete, character-delete, and escape).</li></ul>

## CP Commands

Table 5 lists the CP commands that have changed. For additional information, see the *z/VM: CP Command and Utility Reference*.



Table 5. Changes to CP Commands

Command	Changes
ATTACH	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New messages: HCP6860E, HCP6863E, HCP6864E.</li> </ul>
AUTOLOG	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New message: HCP093E.</li> </ul>
BEGIN	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Supports hexadecimal storage locations up to 16 digits.</li> </ul>
COUPLE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New message: HCP6024E.</li> </ul>
CPLISTFILE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New operands to specify date format: SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] Responses include 4-digit years for FULLDATE and ISODATE operands.</li> </ul>
DEFINE (in general)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] New operand: MSGPROC.</li> <li>• [2.3.0] Additional messages: HCP045E, HCP260E, HCP1014E, HCP2800E, HCP2801E, HCP2802E, HCP2803E, HCP2804I, HCP2806E, HCP2811E.</li> <li>• [2.4.0] New operand: EXIT.</li> <li>• See DEFINE commands below.</li> </ul>
DEFINE CHPID / PATH	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New operands: INTEGRATED_SYSTEM_DEVICE, ISD.</li> <li>• [2.4.0] New operands: CLUSTER_BUS_SENDER_CHANNEL, CBS, FICON, FC, FICON_CONVERTER, FCV, OSA_DIRECT_EXPRESS, OSD, OSA_EXPRESS, OSE.</li> <li>• [2.4.0] New return codes for message HCP6806E.</li> <li>• [3.1.0] New operands: CLUSTER_BUS_PEER_CHANNEL, CBP, CLUSTER_BUS_RECEIVER_CHANNEL, CBR, COUPLING_FACILITY_RECEIVER_CHANNEL, CFR, DIRECT_SYSTEM_DEVICE, DSD, EMULATED_I/O, EIO, INTERNAL_COUPLING_PEER_CHANNEL, ICP, IOCLUSTER <i>iocname</i>, MANAGED, PEERPATH <i>nn</i>.</li> </ul>
DEFINE CRYPTO	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] New response.</li> <li>• [2.4.0] New message: HCP1716E.</li> </ul>
DEFINE CTCA	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operands: SCTC, BCTC, CNC.</li> </ul>
DEFINE CU / CNTLUNIT	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] CFS operand has been changed to CF because this operand now indicates that the control unit is connected to a coupling facility sender channel path or a cluster bus sender channel path, or both. However, the CFS form of the operand is still supported for compatibility.</li> <li>• [2.4.0] Range of permitted values for CU_LOGICAL_ADDRESS has been increased.</li> <li>• [3.1.0] New option for TYPE: *.</li> <li>• [3.1.0] New operand: MANAGED_PATHS <i>n</i>.</li> </ul>

## Interface Changes – CP

Table 5. Changes to CP Commands (continued)

Command	Changes
DEFINE DEVICE / IODEVICE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] CFS operand has been changed to CF because this operand now indicates that the control unit is connected to a coupling facility sender channel path or a cluster bus sender channel path, or both. However, the CFS form of the operand is still supported for compatibility.</li> </ul>
DEFINE STORAGE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operand: CONFIGURATION.</li> </ul>
DELETE (in general)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• See DELETE commands below.</li> </ul>
DELETE CHPID / PATH	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] New return codes for message HCP6806E.</li> </ul>
DELETE CU / CNTLUNIT	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] CFS operand has been changed to CF because this operand now indicates that the control unit is connected to a coupling facility sender channel path or a cluster bus sender channel path, or both. However, the CFS form of the operand is still supported for compatibility.</li> </ul>
DELETE DEVICE / IODEVICE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] CFS operand has been changed to CF because this operand now indicates that the control unit is connected to a coupling facility sender channel path or a cluster bus sender channel path, or both. However, the CFS form of the operand is still supported for compatibility.</li> <li>• [3.1.0] Additional format for message HCP6818E.</li> </ul>
DELETE RDEVICE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New message: HCP6862E.</li> </ul>
DETACH (in general)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] New operand: MSGPROC.</li> <li>• [2.3.0] Additional messages: HCP260E, HCP2805E, HCP2807E.</li> </ul>
DIAL	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] The fields of an IP address included in the response no longer contain leading zeros.</li> </ul>
DISCONNECT	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Time the system waits between disconnect and automatic logoff can be set on the FEATURES system configuration statement.</li> </ul>
DISPLAY (in general)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Message HCP6150E may specify z/Architecture mode.</li> <li>• [3.1.0] New operand: PSWG.</li> <li>• See DISPLAY commands below.</li> </ul>
DISPLAY ESA/XC Storage	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operands for indirect addressing and displaying data in ASCII format.</li> <li>• [3.1.0] Supports 64-bit addressing.</li> </ul>
DISPLAY Guest Storage	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operands for indirect addressing and displaying data in ASCII format.</li> <li>• [3.1.0] Supports 64-bit addressing.</li> <li>• [3.1.0] New z/Architecture variation.</li> </ul>

Table 5. Changes to CP Commands (continued)

Command	Changes
DISPLAY Host Storage	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operands for indirect addressing and displaying data in ASCII format.</li> <li>• [3.1.0] Supports 64-bit addressing.</li> </ul>
DISPLAY Linkage Stack	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] Response indicates the called-space identification (CSID) if the linkage-stack entry type is a program-call state entry with a called-space ID.</li> <li>• [3.1.0] New responses for 64-bit mode.</li> </ul>
DISPLAY PSW	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] For a z/Architecture guest, translates a z/Architecture format PSW into an ESA/390 format PSW (if the SET PSWTRANS ALL or SET PSWTRANS STORE command was previously issued).</li> </ul>
DISPLAY Registers	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] New operand: FPC.</li> <li>• [2.4.0] Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor.</li> <li>• [2.4.0] New response when FPC operand is used.</li> <li>• [2.4.0] New messages: HCP6153E, HCP6154E.</li> <li>• [3.1.0] New operands: GGreg1, GHreg1, XGreg1, XHreg1.</li> </ul>
DUMP (in general)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Message HCP6150E may specify z/Architecture mode.</li> <li>• [3.1.0] New operand: PSWG.</li> <li>• See DUMP commands below.</li> </ul>
DUMP ESA/XC Storage	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operands for indirect addressing and dumping data in ASCII format.</li> <li>• [3.1.0] Supports 64-bit addressing.</li> </ul>
DUMP Guest Storage	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operands for indirect addressing and dumping data in ASCII format.</li> <li>• [3.1.0] Supports 64-bit addressing.</li> <li>• [3.1.0] New z/Architecture version of this command.</li> </ul>
DUMP Host Storage	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operands for indirect addressing and dumping data in ASCII format.</li> <li>• [3.1.0] Supports 64-bit addressing.</li> </ul>
DUMP Linkage Stack	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] Response indicates the called-space identification (CSID) if the linkage-stack entry type is a program-call state entry with a called-space ID.</li> <li>• [3.1.0] New responses for 64-bit mode.</li> </ul>
DUMP Registers	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operands: GG, GH, XG, XH.</li> </ul>
INDICATE (in general)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• See INDICATE commands below.</li> </ul>

## Interface Changes – CP

Table 5. Changes to CP Commands (continued)

Command	Changes
INDICATE QUEUES	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Responses changed to support 64-bit architecture.</li> </ul>
INDICATE SPACES	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Responses changed to support 64-bit architecture.</li> </ul>
INDICATE USER	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Responses changed to support 64-bit architecture.</li> </ul>
IPL	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"> <li>• [2.2.0] New response is issued if tracing is active when a preferred guest is IPLed. Tracing must be turned off and the guest reIPLed.</li> <li>• [2.3.0] Previously, when you IPLed CMS with the PARM operand, CMS initialization attached a fence of 8 bytes of X'FF's to the end of the PARM data before passing it to the SYSPROF EXEC. Now, no fence is attached, and only the actual PARM data (up to 64 characters) is passed. If you have tailored your SYSPROF EXEC to use the fence to determine the end of the PARM data, you must modify your SYSPROF EXEC to use a different method.</li> </ul>
LOCATE (in general)	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• See LOCATE commands below.</li> </ul>
LOCATE CMDBK	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New message: HCP026E.</li> </ul>
LOCATE DGNBK	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New message: HCP009E.</li> </ul>
LOCATE FILID	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New message: HCP026E.</li> </ul>
LOCATE FRAMETBL	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New message: HCP026E.</li> <li>• [3.1.0] New response formats for 64-bit addresses.</li> </ul>
LOCATE LDEV	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New message: HCP026E.</li> </ul>
LOCATE (Storage)	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New operand: ASCII.</li> <li>• [3.1.0] New messages: HCP002E, HCP003E, HCP004E, HCP009E, HCP6704E.</li> </ul>
LOCATE VDEV	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New messages: HCP020E, HCP026E.</li> </ul>
LOCATE VSMBK	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New message: HCP026E.</li> </ul>
LOCATEVM	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New operand: ASCII.</li> <li>• [3.1.0] New messages: HCP002E, HCP003E, HCP004E, HCP009E, HCP6704E.</li> </ul>
LOCK	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New response variations to support 64-bit architecture.</li> </ul>

Table 5. Changes to CP Commands (continued)

Command	Changes
LOGON / LOGIN	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] The fields of an IP address included in the response no longer contain leading zeros.</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] New message: HCP2808E.</li> <li>• [3.1.0] Can specify larger amount of storage on the STORAGE operand.</li> <li>• [3.1.0] New message: HCP093E.</li> </ul>
MODIFY CHPID / PATH	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] New return codes for message HCP6806E.</li> </ul>
MODIFY CU / CNTLUNIT	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] New operands: TYPE CF.</li> <li>• [2.4.0] Range of permitted values for CU_LOGICAL_ADDRESS has been increased.</li> <li>• [3.1.0] New operand: MANAGED_PATHS <i>n</i>.</li> </ul>
MODIFY DEVICE / IODEVICE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New format for message HCP6818E.</li> </ul>
QUERY (in general)	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• See QUERY VIRTUAL CONSOLE command below.</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operands: PAV, PSWTRANS, UNDERSCORE.</li> <li>• See QUERY commands below.</li> </ul>
QUERY CACHEFW	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] Response indicates if the cache fast write function is suspended for the subsystem.</li> </ul>
QUERY CHPID	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] New operand: TYPE.</li> <li>• [2.3.0] New responses if TYPE is specified.</li> <li>• [2.4.0] New responses for the TYPE operand to indicate OSA Express, OSA Direct-Express, cluster-bus-sender, and FICON channels.</li> <li>• [3.1.0] New responses for the TYPE operand to indicate coupling-facility-sender, cluster-bus-receiver, internal-coupling-sender, internal-coupling-receiver, direct-system-device, emulated-I/O, cluster-bus-peer, coupling-facility-peer, and internal-coupling-peer channels.</li> </ul>
QUERY CPLEVEL	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New operands to specify date format: SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] Response includes 4-digit years for FULLDATE and ISODATE operands.</li> <li>• [2.2.0] If the user's default date format is changed from SHORTDATE to FULLDATE or ISODATE, and a date format option is not specified on the QUERY CPLEVEL command, the output from QUERY CPLEVEL uses the new default date format. This causes the date to be expanded to include the 4-digit year and also causes fields to the right of the date to be shifted.</li> <li>• [3.1.0] The response contains a new field after the service level to identify the CP image that is running: 32-bit or 64-bit.</li> <li>• [3.1.0] In the response, the release level value has changed.</li> </ul>

## Interface Changes – CP

Table 5. Changes to CP Commands (continued)

Command	Changes
QUERY CPLOAD	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Responses have changed because you can no longer load CP from a nucleus, only from a module.</li> </ul>
QUERY CRYPTO	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] New operand: CAMQS.</li> <li>• [2.4.0] Two new responses.</li> </ul>
QUERY DASD	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operand: QUIESCED.</li> <li>• [3.1.0] Response for QUERY DASD DETAILS may contain a new line containing Parallel Access Volumes information.</li> </ul>
QUERY DASDFW	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] Response indicates if the DASD fast write function is suspended for the subsystem.</li> </ul>
QUERY EXITS	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] Additional response information provided for a dynamic CP exit.</li> </ul>
QUERY FRAMES	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Responses changed to support 64-bit architecture.</li> </ul>
QUERY GRAF	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] The fields of an IP address included in the response no longer contain leading zeros.</li> </ul>
QUERY IMG	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New operands to specify date format: SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] Response includes 4-digit years for FULLDATE and ISODATE operands.</li> <li>• [2.2.0] If the user's default date format is changed from SHORTDATE to FULLDATE or ISODATE, and a date format option is not specified on the QUERY IMG command, the output from QUERY IMG uses the new default date format. This causes the date to be expanded to include the 4-digit year and also causes fields to the right of the date to be shifted.</li> </ul>
QUERY LDEVS	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] The fields of an IP address included in the response no longer contain leading zeros.</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] The response may include the IP address for TCP/IP.</li> </ul>
QUERY NAMES	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] The fields of an IP address included in the response no longer contain leading zeros.</li> </ul>

Table 5. Changes to CP Commands (continued)

Command	Changes
QUERY NLS	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New operands to specify date format: SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] Response includes 4-digit years for FULLDATE and ISODATE operands.</li> <li>• [2.2.0] If the user's default date format is changed from SHORTDATE to FULLDATE or ISODATE, and a date format option is not specified on the QUERY NLS command, the output from QUERY NLS uses the new default date format. This causes the date to be expanded to include the 4-digit year and also causes fields to the right of the date to be shifted.</li> </ul>
QUERY NSS	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New operands to specify date format: SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] Response includes 4-digit years for FULLDATE and ISODATE operands.</li> <li>• [2.2.0] If the user's default date format is changed from SHORTDATE to FULLDATE or ISODATE, and a date format option is not specified on the QUERY NSS command, the output from QUERY NSS uses the new default date format. This causes the date to be expanded to include the 4-digit year and also causes fields to the right of the date to be shifted.</li> </ul>
QUERY READER / PRINTER / PUNCH	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New operands to specify date format: SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] Responses include 4-digit years for FULLDATE and ISODATE operands.</li> <li>• [2.2.0] New operands DIST and NODIST, available only with operands FULLDATE and ISODATE, specify whether the distribution code is to be included in the response. The default is NODIST, so the output record fits within an 80-character buffer.</li> <li>• [2.2.0] If the user's default date format is changed from SHORTDATE to FULLDATE or ISODATE, and a date format option is not specified on the QUERY RDR PRT PUN command, the output from QUERY RDR PRT PUN uses the new default date format. This causes the date to be expanded to include the 4-digit year, the NAME and TYPE fields to the right of the date to be shifted, and the distribution code to be omitted (by default).</li> </ul>
QUERY SPACES	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Response may contain new values to indicate larger address space.</li> </ul>
QUERY STORAGE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Response may contain new values to indicate greater storage.</li> </ul>
QUERY TIME	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New operands to specify date format: SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] Response includes 4-digit years for FULLDATE and ISODATE operands.</li> <li>• [2.2.0] If the user's default date format is changed from SHORTDATE to FULLDATE or ISODATE, and a date format option is not specified on the QUERY TIME command, the output from QUERY TIME uses the new default date format. This causes the date to be expanded to include the 4-digit year.</li> </ul>



## Interface Changes – CP

Table 5. Changes to CP Commands (continued)

Command	Changes
QUERY TOKEN	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New message: HCP6815E.</li> </ul>
QUERY TRACE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Responses may contain new values and expanded address field for 64-bit.</li> </ul>
QUERY TRFILES	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New operands to specify date format: SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] Response includes 4-digit years for FULLDATE and ISODATE operands.</li> <li>• [2.2.0] If the user's default date format is changed from SHORTDATE to FULLDATE or ISODATE, and a date format option is not specified on the QUERY TRFILES command, the output from QUERY TRFILES uses the new default date format. This causes the date to be expanded to include the 4-digit year and also causes fields to the right of the date to be shifted.</li> </ul>
QUERY UCR	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New operands to specify date format: SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] Response includes 4-digit years for FULLDATE and ISODATE operands.</li> <li>• [2.2.0] If the user's default date format is changed from SHORTDATE to FULLDATE or ISODATE, and a date format option is not specified on the QUERY UCR command, the output from QUERY UCR uses the new default date format. This causes the date to be expanded to include the 4-digit year and also causes fields to the right of the date to be shifted.</li> </ul>
QUERY USERID	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Response may contain additional fields to indicate IP address.</li> </ul>
QUERY USERS	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] The fields of an IP address included in the response no longer contain leading zeros.</li> </ul>
QUERY VIRTUAL ALL	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] Supports virtual message processors.</li> </ul>
QUERY VIRTUAL CONSOLE	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] Response may include a new line containing TCP/IP information.</li> </ul>
QUERY VIRTUAL CRYPTO	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] New response.</li> </ul>
QUERY VIRTUAL CTCA	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Response contains new subclass field.</li> </ul>
QUERY (Virtual Device)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] Response indicates REAL-MPLF or SIMULATED-MPLF if enabled for the device.</li> <li>• [2.3.0] Supports virtual message devices. The device type MSGD appears in the response.</li> </ul>
QUERY VIRTUAL GRAF	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] The fields of an IP address included in the response no longer contain leading zeros.</li> </ul>



Table 5. Changes to CP Commands (continued)

Command	Changes
QUERY VIRTUAL OSA	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.4.0] New lines in the response for OSA devices that use the Queued-Direct-I/O (QDIO) Facility.</li> </ul>
QUERY VIRTUAL STORAGE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New response to support the new CONFIGURATION operand on the DEFINE STORAGE command.</li> </ul>
SEND	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Class C user can send input to any virtual machine.</li> </ul>
SET (in general)	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New operands: PSWTRANS, UNDERSCORE.</li> <li>• See SET commands below.</li> </ul>
SET CPTRACE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.4.0] New trace category and trace codes for QDIO instructions.</li> <li>• [3.1.0] New trace codes.</li> </ul>
SET CRYPTO	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.4.0] New operands: MODIFY, ON, OFF.</li> <li>• [2.4.0] Changed responses.</li> <li>• [2.4.0] Changed messages: HCP1706I, HCP1709E, HCP1710E, HCP1711I.</li> <li>• [2.4.0] New messages: HCP1714E, HCP1715E.</li> </ul>
SET DUMP	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New operand: XF.</li> <li>• [3.1.0] New message: HCP1851E.</li> </ul>
SET MACHINE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New message: HCP1016E.</li> </ul>
SET MDCACHE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Supports defining a minidisk cache larger than 2GB for z/Architecture virtual machines.</li> </ul>
SET RDEVICE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New messages: HCP6580E, HCP6862E, HCP6870E.</li> </ul>
SHUTDOWN	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New operand: WAIT.</li> <li>• [3.1.0] If REIPL is specified, an automatic warmstart is done from a CP module, not a nucleus.</li> </ul>
SPTAPE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.2.0] First two digits of the 4-digit year are included in hexadecimal format in the SFBLOK dumped to tape.</li> </ul>
STORE (in general)	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Message HCP6150E may specify z/Architecture mode.</li> <li>• [3.1.0] New operands: PSWA, PSWG.</li> <li>• See STORE commands below.</li> </ul>
STORE ESA/XC Storage	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New operands for indirect addressing and storing data in ASCII format.</li> <li>• [3.1.0] Supports 64-bit addressing.</li> </ul>

## Interface Changes – CP

Table 5. Changes to CP Commands (continued)

Command	Changes
STORE Guest Storage	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operands for indirect addressing and storing data in ASCII format.</li> <li>• [3.1.0] Supports 64-bit addressing.</li> <li>• [3.1.0] New z/Architecture version of this command.</li> </ul>
STORE Host Storage	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operands for indirect addressing and storing data in ASCII format.</li> <li>• [3.1.0] Supports 64-bit addressing.</li> </ul>
STORE PSW	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] For a z/Architecture guest, translates an ESA/390 format PSW into a z/Architecture format PSW (if the SET PSWTRANS ALL or SET PSWTRANS STORE command was previously issued).</li> </ul>
STORE (Registers)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] New operands: FPC <i>hexword</i>.</li> <li>• [2.4.0] Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor.</li> <li>• [2.4.0] New messages: HCP6153E, HCP6154E.</li> <li>• [3.1.0] New operands: GGreg1, GHreg1, XGreg1, XHreg1.</li> <li>• [3.1.0] Message HCP6150E may specify z/Architecture mode.</li> </ul>
STORE STATUS	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] Stores the address of the extended save area at address 212 (X'D4'). This save area contains floating-point registers 0-15 and the floating-point control register.</li> <li>• [3.1.0] Stores virtual machine data for a z/Architecture virtual machine.</li> </ul>
SYSTEM	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Stores virtual machine data for a z/Architecture virtual machine.</li> </ul>
TRACE (in general)	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New message: HCP1038E.</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] New command responses.</li> <li>• [2.4.0] New response for floating-point register information.</li> <li>• [3.1.0] Message HCP6150E may specify z/Architecture mode.</li> <li>• [3.1.0] New operand: GG.</li> <li>• [3.1.0] New option: ASCE.</li> <li>• [3.1.0] New response formats for 64-bit.</li> <li>• See TRACE commands below.</li> </ul>
TRACE <i>mnemonic1</i>	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] New mnemonic: BSA.</li> <li>• [3.1.0] New mnemonics: EPSW, LCTLG, LPSWE, STCTG, STFL, STSI.</li> </ul>
TRACE TABLE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New response formats for 64-bit.</li> </ul>
TRSOURCE ID	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New values can be specified on the DL operand for 64-bit registers or indirect addressing.</li> </ul>

Table 5. Changes to CP Commands (continued)

Command	Changes
UNLOCK	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"> <li>• [3.1.0] RIO370 operand no longer supported.</li> </ul>
VARY (Real Device)	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New message: HCP6861I.</li> </ul>
VMDUMP	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Can dump larger storage areas and discontinuous storage areas.</li> </ul>
XAUTOLOG	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Can specify larger amount of storage on the STORAGE operand.</li> </ul>

## CP Utilities

Table 6 lists the utility programs that have changed. For additional information, see the *z/VM: CP Command and Utility Reference*.

Table 6. Changes to Utility Programs

Utility	Changes
DIRECTXA	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"> <li>• [2.3.0] If a USER statement within the directory specifies a logical line edit symbol that is not valid (a letter A-Z, number 0-9, or bytes X'OE' (shift out) or X'OF' (shift in)), DIRECTXA issues message HCP786I, uses the system default line edit symbol, and continues processing. If no error prevents the directory from being written, DIRECTXA returns to CMS with RC=9.</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operands: &amp;SYSRES <i>valid</i>.</li> <li>• [3.1.0] New message: HCP493E.</li> <li>• [3.1.0] In the response, the release level value has changed.</li> </ul>
SALIPL	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"> <li>• [2.2.0] Message HCP039E has been deleted and replaced by new message HCP394E with the same text.</li> <li>• [2.2.0] Date field on the file list panel displays the year with 4 digits.</li> </ul>

## DIAGNOSE Codes

Table 7 lists the DIAGNOSE codes that have changed. For additional information, see *z/VM: CP Programming Services*.

## Interface Changes – CP

Table 7. Changes to DIAGNOSE Codes

Code	Changes
X'00' Storage Extended Identification Code	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] In the program product bit map, bit 13 (X'0004000000000000') indicates whether Year 2000 support is present in CP.</li> <li>• [3.1.0] In the execution environment field of the extended-identification code bit map, bit 1 (formerly reserved) indicates the CP image that is running: 0=32-bit, 1=64-bit.</li> <li>• [3.1.0] Licensed program bit map field of the extended-identification code bit map contains a new value to indicate the new product level.</li> <li>• [3.1.0] Release information field of the extended-identification code bit map contains the new release level.</li> </ul>
X'04' Examine Real Storage	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Supports 64-bit addressing.</li> </ul>
X'14' Input Spool File Manipulation	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] For subcodes X'0004', X'0008', X'0FFE', and X'0FFF', a one-byte century indicator was added to the SFBLOK data area.</li> </ul>
X'68' Virtual Machine Communication Facility	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] New function: SETLIMIT (Subcode X'000C').</li> </ul>
X'7C' Logical Device Support Facility	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] For the INITIATE function, bit 3 of the first byte of Rx+1 indicates that Ry+1 contains the IP address associated with the logical device.</li> </ul>
X'84' Directory Update-in-Place	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] For EDITCHAR operation, letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'OF' (shift in) cannot be specified as logical line edit symbols (line-end, line-delete, character-delete, and escape).</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] Can now replace the user's default date format setting.</li> <li>• [2.2.0] New operation: DATEFMT.</li> <li>• [2.2.0] New return codes due to new function (in hex): 122, 123, 124.</li> </ul>
X'98' Real I/O	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New subfunctions: BLOCK DIAGNOSE, MULTIPLE REQUEST BLOCK.</li> </ul>
X'BC' Open and Query Spool File Characteristics	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] Depending on the specified buffer length, following the SECLABEL field the user's buffer will include the full (4-digit-year) date and the ISO date.</li> </ul>
X'D8' Read Spool File Blocks	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] For subcode X'0000', a one-byte century indicator was added to the SFBLOK data area.</li> </ul>

Table 7. Changes to DIAGNOSE Codes (continued)

Code	Changes
X'210' Retrieve Device Information	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>[3.1.0] A byte code X'0C' of the VRDCBLOK control block has been added so that application programs can find out the underlying real hardware associated with tape devices operating in emulation mode.</li> <li>[3.1.0] New messages: HCP2327I, HCP2328I, HCP2340I, HCP2341I, HCP2342I, HCP2343I, HCP2344I, HCP2345I, HCP2346I, HCP2347I, HCP2348I, HCP2349I, HCP2350I, HCP2351I, HCP2352I, HCP2353I, HCP2354I, HCP2355I, HCP2356I, HCP2357I, HCP2358I, HCP2359I, HCP2360I, HCP2361I, HCP2362I, HCP6365I.</li> </ul>
X'270' Pseudo Timer Extended	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>[2.4.0] Output has been expanded to include the version of DIAGNOSE code X'270', the user's default date format, and the system default date format.</li> </ul>

## CP Macros

This section identifies the CP macros that have changed. It contains the following subsections:

- IUCV Functions
- “APPCVM Macro Functions”
- “VM Data Spaces CP Macros”
- “Other CP Macros”

### IUCV Functions

There are no changes to IUCV functions.

### APPCVM Macro Functions

There are no changes to APPCVM macro functions.

### VM Data Spaces CP Macros

There are no changes to VM data spaces CP macros.

### Other CP Macros

Table 8 lists the other CP macros that have changed.

Table 8. Changes to Other CP Macros

Macro	Changes
HCPTKDEF	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>[2.4.0] New conversion type: INSTRUCT. For additional information, see <i>z/VM: CP Exit Customization</i>.</li> </ul>

## CP Messages

The following CP messages do not exist in z/VM 3.1.0:

HCP353W	HCP1602W	HCP8039S
HCP354W	HCP1605W	HCP8167E
HCP543A	HCP1606W	HCP8611T
HCP972W	HCP1607W	HCP9010W
HCP1161I	HCP1609W	HCP9016W
HCP1365E	HCP6775E	

## Interface Changes – CP

The following CP messages have changed. For additional information, see *z/VM: System Messages and Codes*.

**Note:** If a message has changed in more than one release, only the latest release is indicated.

[3.1.0] HCP047E	[2.4.0] HCP1706I	[3.1.0] HCP2801E
[3.1.0] HCP296E	[2.4.0] HCP1709E	[3.1.0] HCP2805E
[3.1.0] HCP319E	[2.4.0] HCP1710E	[3.1.0] HCP2806E
[3.1.0] HCP332E	[2.4.0] HCP1711I	[3.1.0] HCP2819E
[3.1.0] HCP513I	[2.4.0] HCP2234E	[3.1.0] HCP5769E
[2.3.0] HCP580I	[2.2.0] HCP2252E	[2.4.0] HCP6111I
[2.4.0] HCP1003E	[3.1.0] HCP2601E	[2.4.0] HCP6706E
[2.3.0] HCP1016E	[2.3.0] HCP2768E	[2.4.0] HCP6789E
[2.3.0] HCP1512E	[3.1.0] HCP2800E	

---

## CMS Changes

This section identifies the changes to CMS external interfaces. It contains the following subsections:

- General CMS Commands
- “CMS File Pool Administration and Operator Commands” on page 49
- “OpenExtensions (OPENVM) Commands” on page 49
- “XEDIT Subcommands” on page 50
- “CMS Pipelines” on page 51
- “CMS Routines” on page 52
- “CMS Macros” on page 58
- “CMS Messages” on page 61

### General CMS Commands

Table 9 lists the general CMS commands that have changed. For additional information, see the *z/VM: CMS Command Reference*.

Table 9. Changes to General CMS Commands

Command	Changes
CMSDESK	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"><li>• [2.2.0] Message DMS2302E replaced by new format of DMS622E, same return code.</li></ul> <b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.2.0] Supports three date formats: short date, full date, ISO date.</li><li>• [2.2.0] Display of the contents of the File Manager application has been changed.</li><li>• [2.2.0] Displays a toolbar on the applications.</li><li>• [2.2.0] New and changed menu items for some applications.</li><li>• [3.1.0] See “Graphical User Interface (GUI) Facility Changes [3.1.0]” on page 12.</li></ul>
COPYFILE	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"><li>• [2.2.0] New message: DMS516E.</li></ul>
CREATE DIRECTORY	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.2.0] New message for authorization failure from ESM: DMS1331E.</li></ul>

Table 9. Changes to General CMS Commands (continued)

Command	Changes
CSLLIST	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] When an = or ? is typed as the first character in the “Cmd” area of a line in the CSLLIST display screen, all characters following the = or ? are ignored.</li> </ul>
CSLMAP	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] When an = or ? is typed as the first character in the “Cmd” area of a line in the CSLMAP display screen, all characters following the = or ? are ignored.</li> </ul>
DEFAULTS	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New options supported as parameters for FILELIST and RDRLIST: VMDATE, SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.3.0] New options supported as parameters for NETDATA: VMDATE, SHORTDATE, FULLDATE, ISODATE.</li> </ul>
DIRLIST	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] When an = or ? is typed as the first character in the “Cmd” area of a line in the DIRLIST display screen, all characters following the = or ? are ignored.</li> </ul>
ERASE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New message for authorization failure from ESM: DMS1332E.</li> </ul>
FILEDEF	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New option: LIBSRV.</li> <li>• [2.4.0] Allows LRECL definitions up to 65535 bytes for OS variable spanned records (under XLRI processing) and non-OS CMS files.</li> <li>• [2.4.0] Allows BLKSIZE definitions up to 65535 bytes for non-OS CMS files.</li> </ul>
FILELIST	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] When FILELIST is specified with the SHARE option, if a pre-VM/ESA 2.2.0 profile (PROFFSHR XEDIT) resides on a disk accessed ahead of the S-disk, sorts by date or size will not work. IBM recommends that you recreate all non-system FILELIST profiles. See Appendix A of the <i>z/VM: CMS Command Reference</i>.</li> <li>• [2.2.0] If you file the file created by FILELIST, that file might contain new and changed fields (on the far right).</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] BEFORE <i>date</i> and AFTER <i>date</i> options support 4-digit years.</li> <li>• [2.2.0] Screens and responses support 4-digit years.</li> <li>• [2.2.0] If a date format option is not specified on the FILELIST command, the CMS DEFAULTS date format setting for FILELIST will be used.</li> <li>• [2.3.0] When an = or ? is typed as the first character in the “Cmd” area of a line in the FILELIST display screen, all characters following the = or ? are ignored.</li> </ul>
GLOBAL	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New message for duplicate library name in input list: DMS045W. The duplicates are ignored.</li> </ul>

## Interface Changes – CMS

Table 9. Changes to General CMS Commands (continued)

Command	Changes
IDENTIFY	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] New options to specify date format: SHORTDATE, FULLDATE, ISODATE.</li><li>• [2.2.0] Responses support 4-digit years.</li><li>• [2.2.0] If the user's default date format is changed from SHORTDATE to FULLDATE or ISODATE, and a date format option is not specified on the IDENTIFY command, the output from IDENTIFY uses the new default date format. This causes the date to be expanded to include the 4-digit year and also causes fields to the right of the date to be shifted.</li><li>• [2.3.0] New option for TCP/IP mail integration: TCPIP.</li></ul>
LISTDS	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.3.0] New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE.</li><li>• [2.3.0] Responses support 4-digit years.</li><li>• [2.3.0] If the user's default date format is changed to FULLDATE or ISODATE, and the SHORTDATE option is not specified on the LISTDS command, any output from LISTDS with dates uses the new default date format. This causes the dates to be expanded to include the 4-digit year and also causes fields to be shifted.</li></ul>
LISTFILE	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"><li>• [2.4.0] Message DMS550E changed to DMS765E.</li></ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] New options to specify date format: SHORTDATE, FULLDATE, ISODATE.</li><li>• [2.2.0] BEFORE <i>date</i> and AFTER <i>date</i> options support 4-digit years.</li><li>• [2.2.0] Responses support 4-digit years.</li><li>• [2.2.0] If the user's default date format is changed from SHORTDATE to FULLDATE or ISODATE, and a date format option is not specified on the LISTFILE command, any output from LISTFILE with dates uses the new default date format. This causes the dates to be expanded to include the 4-digit year and also causes fields to be shifted.</li></ul>
LKED	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.4.0] Defaults for <i>value1</i> and <i>value2</i> on the SIZE option have been increased to 400K and 100K, respectively.</li></ul>
LOAD	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] New message for insufficient storage above 16MB: DMS891W.</li></ul>
LOADMOD	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] Changed message (new text possible): DMS639E.</li></ul>
MACLIST	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.3.0] When an = or ? is typed as the first character in the "Cmd" area of a line in the MACLIST display screen, all characters following the = or ? are ignored.</li></ul>



Table 9. Changes to General CMS Commands (continued)

Command	Changes
MOVEFILE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] Can process OS variable spanned records (under XLRI processing) and non-OS CMS files with record lengths up to 65535 bytes.</li> <li>• [2.4.0] Adjusts output file sizes for compatibility between CMS and OS.</li> <li>• [2.4.0] Allows greater FILEDEF default flexibility for file attributes (RECFM, LRECL, BLKSIZE).</li> <li>• [2.4.0] If default size values are used, fixes record truncation problems when moving data files from fixed to variable format.</li> <li>• [2.4.0] New message: DMS1116E.</li> </ul>
NETDATA	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.3.0] Supports 4-digit-year date formats for entries in the <i>userid</i> NETLOG file.</li> <li>• [2.3.0] Responses support 4-digit years.</li> <li>• [2.3.0] If the user's default date format is changed to FULLDATE or ISODATE, and the SHORTDATE option is not specified on the NETDATA command, any output from NETDATA with dates uses the new default date format. This causes the dates to be expanded to include the 4-digit year and also causes fields to be shifted.</li> </ul>
NOTE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] In the Date field of the note header, the year is now displayed with four digits.</li> <li>• [2.3.0] TCP/IP domain names accepted as user IDs or as the resolution of nicknames.</li> <li>• [2.3.0] Supports 4-digit-year date formats for entries in the <i>userid</i> NETLOG file.</li> </ul>
NUCXLOAD	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] Changed message (new text possible): DMS639E.</li> </ul>
PEEK	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] TCP/IP origin domain name address used when available and shown on PEEK message line for origin within current space and formatting limitations.</li> </ul>
PIPE	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] All CMS Pipelines messages have a new prefix, and many messages have new numbers and text. See “Appendix A. CMS Pipelines Message Cross-Reference [2.3.0]” on page 99.</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• See “CMS Pipelines Stages, Subcommands, and Macros” on page 51.</li> </ul>
QUERY (in general)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operand: TAPENEVR.</li> <li>• See QUERY commands below.</li> </ul>
QUERY CMSLEVEL	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] In the response, the CMS level value has changed.</li> </ul>
QUERY CMSREL	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] In the response, the release level value has changed.</li> </ul>

## Interface Changes – CMS

Table 9. Changes to General CMS Commands (continued)

Command	Changes
QUERY FILEDEF	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] New operand, ATTRIBUT, and its response.</li> </ul>
RDRLIST	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] If you file the file created by RDRLIST, that file might contain new and changed fields (on the far right).</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] Screen supports 4-digit years.</li> <li>• [2.2.0] If a date format option is not specified on the RDRLIST command, the CMS DEFAULTS date format setting for RDRLIST will be used.</li> <li>• [2.3.0] When an = or ? is typed as the first character in the “Cmd” area of a line in the RDRLIST display screen, all characters following the = or ? are ignored.</li> <li>• [2.3.0] TCP/IP origin domain name address used when available and shown on the RDRLIST panel origin area within current space and formatting limitations.</li> </ul>
RECEIVE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] Supports 4-digit-year date formats for entries in the <i>userid</i> NETLOG file.</li> </ul>
SENDFILE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] TCP/IP domain names accepted as user IDs or as the resolution of nicknames.</li> <li>• [2.3.0] New options to specify the transmission method: SMTP, MIME, UFTSYNC, UFTASYNC.</li> <li>• [2.3.0] Supports 4-digit-year date formats for entries in the <i>userid</i> NETLOG file.</li> </ul>
SET (in general)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New operand: TAPENEVR.</li> </ul>
TAPE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] If the tape is under the control of a Tape Library Dataserver machine, and the DFSMS/VM Removable Media Services (RMS) FSMPPSI CSLLIB is available to CMS, the RUN (rewind and unload) function of TAPE calls the RMS FSMRMDMT (Demount) CSL routine to have the Dataserver unmount the tape.</li> <li>• [2.2.0] Added message for Tape Library Dataserver support: DMS2147W.</li> </ul>
TELL	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] Accepts a TCP/IP domain name as part of the destination information.</li> </ul>
VMFPLC2	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] If the tape is under the control of a Tape Library Dataserver machine, and the DFSMS/VM Removable Media Services (RMS) FSMPPSI CSLLIB is available to CMS, the RUN (rewind and unload) function of VMFPLC2 calls the RMS FSMRMDMT (Demount) CSL routine to have the Dataserver unmount the tape.</li> <li>• [2.2.0] Added message for Tape Library Dataserver support: DMS2147W.</li> </ul>

Table 9. Changes to General CMS Commands (continued)

Command	Changes
VMLINK	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Previously, parameters passed to exits were automatically uppercased before the exit was called. Now they remain in the case entered by the user.</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] When an = or ? is typed as the first character in the “Cmd” area of a line in the VMLINK display screen, all characters following the = or ? are ignored.</li> <li>• [3.1.0] New variables: .EX, .PX.</li> <li>• [3.1.0] Error message returned for each failing INVOKE command.</li> <li>• [3.1.0] Message DMS2062I includes the name of the nickname being used when the QUERY option is processed.</li> <li>• [3.1.0] Additional changes to operation and documentation. See “VMLINK Improvements [3.1.0]” on page 12.</li> </ul>

## CMS File Pool Administration and Operator Commands

Table 10 lists the CMS file pool administration and operator commands that have changed. For additional information, see *z/VM: CMS File Pool Planning, Administration, and Operation*.

Table 10. Changes to CMS File Pool Administration and Operator Commands

Command	Changes
AUDIT	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New operands: <i>fn ft</i>, REPLACE.</li> <li>• [2.2.0] Added messages: DMS024E, DMS1258E, DMS3253I, DMS3254E, DMS3255E.</li> <li>• [2.2.0] Changed message: DMS3470W (new text possible).</li> </ul>
DELETE USER	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New options: DELAUTH   KEEPAUTH.</li> <li>• [2.2.0] Added message: DMS2023E.</li> </ul>
FILEPOOL RELOAD	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New message: DMS3455I.</li> </ul>
FILEPOOL UNLOAD	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New message: DMS3455I.</li> </ul>

## OpenExtensions (OPENVM) Commands

Table 11 lists the OpenExtensions (OPENVM) commands that have changed. For additional information, see the *z/VM: OpenExtensions Command Reference*.

## Interface Changes – CMS

Table 11. Changes to OpenExtensions (OPENVM) Commands

Command	Changes
OPENVM DEBUG	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New operands: ALL, NOALL, DUMP, NODUMP, FILEIO, NOFILEIO, FLOW, NOFLOW, MOUNT, NOMOUNT, NFSREQUEST, NONFSREQUEST, OTHER, NOOTHER, RPCBUFFERS, NORPCBUFFERS, RPCLIBRARY, NORPCLIBRARY, WRAPSIZE 500, WRAPSIZE <i>n</i>, FORMAT.</li> <li>• [3.1.0] Supports tracing NFS and BFS Client events.</li> </ul>
OPENVM MOUNT	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Supports new local NFS options.</li> </ul>
OPENVM OWNER	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New operands: <i>gid</i>, <i>uid</i>.</li> </ul>
OPENVM QUERY MOUNT	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New options: NODETAILS, DETAILS.</li> <li>• [3.1.0] Supports mounted NFS file systems.</li> </ul>
OPENVM RUN	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"> <li>• [2.2.0] Changed message (new text possible): DMS639E.</li> </ul>

## XEDIT Subcommands

Table 12 lists the XEDIT subcommands that have changed. For additional information, see the *z/VM: XEDIT Command and Macro Reference*.

Table 12. Changes to XEDIT Subcommands

Subcommand	Changes
FILE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Updated return codes: 32, 55.</li> <li>• [3.1.0] New messages: 1019, 1020.</li> <li>• [3.1.0] Supports mounted NFS file systems.</li> </ul>
GET	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Added message 1019 and 1020; updated return code 32 and 55.</li> <li>• [3.1.0] Supports mounted NFS file systems.</li> </ul>
LOAD	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Added message 1019 and 1020; updated return code 32 and 55.</li> <li>• [3.1.0] Supports mounted NFS file systems.</li> </ul>
PUT	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Added message 1019 and 1020; updated return code 32 and 55.</li> <li>• [3.1.0] Supports mounted NFS file systems.</li> </ul>
PUTD	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Added message 1019 and 1020; updated return code 32 and 55.</li> <li>• [3.1.0] Supports mounted NFS file systems.</li> </ul>
SAVE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Added message 1019 and 1020; updated return code 32 and 55.</li> <li>• [3.1.0] Supports mounted NFS file systems.</li> </ul>
XEDIT	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Added message 1019 and 1020; updated return code 32 and 55.</li> <li>• [3.1.0] Supports mounted NFS file systems.</li> </ul>

## CMS Pipelines

This section identifies the CMS Pipelines functions that have changed. It contains the following subsections:

- CMS Pipelines Stages, Subcommands, and Macros
- “CMS Pipelines Messages”

### CMS Pipelines Stages, Subcommands, and Macros

Table 13 lists the CMS Pipelines stages, subcommands, and macros that have changed. For additional information, see the *z/VM: CMS Pipelines Reference*.

#### [2.3.0]

The code bases for CMS Pipelines and CMS/TSO Pipelines have been merged. Some new function exists as a result of the code merge. See “Pipelines Code Bases Merged [2.3.0]” on page 20.

Table 13. Changes to CMS Pipelines Stages, Subcommands, and Macros

Function	Changes
BFSQUERY stage	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Output record contains the new levels of CP and CMS.</li> </ul>
DATECONVERT stage	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.4.0] Provides timestamp output.</li> </ul>
PIPEPVR macro	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.4.0] New option: <i>label</i>.</li> </ul>
QUERY stage	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Information returned for QUERY VERSION indicates the new version level.</li> <li>• [3.1.0] Information returned for QUERY LEVEL may indicate a new service level.</li> </ul>
READER stage	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.4.0] New options: HOLD, NOHOLD, KEEP, NOKEEP, PURGE.</li> </ul>

### CMS Pipelines Messages

The following CMS Pipelines messages have changed. For additional information, see *z/VM: System Messages and Codes*.

#### [2.3.0]

All CMS Pipelines messages have been renamed and renumbered from a DMS prefix to an FPL prefix. All of the FPL message numbers are consistent with those from CMS/TSO Pipelines. For a cross-reference between DMS and FPL messages, see “Appendix A. CMS Pipelines Message Cross-Reference [2.3.0]” on page 99.

**Note:** If a message has changed in more than one release, only the latest release is indicated.

[2.4.0] FPL1170E

[2.4.0] FPL1171W

[2.4.0] FPL1182E

[2.4.0] FPL1183E

[2.4.0] FPL1184E

[2.4.0] FPL1185E

[2.4.0] FPL1186W

## Interface Changes – CMS

### CMS Routines

This section identifies the CMS routines that have changed. It contains the following subsections:

- CMS Callable Services
- “CMS Multitasking Routines” on page 57
- “OpenExtensions Callable Services” on page 57
- “CMS Compatibility-Interface Routines” on page 58

#### CMS Callable Services

Table 14 lists the CMS callable services that have changed. For additional information, see the *z/VM: CMS Callable Services Reference*.

Table 14. Changes to CMS Callable Services

Routine	Changes
DMSCLBLK	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.</li><li>• [2.2.0] <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li><li>• [2.2.0] New reason codes: 90310, 90320, 90330, 90492, 90495.</li></ul>
DMSCLDBK	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.</li><li>• [2.2.0] <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li><li>• [2.2.0] New reason codes: 90320, 90330, 90495.</li></ul>
DMSCLOSE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.</li><li>• [2.2.0] <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li><li>• [2.2.0] New reason codes: 90320, 90330, 90495.</li></ul>
DMSCRDIR	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.</li><li>• [2.2.0] <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li><li>• [2.2.0] New reason code: 90495.</li></ul>
DMSCRFIL	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.</li><li>• [2.2.0] <i>date</i> and <i>create_date</i> parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li><li>• [2.2.0] New reason code: 90495.</li></ul>
DMSCROB	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.</li><li>• [2.2.0] <i>date</i> and <i>create_date</i> parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li><li>• [2.2.0] New reason code: 90495.</li></ul>

Table 14. Changes to CMS Callable Services (continued)

Routine	Changes
DMSDEUSR	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New parameters: DELAUTH, KEEPAUTH, and <i>length4</i>.</li> <li>• [2.2.0] New reason code: 98700.</li> </ul>
DMSENUSR	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE, <i>length7</i>.</li> <li>• [2.2.0] <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li> <li>• [2.2.0] New reason codes: 90310, 90330, 90495.</li> </ul>
DMSERP	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New information names for Year 2000 support (FILE_DATE_CENTURY, ACT_FILE_DATE_CENTRY (note that U is omitted), and YEAR2000_SUPPORT).</li> </ul>
DMSEXIDI	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] When this routine is called from a REXX program, the date field is returned as 10 characters. Previously, it was returned as 8 characters.</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] <i>last_change_date</i> and <i>create_date</i> parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li> <li>• [2.2.0] New reason codes: 90320, 90330.</li> </ul>
DMSEXIFI	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] When this routine is called from a REXX program, the date field is returned as 10 characters. Previously, it was returned as 8 characters.</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] <i>date</i>, <i>dateref</i>, <i>create_date</i>, and <i>last_change_date</i> parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li> <li>• [2.2.0] New reason codes: 90320, 90330.</li> </ul>

## Interface Changes – CMS

Table 14. Changes to CMS Callable Services (continued)

Routine	Changes																																												
DMSEXIST	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>[2.2.0] Offsets have changed in the FILE data record (and the record length has increased to 436 bytes):</li> </ul> <table border="1"> <thead> <tr> <th>OFFSET</th> <th>Field Name — Change Description</th> </tr> </thead> <tbody> <tr> <td>339 (X'153')</td> <td><i>dec_date_ext</i> — Previously reserved</td> </tr> <tr> <td>343 (X'157')</td> <td><i>date_ext</i> — Added</td> </tr> <tr> <td>353 (X'161')</td> <td><i>iso_date_ext</i> — Added</td> </tr> <tr> <td>363 (X'16B')</td> <td><i>dec_dateref_ext</i> — Added</td> </tr> <tr> <td>367 (X'16F')</td> <td><i>dateref_ext</i> — Added</td> </tr> <tr> <td>377 (X'179')</td> <td><i>iso_dateref_ext</i> — Added</td> </tr> <tr> <td>387 (X'183')</td> <td><i>dec_cr_date_ext</i> — Added</td> </tr> <tr> <td>391 (X'187')</td> <td><i>cr_date_ext</i> — Added</td> </tr> <tr> <td>401 (X'191')</td> <td><i>iso_cr_date_ext</i> — Added</td> </tr> <tr> <td>411 (X'19B')</td> <td><i>dec_last_change_date_ext</i> — Added</td> </tr> <tr> <td>415 (X'19F')</td> <td><i>last_change_date_ext</i> — Added</td> </tr> <tr> <td>425 (X'1A9')</td> <td><i>iso_last_change_date_ext</i> — Added</td> </tr> <tr> <td>435 (X'1B3')</td> <td>Reserved — Added</td> </tr> </tbody> </table> <ul style="list-style-type: none"> <li>[2.2.0] Offsets have changed in the DIRECTORY data record (and the record length has increased to 308 bytes):</li> </ul> <table border="1"> <thead> <tr> <th>OFFSET</th> <th>Field Name — Change Description</th> </tr> </thead> <tbody> <tr> <td>254 (X'FE')</td> <td><i>dec_last_change_date_ext</i> — Previously reserved</td> </tr> <tr> <td>258 (X'102')</td> <td><i>last_change_date_ext</i> — Added</td> </tr> <tr> <td>268 (X'10C')</td> <td><i>iso_last_change_date_ext</i> — Added</td> </tr> <tr> <td>278 (X'116')</td> <td><i>dec_cr_date_ext</i> — Added</td> </tr> <tr> <td>282 (X'11A')</td> <td><i>cr_date_ext</i> — Added</td> </tr> <tr> <td>292 (X'124')</td> <td><i>iso_cr_date_ext</i> — Added</td> </tr> <tr> <td>302 (X'12E')</td> <td>Reserved — Added</td> </tr> </tbody> </table>	OFFSET	Field Name — Change Description	339 (X'153')	<i>dec_date_ext</i> — Previously reserved	343 (X'157')	<i>date_ext</i> — Added	353 (X'161')	<i>iso_date_ext</i> — Added	363 (X'16B')	<i>dec_dateref_ext</i> — Added	367 (X'16F')	<i>dateref_ext</i> — Added	377 (X'179')	<i>iso_dateref_ext</i> — Added	387 (X'183')	<i>dec_cr_date_ext</i> — Added	391 (X'187')	<i>cr_date_ext</i> — Added	401 (X'191')	<i>iso_cr_date_ext</i> — Added	411 (X'19B')	<i>dec_last_change_date_ext</i> — Added	415 (X'19F')	<i>last_change_date_ext</i> — Added	425 (X'1A9')	<i>iso_last_change_date_ext</i> — Added	435 (X'1B3')	Reserved — Added	OFFSET	Field Name — Change Description	254 (X'FE')	<i>dec_last_change_date_ext</i> — Previously reserved	258 (X'102')	<i>last_change_date_ext</i> — Added	268 (X'10C')	<i>iso_last_change_date_ext</i> — Added	278 (X'116')	<i>dec_cr_date_ext</i> — Added	282 (X'11A')	<i>cr_date_ext</i> — Added	292 (X'124')	<i>iso_cr_date_ext</i> — Added	302 (X'12E')	Reserved — Added
OFFSET	Field Name — Change Description																																												
339 (X'153')	<i>dec_date_ext</i> — Previously reserved																																												
343 (X'157')	<i>date_ext</i> — Added																																												
353 (X'161')	<i>iso_date_ext</i> — Added																																												
363 (X'16B')	<i>dec_dateref_ext</i> — Added																																												
367 (X'16F')	<i>dateref_ext</i> — Added																																												
377 (X'179')	<i>iso_dateref_ext</i> — Added																																												
387 (X'183')	<i>dec_cr_date_ext</i> — Added																																												
391 (X'187')	<i>cr_date_ext</i> — Added																																												
401 (X'191')	<i>iso_cr_date_ext</i> — Added																																												
411 (X'19B')	<i>dec_last_change_date_ext</i> — Added																																												
415 (X'19F')	<i>last_change_date_ext</i> — Added																																												
425 (X'1A9')	<i>iso_last_change_date_ext</i> — Added																																												
435 (X'1B3')	Reserved — Added																																												
OFFSET	Field Name — Change Description																																												
254 (X'FE')	<i>dec_last_change_date_ext</i> — Previously reserved																																												
258 (X'102')	<i>last_change_date_ext</i> — Added																																												
268 (X'10C')	<i>iso_last_change_date_ext</i> — Added																																												
278 (X'116')	<i>dec_cr_date_ext</i> — Added																																												
282 (X'11A')	<i>cr_date_ext</i> — Added																																												
292 (X'124')	<i>iso_cr_date_ext</i> — Added																																												
302 (X'12E')	Reserved — Added																																												
DMSGETDA	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>[2.2.0] When this routine is called from a REXX program, the date field is returned as 10 characters. Previously, it was returned as 8 characters.</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>[2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and <i>length2</i>.</li> <li>[2.2.0] <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li> <li>[2.2.0] New reason codes: 90310, 90320, 90330.</li> </ul>																																												



Table 14. Changes to CMS Callable Services (continued)

Routine	Changes																																														
DMSGETDF	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"> <li>[2.2.0] When this routine is called from a REXX program, the date field is returned as 10 characters. Previously, it was returned as 8 characters.</li> </ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>[2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and <i>length2</i>.</li> <li>[2.2.0] <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li> <li>[2.2.0] New reason codes: 90310, 90320, 90330.</li> </ul>																																														
DMSGETDI	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>[2.2.0] Offsets have changed in the FILE data record (and the record length has increased to 112 bytes): <table border="1"> <thead> <tr> <th>OFFSET</th> <th>Field Name — Change Description</th> </tr> </thead> <tbody> <tr> <td>88 (X'58')</td> <td><i>dec_date_ext</i> — Added</td> </tr> <tr> <td>92 (X'5C')</td> <td><i>date_ext</i> — Added</td> </tr> <tr> <td>102 (X'66')</td> <td><i>iso_date_ext</i> — Added</td> </tr> </tbody> </table> </li> <li>[2.2.0] Offsets have changed in the FILEEXT data record (and the record length has increased to 284 bytes): <table border="1"> <thead> <tr> <th>OFFSET</th> <th>Field Name — Change Description</th> </tr> </thead> <tbody> <tr> <td>185 (X'B9')</td> <td><i>dec_date_ext</i> — Previously reserved</td> </tr> <tr> <td>189 (X'BD')</td> <td><i>date_ext</i> — Added</td> </tr> <tr> <td>199 (X'C7')</td> <td><i>iso_date_ext</i> — Added</td> </tr> <tr> <td>209 (X'D1')</td> <td><i>dec_dateref_ext</i> — Added</td> </tr> <tr> <td>213 (X'D5')</td> <td><i>dateref_ext</i> — Added</td> </tr> <tr> <td>223 (X'DF')</td> <td><i>iso_dateref_ext</i> — Added</td> </tr> <tr> <td>233 (X'E9')</td> <td><i>dec_cr_date_ext</i> — Added</td> </tr> <tr> <td>237 (X'ED')</td> <td><i>cr_date_ext</i> — Added</td> </tr> <tr> <td>247 (X'F7')</td> <td><i>iso_cr_date_ext</i> — Added</td> </tr> <tr> <td>257 (X'101')</td> <td><i>dec_last_change_date_ext</i> — Added</td> </tr> <tr> <td>261 (X'105')</td> <td><i>last_change_date_ext</i> — Added</td> </tr> <tr> <td>271 (X'10F')</td> <td><i>iso_last_change_date_ext</i> — Added</td> </tr> <tr> <td>281 (X'119')</td> <td>Reserved — Added</td> </tr> </tbody> </table> </li> <li>[2.2.0] Offsets have changed in the SEARCHALL and SEARCHAUTH data records (and the record length has increased to 252 bytes): <table border="1"> <thead> <tr> <th>OFFSET</th> <th>Field Name/Change Description</th> </tr> </thead> <tbody> <tr> <td>226 (X'E2')</td> <td>Reserved — Added</td> </tr> <tr> <td>228 (X'E4')</td> <td><i>dec_date_ext</i> — Added</td> </tr> <tr> <td>232 (X'E8')</td> <td><i>date_ext</i> — Added</td> </tr> <tr> <td>242 (X'F2')</td> <td><i>iso_date_ext</i> — Added</td> </tr> </tbody> </table> </li> </ul>	OFFSET	Field Name — Change Description	88 (X'58')	<i>dec_date_ext</i> — Added	92 (X'5C')	<i>date_ext</i> — Added	102 (X'66')	<i>iso_date_ext</i> — Added	OFFSET	Field Name — Change Description	185 (X'B9')	<i>dec_date_ext</i> — Previously reserved	189 (X'BD')	<i>date_ext</i> — Added	199 (X'C7')	<i>iso_date_ext</i> — Added	209 (X'D1')	<i>dec_dateref_ext</i> — Added	213 (X'D5')	<i>dateref_ext</i> — Added	223 (X'DF')	<i>iso_dateref_ext</i> — Added	233 (X'E9')	<i>dec_cr_date_ext</i> — Added	237 (X'ED')	<i>cr_date_ext</i> — Added	247 (X'F7')	<i>iso_cr_date_ext</i> — Added	257 (X'101')	<i>dec_last_change_date_ext</i> — Added	261 (X'105')	<i>last_change_date_ext</i> — Added	271 (X'10F')	<i>iso_last_change_date_ext</i> — Added	281 (X'119')	Reserved — Added	OFFSET	Field Name/Change Description	226 (X'E2')	Reserved — Added	228 (X'E4')	<i>dec_date_ext</i> — Added	232 (X'E8')	<i>date_ext</i> — Added	242 (X'F2')	<i>iso_date_ext</i> — Added
OFFSET	Field Name — Change Description																																														
88 (X'58')	<i>dec_date_ext</i> — Added																																														
92 (X'5C')	<i>date_ext</i> — Added																																														
102 (X'66')	<i>iso_date_ext</i> — Added																																														
OFFSET	Field Name — Change Description																																														
185 (X'B9')	<i>dec_date_ext</i> — Previously reserved																																														
189 (X'BD')	<i>date_ext</i> — Added																																														
199 (X'C7')	<i>iso_date_ext</i> — Added																																														
209 (X'D1')	<i>dec_dateref_ext</i> — Added																																														
213 (X'D5')	<i>dateref_ext</i> — Added																																														
223 (X'DF')	<i>iso_dateref_ext</i> — Added																																														
233 (X'E9')	<i>dec_cr_date_ext</i> — Added																																														
237 (X'ED')	<i>cr_date_ext</i> — Added																																														
247 (X'F7')	<i>iso_cr_date_ext</i> — Added																																														
257 (X'101')	<i>dec_last_change_date_ext</i> — Added																																														
261 (X'105')	<i>last_change_date_ext</i> — Added																																														
271 (X'10F')	<i>iso_last_change_date_ext</i> — Added																																														
281 (X'119')	Reserved — Added																																														
OFFSET	Field Name/Change Description																																														
226 (X'E2')	Reserved — Added																																														
228 (X'E4')	<i>dec_date_ext</i> — Added																																														
232 (X'E8')	<i>date_ext</i> — Added																																														
242 (X'F2')	<i>iso_date_ext</i> — Added																																														

## Interface Changes – CMS

Table 14. Changes to CMS Callable Services (continued)

Routine	Changes
DMSGETDS	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] When this routine is called from a REXX program, the date field is returned as 10 characters. Previously, it was returned as 8 characters.</li></ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and <i>length2</i>.</li><li>• [2.2.0] <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li><li>• [2.2.0] New reason codes: 90310, 90320, 90330.</li></ul>
DMSGETDX	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] When this routine is called from a REXX program, the date field is returned as 10 characters. Previously, it was returned as 8 characters.</li></ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and <i>length2</i>.</li><li>• [2.2.0] <i>date</i>, <i>dateref</i>, <i>create_date</i>, and <i>update_date</i> parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li><li>• [2.2.0] New reason codes: 90310, 90320, 90330.</li></ul>
DMSOPBLK	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] When this routine is called from a REXX program, the date field is returned as 10 characters. Previously, it was returned as 8 characters.</li></ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.</li><li>• [2.2.0] <i>date</i>, <i>create_date</i>, and <i>dateref</i> parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li><li>• [2.2.0] New reason code: 90495.</li></ul>
DMSOPDBK	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.</li><li>• [2.2.0] <i>create_date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li><li>• [2.2.0] New reason code: 90495.</li></ul>
DMSOPEN	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.</li><li>• [2.2.0] <i>create_date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li><li>• [2.2.0] New reason code: 90495.</li></ul>
DMSQEFL	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [3.1.0] New values returned for the <i>cp_level</i> and <i>cms_level</i> parameters.</li></ul>
DMSQSFSL	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [3.1.0] New value returned for the <i>server_level</i> parameter.</li></ul>

Table 14. Changes to CMS Callable Services (continued)

Routine	Changes								
DMSRDCAT	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] In the OBJECTCAT record for SFS: <ul style="list-style-type: none"> <li>– New bit settings in FILEFLAGS field to support 4-digit years (century setting for DATE and DATEREF fields).</li> <li>– Reserved CHAR(1) field preceding the LAST_CHANGE_DATE field has changed to: <table border="1"> <thead> <tr> <th>Field Name</th> <th>Field Type/Description</th> </tr> </thead> <tbody> <tr> <td>CHGDATE_CENTURY</td> <td>CHAR(1) Century byte for LAST_CHANGE_DATE</td> </tr> </tbody> </table> </li> <li>– Reserved CHAR(1) field preceding the CREATIONDATE field has changed to: <table border="1"> <thead> <tr> <th>Field Name</th> <th>Field Type/Description</th> </tr> </thead> <tbody> <tr> <td>CREATIONDATE_CENTURY</td> <td>CHAR(1) Century byte for CREATIONDATE</td> </tr> </tbody> </table> </li> </ul> </li> </ul>	Field Name	Field Type/Description	CHGDATE_CENTURY	CHAR(1) Century byte for LAST_CHANGE_DATE	Field Name	Field Type/Description	CREATIONDATE_CENTURY	CHAR(1) Century byte for CREATIONDATE
Field Name	Field Type/Description								
CHGDATE_CENTURY	CHAR(1) Century byte for LAST_CHANGE_DATE								
Field Name	Field Type/Description								
CREATIONDATE_CENTURY	CHAR(1) Century byte for CREATIONDATE								
DMSTRUNC	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.</li> <li>• [2.2.0] <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li> <li>• [2.2.0] New reason code: 90495.</li> </ul>								

### CMS Multitasking Routines

Table 15 lists the CMS multitasking routines that have changed. For additional information, see *z/VM: CMS Application Multitasking*.

Table 15. Changes to CMS Multitasking Routines

Routine	Changes
DateTimeSubtract	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.3.0] Supports new date and time formats.</li> </ul>

### OpenExtensions Callable Services

Table 16 lists the OpenExtensions callable services that have changed. For additional information, see the *z/VM: OpenExtensions Callable Services Reference*.

Table 16. Changes to OpenExtensions Callable Services

Routine	Changes
close (BPX1CLO)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Supports sockets.</li> </ul>
fcntl (BPX1FCT)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Supports sockets.</li> </ul>
mount (BPX1MNT)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Mounts a local or remote file system.</li> </ul>
openvmf (BPX1VM5)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] New function code: VM5_RESOLVE_PATH.</li> </ul>

## Interface Changes – CMS

Table 16. Changes to OpenExtensions Callable Services (continued)

Routine	Changes
read (BPX1RED)	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [3.1.0] Supports sockets.</li></ul>
wrote (BPX1WRT)	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [3.1.0] Supports sockets.</li></ul>

### CMS Compatibility-Interface Routines

Table 17 lists the CMS compatibility-interface routines that have changed. For additional information, see the *z/VM: CMS Application Development Guide for Assembler*.

Table 17. Changes to CMS Compatibility-Interface Routines

Routine	Changes
DMSTVS	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.2.0] New LIBSRV plist parameter.</li><li>• [2.4.0] New message DMS2139I is issued if SENSE data from the tape mount indicates that the mounted tape cartridge may be incorrect for the tape device in use.</li></ul>

## CMS Macros

This section identifies the CMS macros that have changed. It contains the following subsections:

- CMS General-Interface Macros
- “OpenExtensions Macros” on page 59
- “CMS Compatibility-Interface Macros” on page 59
- “CMS OS Simulation Macros and Supervisor Calls” on page 60

### CMS General-Interface Macros

Table 18 lists the CMS general-interface macros that have changed. For additional information, see the *z/VM: CMS Macros and Functions Reference*.

Table 18. Changes to CMS General-Interface Macros

Macro	Changes
CMSLEVEL	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"><li>• [2.2.0] Returned CMS level has been frozen at X'0F' (Level 12) for VM/ESA 2.1.0 and later. Use the new DMSQEFL macro or the DMSQEFL CSL routine instead.</li></ul>
DIRBUFF	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.2.0] FILE record contains the following new fields: DIRFDAXD, DIRFDAXC, DIRFDAXI, DIRFLV13.</li><li>• [2.2.0] FILEEXT record contains the following new fields: DIREDAXD, DIREDAXC, DIREDAXI, DIREDRXD, DIREDRXC, DIREDRXI, DIRECDXD, DIRECDXC, DIRECDXI, DIREDCXD, DIREDCXC, DIREDCXI, DIRELV13.</li><li>• [2.2.0] SEARCHALL and SEARCHAUTH records contain the following new fields: DIRSDAXD, DIRSDAXC, DIRSDAXI, DIRSCEND, DIRSLV13.</li></ul>
DMSQEFL	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [3.1.0] Returns a new value for the CMS level.</li></ul>

Table 18. Changes to CMS General-Interface Macros (continued)

Macro	Changes
EXSBUFF	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] FILE record contains the following new fields: EXSFDAXD, EXSFDAXC, EXSFDAXI, EXSFDRXD, EXSFDRXC, EXSFDRXI, EXSFCDXD, EXSFCDXC, EXSFCDXI, EXSFDCXD, EXSFDCXC, EXSFDCXI, EXSF2000, EXSFLV13.</li> <li>• [2.2.0] In the FILE record, the following field has changed: EXSFRES.</li> <li>• [2.2.0] DIR record contains the following new fields: EXSDDCXD, EXSDDCXC, EXSDDCXI, EXSDCDXD, EXSDCDXC, EXSDCDXI, EXSD2000, EXSDLV13.</li> <li>• In the DIR record, the following field has changed: EXSDRES.</li> </ul>
FSSTATE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] In the FST flag byte, bit 4 indicates the century (first two digits of the year) the file was last written or updated (0=19<math>nn</math>, 1=20<math>nn</math>, where <math>nn</math> is the 2-digit year). Previously, this bit was not used.</li> </ul>
FSTD	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] FSTFLAGS section contains the new FSTCENTRY field, which is a bit that indicates the century (first two digits of the year) the file was last written or updated (0=19<math>nn</math>, 1=20<math>nn</math>, where <math>nn</math> is the 2-digit year).</li> </ul>
TAPECTL	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.2.0] If the tape is under the control of a Tape Library Dataserver machine, and the DFSMS/VM Removable Media Services (RMS) FSMPPSI CSLLIB is available to CMS, the RUN (rewind and unload) function of TAPECTL calls the RMS FSMRMDMT (Demount) CSL routine to have the Dataserver unmount the tape.</li> </ul>

### OpenExtensions Macros

Table 19 lists the OpenExtensions macros that have changed. For additional information, see the *z/VM: OpenExtensions Callable Services Reference*.

Table 19. Changes to OpenExtensions Macros

Macro	Changes
BPXYERNO	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] Contains equates for new return codes and reason codes.</li> </ul>
BPXYOPNF	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [3.1.0] New equates for new flag values for fcntl (BPX1FCT).</li> </ul>
BPXYVM5	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"> <li>• [2.4.0] New equate for new VM5_RESOLVE_PATH function code for openvmf (BPX1VM5).</li> </ul>

### CMS Compatibility-Interface Macros

Table 20 lists the CMS compatibility-interface macros that have changed.

## Interface Changes – CMS

Table 20. Changes to CMS Compatibility-Interface Macros

Macro	Changes
DEFNUC	<p><b>INCOMPATIBLE:</b></p> <ul style="list-style-type: none"><li>• [2.3.0] In the default CMS IPL heading (used when the VERSION= parameter is specified without a value in DEFNUC) the date is presented in ISO format (yyyy-mm-dd).</li></ul> <p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [3.1.0] Default IPL heading has been updated with the new release level.</li></ul> <p>For additional information, see <i>z/VM: Planning and Administration</i>.</p>

### CMS OS Simulation Macros and Supervisor Calls

Table 21 lists the CMS OS Simulation macros that have changed. Table 22 lists the CMS OS simulation supervisor calls that have changed. For additional information, see the *z/VM: CMS Application Development Guide for Assembler*.

Table 21. Changes to CMS OS Simulation Macros

Macro	Changes
CLOSE	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.4.0] Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.</li></ul>
DCB	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.4.0] Can describe and pass both LRI and XLRI conventions for QSAM variable spanned long records (up to 65535 bytes) for subsequent OPEN, CLOSE, GET, or PUT processing. Can also describe non-OS CMS files up to 65535 bytes in length.</li></ul>
GET	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.4.0] Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.</li></ul>
OPEN	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.4.0] Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.</li></ul>
PUT	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.4.0] Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.</li></ul>
TIME	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.2.0] Second half-byte of the date format is a century indicator, where 0 indicates the 1900's, 1 indicates the 2000's, and 2 indicates the 2100's. This corresponds to the MVS implementation of the TIME macro.</li></ul>

Table 22. Changes to CMS OS Simulation Supervisor Calls

SVC	Changes
SVC 19 (OPEN)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.4.0] Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.</li></ul>
SVC 20 (CLOSE)	<p><b>Upwardly compatible:</b></p> <ul style="list-style-type: none"><li>• [2.4.0] Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.</li></ul>

## CMS Messages

### [2.3.0]

All CMS Pipelines messages have been renamed and renumbered from a DMS prefix to an FPL prefix. All of the FPL message numbers are consistent with those from CMS/TSO Pipelines. For a cross-reference between DMS and FPL messages, see “Appendix A. CMS Pipelines Message Cross-Reference [2.3.0]” on page 99.

In the DMSMES system repository, the message texts for message numbers 2571 through 2999 have been replaced with pointers to the corresponding FPL equivalent message numbers in the FPLMES system repository. This may or may not be maintained in any future releases of VM.

The following CMS message does not exist in z/VM 3.1.0:

DMS683E

The following CMS messages have changed. For additional information, see *z/VM: System Messages and Codes*.

**Note:** If a message has changed in more than one release, only the latest release is indicated.

[3.1.0] DMS149E	[2.2.0] DMS1229E	[3.1.0] DMS2119E
[2.2.0] DMS516E	[2.3.0] DMS1433I	[2.3.0] DMS3009R
[2.2.0] DMS622E	[2.3.0] DMS1437I	[2.2.0] DMS3455I
[2.2.0] DMS639E	[2.4.0] DMS2010E	[2.2.0] DMS3470W

---

## REXX/VM Changes

Table 23 lists the REXX/VM external interfaces (instructions, functions, and external functions) that have changed. For additional information, see the *z/VM: REXX/VM Reference*.

Table 23. Changes to REXX/VM External Interfaces

Interface	Changes
CMSFLAG	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>[2.2.0] New value for <i>flag</i>: YEAR2000.</li> </ul>
DATE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>[2.2.0] New parameters allow you to specify a date to be converted to a different format.</li> <li>[2.4.0] New parameters: <i>output_separator_char</i>, <i>input_separator_char</i>.</li> </ul>
DIAG / DIAGRC	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>[2.2.0] New DIAGNOSE code is supported: X'270'.</li> </ul>
PARSE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>[3.1.0] Information obtained by PARSE VERSION may contain new values.</li> </ul>

## Interface Changes – GCS

---

### GCS Changes

Table 24 lists the GCS external interfaces (commands and macros) that have changed. For additional information, see *z/VM: Group Control System*.

Table 24. Changes to GCS Commands and Macros

Interface	Changes
GCSLEVEL macro	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [3.1.0] Contains new equates for new release levels.</li></ul>
GETMAIN macro	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"><li>• [2.2.0] Previously, although LOC=RES was documented as the default, the actual default was LOC=BELOW, and all the requested virtual storage was allocated below 16MB. The actual default has been changed to LOC=RES. If the requester resides above 16MB, virtual storage may be allocated anywhere. If you have any programs that invoke GETMAIN with the default, you must make sure they can accommodate addresses above 16MB, or you must recode them to invoke GETMAIN with LOC=BELOW.</li></ul>
QUERY GCSLEVEL command	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [3.1.0] Response contains new values for the new release.</li></ul>
QUERY MODDATE command	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"><li>• [2.2.0] Full 4-digit year is now returned in the date field of the response instead of a 2-digit year.</li></ul>
TIME macro	<b>Upwardly compatible:</b> <ul style="list-style-type: none"><li>• [2.2.0] Second half-byte of the date format is a century indicator, where 0 indicates the 1900's, 1 indicates the 2000's, and 2 indicates the 2100's. This corresponds to the MVS implementation of the TIME macro.</li></ul>

---

### TSAF Changes

There are no changes to TSAF external interfaces.

---

### AVS Changes

There are no changes to AVS external interfaces.

---

### VMSES/E Changes

This section identifies the changes to VMSES/E external interfaces. It contains the following subsections:

- \$PPF File Names
- “VMSES/E Execs”
- “VMSES/E Messages” on page 64

#### \$PPF File Names

See “\$PPF File Names [3.1.0]” on page 5.

#### VMSES/E Execs

Table 25 lists the VMSES/E execs that have changed. For additional information, see the *z/VM: VMSES/E Introduction and Reference*.



Table 25. Changes to VMSES/E Execs

Exec	Changes
GENCPBLS	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New options: ALTCNTRL, DUAL, NODUAL, LOADLIST, ALTLOADLIST, PREEXIT.</li> </ul>
ITNVTSTR	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.3.0] New operands: KEY, PROD.</li> </ul>
VMFAPPLY	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New option: PREEXIT.</li> </ul>
VMFASM	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New option: PREEXIT.</li> </ul>
VMFBLD	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New options: PREEXIT, ULOG, WILD.</li> </ul>
VMFEXUPD	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New options: PREEXIT.</li> </ul>
VMFHASM	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New options: PREEXIT.</li> </ul>
VMFHLASM	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New options: ALTCNTRL, DUAL, NODUAL, LOADLIST, ALTLOADLIST, PREEXIT.</li> </ul>
VMFINS (in general)	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.3.0] New operands: DISABLE, ENABLE.</li> <li>• See VMFINS commands below.</li> </ul>
VMFINS DELETE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.3.0] New operand: DISABLE.</li> </ul>
VMFINS INSTALL	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.3.0] New operands: DISABLE, ENABLE, NOSETUP, SETUP.</li> </ul>
VMFINS MIGRATE	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.3.0] New operands: DISABLE, ENABLE.</li> </ul>
VMFMRDSK	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New options: PREEXIT.</li> </ul>
VMFNLS	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New options: PREEXIT.</li> </ul>
VMFREC	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New options: PREEXIT.</li> </ul>
VMFREM	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New options: PREEXIT.</li> </ul>
VMFREPL	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New options: LOG, LOGLVL, MOD, PREEXIT, PREFIX, PTF.</li> </ul>
VMFSETUP	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.3.0] New operands: NOCONS, NOPROMPT, PROMPT.</li> </ul>
VMFSIM (in general)	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• See VMFSIM commands below.</li> </ul>
VMFSIM GETLVL	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] New options: MOD, PTF.</li> </ul>

## Interface Changes – VMSES/E

Table 25. Changes to VMSES/E Execs (continued)

Exec	Changes
VMFSUFIN	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.4.0] New operand: ALL.</li> <li>• [2.4.0] New options: CORTAPE, RSUTAPE.</li> </ul>
VMFVIEW	<b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [2.4.0] New operands (all have the same function): REMOVE, VMFREM, \$VMFREM.</li> </ul>

## VMSES/E Messages

The following VMSES/E messages have changed. For additional information, see the *z/VM: System Messages and Codes*.

**Note:** If a message has changed in more than one release, only the latest release is indicated.

[2.3.0] VMF1078I	[2.4.0] VMF2114R	[2.4.0] VMF2206W
[2.4.0] VMF1203E	[2.4.0] VMF2118I	[2.2.0] VMF2225E
[2.4.0] VMF1205E	[2.4.0] VMF2119I	[2.3.0] VMF2507I
[2.4.0] VMF1207I	[2.4.0] VMF2120W	[2.2.0] VMF2509I
[2.4.0] VMF1211I	[2.4.0] VMF2152E	[3.1.0] VMF2733E
[2.2.0] VMF1821E	[3.1.0] VMF2178I	[2.3.0] VMF2760I
[2.4.0] VMF2066E	[2.2.0] VMF2206E	[2.4.0] VMF2767I

## Dump Viewing Facility Changes

Table 26 lists the Dump Viewing Facility functions that have changed. For additional information, see *z/VM: Dump Viewing Facility*.

Table 26. Changes to Dump Viewing Facility Functions

Function	Changes
DUMPSCAN	<b>INCOMPATIBLE:</b> <ul style="list-style-type: none"> <li>• [3.1.0] DUMPSCAN cannot be used for CP dumps. Use the VM Dump Tool.</li> <li>• [3.1.0] The following subcommands are no longer supported: CPEBK, FINDCPE, FINDUSER, FRAMETBL, INSPECT, REAL, RIOBLOK, SELECT, SNAPLIST, TRSAVE, VILOBLOK, VIRT, VMDBK, XTRACE.</li> <li>• [3.1.0] Header record indicates new product name.</li> </ul> <b>Upwardly compatible:</b> <ul style="list-style-type: none"> <li>• [3.1.0] Header record indicates new release level.</li> </ul>

---

## Chapter 4. Migration Tasks

This chapter provides guidance for various tasks and procedures that may apply to your migration. It contains the following major sections:

- “Converting to System Configuration Files from HCPRIO and HCPSYS”
- “Migrating Spool Files and Saved Segments” on page 66
- “Sharing Data between Your Old System and Your New System” on page 72
- “Sharing Data among CMS Users on Multiple Systems” on page 77
- “Migrating Your User Directory” on page 78
- “Migrating Your SFS File Pool Servers” on page 81
- “Establishing Connectivity between Your New and Old Systems” on page 86
- “How to Back Out” on page 88
- “Installing a Backlevel CMS” on page 89
- “Migrating REXX and EXEC 2 Application Programs” on page 98

---

### Converting to System Configuration Files from HCPRIO and HCPSYS

If you have not done so already, you should convert your HCPRIO and HCPSYS files to system configuration files. To convert to using system configuration files, you need to create a SYSTEM CONFIG file that represents your system (as currently defined in HCPSYS and HCPRIO). VM supplies two unsupported sample programs, HCPTRIO and HCPTSYS, that you can use. HCPTRIO and HCPTSYS are REXX programs and can run on any level of VM. The programs read your HCPRIO and HCPSYS files and create files containing statements that you can use in a SYSTEM CONFIG file. HCPTSYS and HCPTRIO can process HCPRIO and HCPSYS files from any previous VM release, so you do not need to upgrade your HCPRIO and HCPSYS files before running the commands.

#### Notes:

1. HCPTSYS and HCPTRIO do not process DMKRIO and DMKSYS files.
2. The SYSTEM CONFIG file created by HCPTSYS and HCPTRIO may not be complete. Because HCPTSYS and HCPTRIO create statements from your existing HCPSYS and HCPRIO statements, they may not create some new statements that are necessary for IPL. For example, HCPTSYS and HCPTRIO do not define the Initial\_Parmdisk\_Access or the Logo\_Config sections in the SYSTEM CONFIG file. You must create these sections. You can use the *z/VM: Planning and Administration* book to help you do this.

In the following example, system configuration statements are created in a file named RIO CONFIG A. The statements are created from the macros contained in the file HCPRIO ASSEMBLE D.

```
hcprio rio config a from hcprio assemble d
```

The HCPTSYS command in the next example creates SYS CONFIG A from HCPSYS ASSEMBLE E.

```
hcpsys sys config a from hcpsys assemble e
```

To use the statements created by HCPTRIO and HCPTSYS, add them to the sample SYSTEM CONFIG file on the PARM disk. Make any other changes you desire or need and re-IPL CP. System configuration statements override the HCPRIO,

## Migration Tasks

HCPSYS, and HCPBOX macros that were used in building the CP nucleus, so there is no need to rebuild CP. All the system configuration statements are documented in *z/VM: Planning and Administration*. See “Appendix B. Sample Utilities for CP Configurability” on page 105 for more about HCPTRIO and HCPTSYS.

If you choose to stick with the old method temporarily, you should convert later. You will need to create a SYSTEM CONFIG file on the PARM disk and re-IPL CP. You can use the sample utility program, HCPDCON, to create the SYSTEM CONFIG statements needed. HCPDCON examines a running system and generates a file of configuration statements. By default, HCPDCON creates a file named SYSTEM CONFIG on the first CMS minidisk accessed in R/W mode. Assuming file mode A is accessed R/W, the following command creates the file SYSTEM CONFIG A:

```
hcpdcon
```

The statements in SYSTEM CONFIG override the HCPRIO, HCPSYS, and HCPBOX macros, so there is no need to rebuild CP. In fact, you can fall back to the old method, if necessary, by renaming the SYSTEM CONFIG file (so that there is none to be found on the PARM disk), and re-IPLing CP. See the *z/VM: Planning and Administration* for a complete description of HCPDCON.

### Difference with Multiple Systems Using Shared Data

When using CP configurability support, you may see DASD attached to systems other than your new system. For example, your installation can have a large set of DASD that is shared by several systems. All the systems' IOCPs could be set up for all the DASD, but each system actually uses only some of the DASD.

Because the new CP configuration senses all the DASD you have defined in your IOCP, you can see DASD not attached to your own system. In previous releases, you used HCPRIO to define exactly which DASD you wanted your system to use. You could see only the DASD defined for your system in HCPRIO.

To have your new system ignore these DASD, vary them offline. You can place the VARY OFFLINE *rdev(s)* command in the PROFILE EXEC file of the system startup user ID that is called during initialization; this is usually the AUTOLOG1 user ID.

### Converting HCPSYS Macros

If you convert to using the SYSTEM CONFIG file, you still need an HCPSYS with a SYSEND macro. See *z/VM: Planning and Administration* for complete descriptions of HCPSYS macros.

### Converting HCPRIO Macros

If you convert to using the SYSTEM CONFIG file, you still need an HCPRIO with an RIOGEN CONS=DYNAMIC macro. See *z/VM: Planning and Administration* for complete descriptions of HCPRIO macros. Note that some devices are no longer supported. See *z/VM: General Information* for information on device support.

---

## Migrating Spool Files and Saved Segments

This section provides you with some options for migrating your spool files, including saved segments, from your old system to your new system.

**Notes:**

1. The IPL process handles migration of spool files during IPL, but as a precaution you may want to consider doing your own spool file migration, especially for important spool files.
2. Use the SPXTAPE command to dump files from your old system and load them on the new system.

The VMFSGMAP EXEC provides a saved segment mapping and planning interface and allows you to define saved segments to VMSES/E. You can then use the VMFBLD EXEC to build saved segments on your system. The *z/VM: Planning and Administration* book provides details on saved segments, including how to use VMFSGMAP to map out and manipulate saved segment layouts and how to use VMFBLD to build saved segments on your system.

Depending on how you plan to migrate to the new system, you can use various methods to migrate your old systems's spool files and saved segments, as shown in Table 27.

Table 27. Methods for Migrating Spool Files

Method	When you can use this method
Using the same warm start and checkpoint areas	<ul style="list-style-type: none"> <li>• When you are ready to cut over completely from your old system to your new system</li> <li>• When you want to migrate all your spool files and saved segments at once</li> </ul>
Using SPXTAPE	<ul style="list-style-type: none"> <li>• When you are ready to cut over completely from your old system to your new system</li> <li>• When you want to use your new system as a test system before cutting over completely</li> <li>• When you want to migrate all your spool files and saved segments at once</li> <li>• When you want to migrate selected spool files or saved segments</li> </ul>
Building saved segments individually	<ul style="list-style-type: none"> <li>• Anytime</li> </ul>

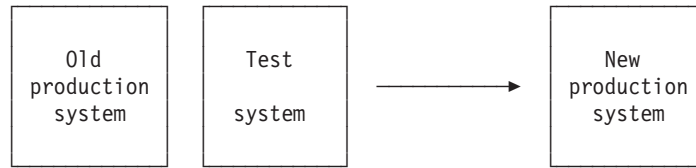
## Using the Same Warm Start and Checkpoint Areas to Migrate All Spool Files at Once

If you are ready to cut over completely from your old system to a new production system, you can set up your new system to use the same warm start and checkpoint areas as your old system. When you IPL your new system with the warm start option, all of the spool files, including your saved segments, are known to your new system. Be sure that no other users are on the system.

**Note:** If you cold start your system, saved segments are saved and known to your system.

The steps below describe the situation where you are currently running the new z/VM release on a test system.

## Migration Tasks



1. Define your new production system using the same warm start area and spool areas you defined in your old production system. Define these areas in the CP\_OWNED statement in SYSTEM CONFIG. List all DASD from your old system first, just as it is defined. Then list any additional volumes directly after.
2. Use SPXTAPE to dump all of your spool files to tape from your old production system. This is for backup purposes.
3. On your old system, purge any saved segments, saved systems, or other spool files that you do not want on your new production system.
4. Use SPXTAPE to dump all of your spool files and system data files to your test system. You have to do this because in the next step you IPL your new system using the same warm start and checkpoint areas as your old system, which means all of these spool files will be gone.
5. IPL your new system using the warm start option. Consider using the NOAUTOLOG option in response to the START prompt to prevent any service machines from coming up with the wrong level of CMS.

**Note:** The spool files of users not known to your new system but who had spool files in your old system are now owned by the OPERATOR user ID.

6. Use SPXTAPE to load the spool files and saved segments from your test system (that you dumped there in step 4) on to your new production system.
7. To see what saved segments and saved systems your new system has, enter:  
q nss

You may see some saved segments or saved systems with the same name or some that you no longer want or need on your new system.

Using the spool ID, purge any saved segments or saved systems that are duplicates or that you no longer want or need. Do not use the names of the saved segments or saved systems to purge them. Using the name may cause you to purge a saved segment you meant to keep.

8. Either shutdown and reIPL your new system, or enter the following command to start up all of your system's service machines:  
xautolog autolog1

**Attention:** Make sure your old production system does not get autologged. Because it is defined with the same checkpoint and warm start areas as your new system, it can corrupt your new system's spool files.

9. Consider using the VMFSGMAP EXEC to further map out your system's saved segment layout and set up VMSES/E to build saved segments. See "Building Saved Segments Individually" on page 69 for a brief description of VMFSGMAP. For further details on managing saved segments see the *z/VM: Planning and Administration* book.

## Using SPXTAPE to Migrate All or Some of Your Spool Files

Another method for migrating your spool files is to use SPXTAPE. You can use this method if you want to migrate all of your spool files or selected spool files. For example, if you want to test out particular saved segments on your new system before bringing it into production, you can migrate only the spool files that contain those segments.

1. If you need to, use QUERY commands on your old system to determine which spool files (reader, printer, punch, saved segments) you want to migrate to your new system.
2. On your old system, use SPXTAPE to dump to tape the spool files you want to migrate to your new system, including saved segments and saved systems.
3. IPL your new system. Make sure you have enough spool space allocated to contain the spool files you intend to load.
4. On your new system, use SPXTAPE to load the spool files you dumped in step 2.
5. To see what saved segments and saved systems your new system has, enter:  
q nss

You may see some saved segments or saved systems with the same name or some that you no longer want or need on your new system.

Using the spool ID, purge any saved segments or saved systems that are duplicates or that you no longer want or need. Do not use the names of the saved segments or saved systems to purge them. Using the name may cause you to purge a saved segment you meant to keep.

6. Consider using the VMFSGMAP EXEC to further map out your system's saved segment layout and to set up VMSES/E to build saved segments. See "Building Saved Segments Individually" for a brief description of VMFSGMAP. For further details on managing saved segments see the *z/VM: Planning and Administration* book.

## Building Saved Segments Individually

If you prefer, you can rebuild each individual saved segment you need on your new system. In general, to plan and map out your new system's saved segment layout, you need to:

- Gather information about the saved segments you need in your system
- Use VMFSGMAP, a segment mapping and planning tool, to help you map your system's saved segments

To plan and set up your saved segment layout in the new system:

1. Identify all of the products or applications in your new system that require saved segments.
2. Collect all of the saved segment definitions (that is, default DEFSEG commands) for each of the products or applications that will use saved segments in your new system.

For products, this information will most likely be in the installation information for each product. You need to gather this information for your system's own applications as well. The type of information you would need is shown in Table 28 on page 70.



## Migration Tasks

Table 28. Example of Saved Segment Information

Product	Segment Name	Space Name	Default Location	Size (pages)	Run Above 16MB?	How is Segment Built?
VM/ESA	CMSBAM	DOSBAM	B10-B3F	30 hex pages (48 decimal)	no	Use VMFBLD

Depending on how you plan to migrate to your new system, you may gather this information in two ways:

- Gradually, as you install or move each product or application
- During your migration planning prior to installing any products

Although it may require more time to plan for your migration, you should consider mapping your entire saved segment layout at once. Otherwise, as you gradually install additional products, you may have to remap and rebuild saved segments on your system.

As you gather saved segment information for your system's products and applications, you may need to take note of additional information about each saved segment that you plan to use in your new system:

- Whether the saved segment is a CMS logical segment or a CP physical segment
- Whether the saved segment can reside above the 16MB line
- What execs or commands are used to build and save the segment

**Note:** There are two situations when you may not need to gather the additional information:

- Depending on how your new system and associated products were packaged, a number of saved segments may already be defined on your system. If this is true, do not collect the information for these segments manually. Later, when you use VMFSGMAP, you can pull saved segment information for these saved segments directly into the VMFSGMAP segment mapping tool using the SEGMERGE macro. You can input any additional segment information later if necessary.
- Some products may be VMSES/E-installed. In this case, these products may already have their default saved segment definitions identified for VMFSGMAP; so you do not need to collect this information manually. When you refer to each individual product's installation manual, the manual should clearly identify whether the product's saved segments are already defined for VMFSGMAP.

The information you gather will be useful to you later when you build your saved segments using the VMFBLD EXEC. You provide this information to VMSES/E using the VMFSGMAP interface.

3. Use VMFSGMAP to add saved segment definitions or to plan for and map out your system's saved segment layout.

With VMFSGMAP, you provide specific information about each saved segment in your system:

- Where it should reside in storage (the DEFSEG statement)
- The name of the saved segment and segment space name if it applies
- Whether it can reside above the 16MB line
- Whether CMS logical segments reside in the saved segment
- What disks must be accessed for the saved segment to be properly built



- Instructions for how to build and save the saved segment

**Note:** Some packaged systems and any products that are VMSES/E enabled already have this information, as well as any other applicable information, identified for you. In general, for these saved segments you update only information about the placement of a saved segment or of a segment space in which the saved segment resides.

Using VMFSGMAP, you can map and manipulate your system's saved segment layout without affecting your running system. How to invoke VMFSGMAP, use its panel interface, and map out and manipulate saved segment layouts is fully described in the *z/VM: Planning and Administration* book.

When manipulating your system's saved segment layout:

- Remember that CMS uses storage locations from 15MB to 20MB. Be sure not to place any saved segments in that area.
  - Make sure that you adequately plan for saved segments that must be run below the 16MB line.
  - Be careful not to overlay saved segments that require one another.
4. Build and save the saved segments you need on your system.

Saved segments that have been completely defined to VMSES/E with the VMFSGMAP interface can be built using the VMFBLD EXEC regardless of whether the saved segment belongs to a product that is completely serviced with VMSES/E. And, for any products or applications that are serviced with VMSES/E, you will be notified whenever service to that product or application requires that a saved segment must be rebuilt. For products or applications that are not VMSES/E-enabled, you can manually provide the necessary information and use the VMFBLD EXEC to build the saved segment.

For example, to build the CMSPIPES saved segment, you would enter:

```
vmfbld ppf segbld esasegs segblist cmspipes ( all
```

For examples of how to use VMFBLD to build saved segments on your system, see the *z/VM: Planning and Administration* book.

## Avoiding the Loss of Spool Files and System Data Files During Migration

Changing how you allocate SPOOL space on your new system may cause you to lose spool files, which can include system data files. For example, you may decide to remove a cylinder or extent of SPOOL space during the migration of your system and make it PERM space for a minidisk. When the minidisk owner formats that space, any spool file that was chained through a page that was changed from SPOOL space to PERM space will then be destroyed. To avoid this potential loss of data:

1. Back up your spool files and system data files using SPXTAPE DUMP before reallocating SPOOL space.
2. Reallocate the SPOOL space.
3. Purge your system data files from the old system.
4. Shutdown the old system. **Do NOT use SHUTDOWN REIPL.**
5. IPL the new system with a cold start.
6. Restore the backed up spool files and system data files using SPXTAPE LOAD.

### Sharing Data between Your Old System and Your New System

If you plan to migrate users from your old system to the new system in a staged fashion, your production will be split between several systems. One major concern is how you will share data between these systems.

z/VM allows you to share information on a DASD volume in the following ways:

- Among multiple virtual machines using virtual reserve/release.
- Among one virtual machine and operating systems running on other processors using real reserve/release.
- Among multiple virtual machines and operating systems running on other processors using concurrent virtual and real reserve/release. The virtual machines and operating systems must support reserve/release CCWs.

### Reserve/Release Considerations for VSE

z/VM supports virtual reserve/release for minidisks that are not a full pack. Therefore, the cross-system communication (also called the “lock file”) volume does not have to be defined as a full pack.

MDISK statements for all DASD you want to mount to VSE as shared (in other words, you want to use the S operand of the IPL ADD statement) must include the V suffix on the link mode. That is, the link mode must be MWV. If this is not done, VSE issues MSG0I23I for the minidisks that do not have link mode MWV on their MDISK statements.

Specifying MWV does not result in any additional overhead because z/VM does not do a reserve/release to any pack unless the guest asks it to. VSE only does a reserve/release to the cross-system communication file (the “lock file”) after IPL.

Note that if the cross-system communication file (the “lock file”) is shared by more than one CPU, SHARED must be YES on the RDEVICE statement in the system configuration file. Also, for sharing a volume concurrently between real and virtual machines, the volume must be defined as a full-pack minidisk.

**Note:** z/VM supports virtual reserve/release for the virtual disks in storage function. Virtual disks in storage are temporary FBA minidisks simulated in system storage rather than mapped to real DASD. Therefore, a virtual disk in storage may be faster than other minidisks because it avoids the overhead of I/O operations. VSE guests may benefit from this function by using a virtual disk in storage instead of a permanent minidisk to store label information areas and the cross-system communication file (the “lock file”). The virtual disk in storage function may be used by a guest running any supported version or release of VSE.

### Examples of Reserve/Release

The following sections include examples of virtual reserve/release and concurrent virtual and real reserve/release. For a complete discussion, refer to the information about DASD sharing in the *z/VM: Planning and Administration* book.

### What If We Cannot Share Data through Reserve/Release?

In some instances, you will not benefit from or will not be able to share DASD through reserve/release. For instance, if:

- You need to share data among multiple virtual machines on multiple systems and the operating systems running in the virtual machines do not support reserve/release CCWs. CMS is an example of a virtual machine that does not support this type of sharing.
- Your system is constrained and you cannot afford the performance degradation that results from shared DASD (particularly concurrent virtual and real reserve/release).
- You need to share databases (such as SQL/DS or DB2<sup>®</sup> Server for VM) between several groups of users and not all of the users can be moved to the new system.
- There are incompatibilities that prevent your old system and new system from accessing the same data.

In these cases, you might:

- Replicate the data.
- Maintain the data on only one system and give two user IDs to users who need to access the data.
- Physically attach the DASDs you want to share to the second system and allow write access from only one system. If you have RACF, you can use it to find out who has access and to limit the write access from one system. You should be aware, however, that this method of sharing is not protected by z/VM; you must set up the controls yourself.
- Use the cross system extensions (CSE) support in z/VM. See “Sharing Data among CMS Users on Multiple Systems” on page 77 for details.

### Sharing Data among Multilevel Virtual Machines

If you want to share data among virtual machines running on the new system that support reserve/release CCWs, and you do not need to share this data with operating systems running on other processors, use virtual reserve/release. “Using Virtual Reserve/Release” explains how to do this. A virtual machine in which MVS is running is an example of a virtual machine that supports reserve/release CCWs.

If you want to share data among virtual machines running on the new system, and the virtual machines do not support reserve/release CCWs, you cannot use virtual reserve/release. A virtual machine in which CMS is running is an example of a virtual machine that does *not* support reserve/release CCWs. “Without Using Virtual Reserve/Release” on page 75 shows you how to share data between virtual machines that do not support virtual reserve/release.

**Using Virtual Reserve/Release:** If you want to share data among virtual machines running on the new system that support reserve/release CCWs, and you do not need to share this data with operating systems running on other processors, use virtual reserve/release.

Virtual reserve/release works the same way on the new system as it does on your old system. Figure 1 on page 74 depicts virtual machines containing MVS that are sharing DASD through virtual reserve/release.

## Migration Tasks

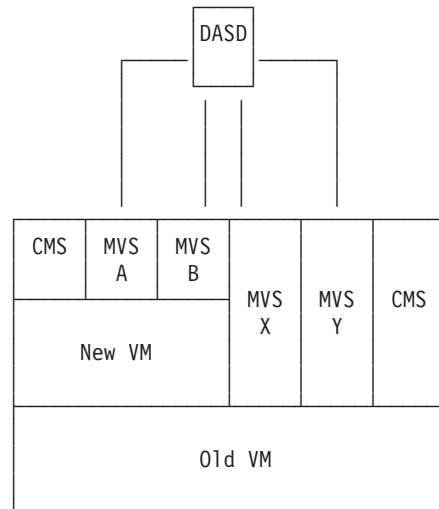


Figure 1. Sharing DASD Using Virtual Reserve/Release

To set up virtual reserve/release, it has to look like concurrent virtual and real reserve/release to the guest, which in the scenario depicted in Figure 1 is your new system.

Do the following on the new (guest) system:

1. Define the DASD as a minidisk for one of the virtual machines, perhaps A, in the new system directory. Specify MWV as the access mode on the MDISK statement. To the second level guest, which is your new VM/ESA system, this minidisk must be viewed as a full pack minidisk.
2. Use the SHARED YES operand on the RDEVICE statement in your SYSTEM CONFIG file on the new system guest for this minidisk. For example:  

```
Rdevice 0cc0 Type Dasd Shared yes
```

Or use the SET SHARED command on the new system guest for this minidisk. For example:

```
set shared on for 0cc0
```

3. Code the LINK statement for the remaining virtual machine, B, in the new system directory. Specify MW as the access mode.

Do the following on the old (first-level) system:

1. Define the DASD where the minidisk resides on the RDEVICE macro in HCPRIO and on the SYSCPVOL or SYSUVOL macro in HCPSYS.
2. Code the MDISK statement in your new system's directory entry in the old system's directory. You must append a V to the primary access mode (read, write, multiple write, and so on) indicating that this minidisk can be shared between virtual machines. For example:  

```
MDISK 197 3390 000 400 WORKPK MWV ORANGE
```
3. Code the LINK statement in X and Y's directory entry. For example:  

```
LINK NEWESA 197 197 MW
```
4. Specify that the DASD will not be shared with another operating system. The default setting of the SHARED option of the RDEVICE macro (SHARED=NO) takes care of this for you.

Now, virtual machines running on your new system (the second-level system) may have write access to the same information as virtual machines running on your old system (the first level system).

**Without Using Virtual Reserve/Release:** If you want to share data among virtual machines running on the new system, and the virtual machines do not support reserve/release CCWs, such as CMS, you cannot use virtual reserve/release. You define the DASD as a minidisk, giving read/write access to only one of the CMS users running on the new system guest; the others can have only read access. This is shown in Figure 2.

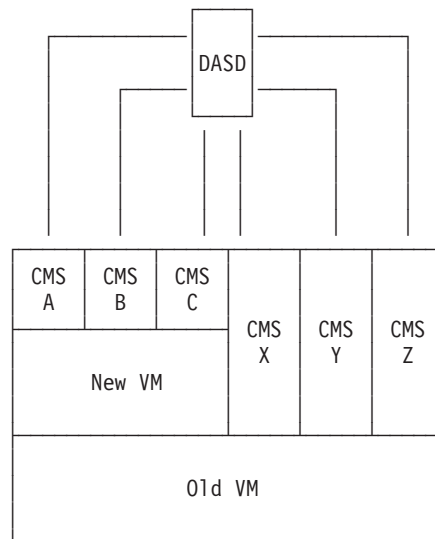


Figure 2. Sharing DASD without Using Virtual Reserve/Release

Do the following on the new (guest) system:

1. Define the DASD as a minidisk for one of the virtual machines, perhaps A, in the new system directory. Specify MW as the access mode on the MDISK statement.
2. Code the LINK statement for the remaining virtual machines, B and C, in the new system directory. Specify RR as the access mode.

Do the following on the old (first-level) system:

1. Define the DASD where the minidisk resides on the RDEVICE macro in HCPRI0 and on the SYSCPVOL or SYSUVOL macro in HCPSYS.
2. Code the MDISK statement in your new system's directory entry in the old system's directory. Do not append a V to the primary access mode. For example:  

```
MDISK 197 3390 000 400 WORKPK MW ORANGE
```
3. Code the LINK statement in X, Y, and Z's directory entry. For example:  

```
LINK NEWESA 197 197 MW
```
4. Specify that the DASD will not be shared with another operating system. The default setting of the SHARED option of the RDEVICE macro (SHARED=NO) takes care of this for you.

## Migration Tasks

Now, virtual machines running in the new system may have read access to the same information as virtual machines running on the old system.

### Sharing Data among Virtual Machines and Other Systems

To share data among multiple virtual machines running on the new system and on other (older) systems, use concurrent virtual and real reserve/release support. You can do this, however, only if the virtual machines that are sharing the data support reserve/release CCWs. In the example shown in Figure 3, the virtual machines containing MVS on the new system can share DASD in this manner with the virtual machine containing MVS on the old system. The virtual machines containing CMS cannot share DASD because CMS does not support reserve/release CCWs. You have to replicate the data for the CMS users or physically attach the DASD and give write access to only one system.

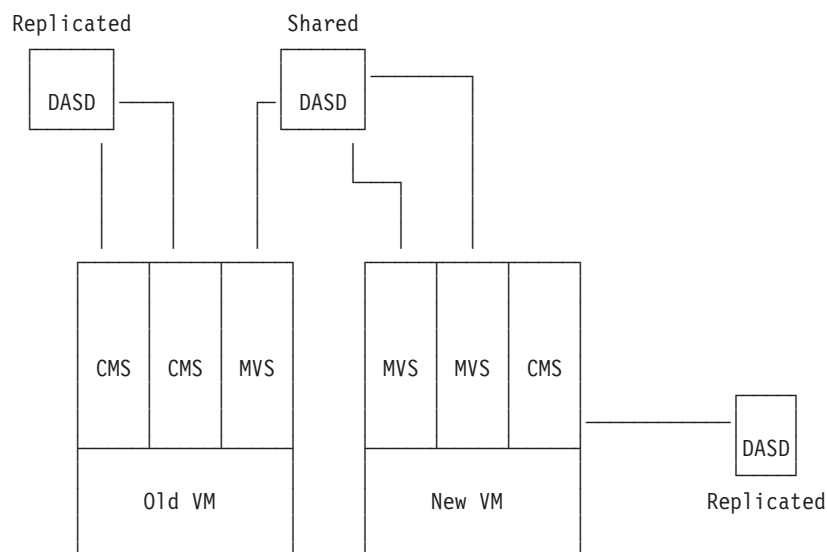


Figure 3. Sharing DASD between Virtual Machines on Multiple Systems

Concurrent virtual and real reserve/release support can be invoked either during system generation or at any time while the system is running.

Do the following to invoke concurrent virtual and real reserve/release while generating the system:

1. Ensure that the operating system running as a guest on the new system and the operating system with which you are sharing the DASD both support reserve/release CCWs.
2. Define the DASD as a shareable full-pack minidisk. To do this, use the MDISK statement in the user directory.

To define the DASD as a full-pack minidisk, the starting cylinder or block number must be zero and the number of cylinders or blocks must equal or exceed the number of cylinders or blocks on the real device. To define the DASD as virtually shareable, include the V in the mode definition. For example:

```
MDISK 327 3380 000 885 MVS003 MWV
```

or,

```
MDISK 328 9336 0000 END MVS003 MWV
```

3. Define the DASD as being shareable between multiple real and virtual systems by specifying the SHARED YES option on an RDEVICE statement in SYSTEM CONFIG. For example:

```
Rdevice 0327 Type Dasd Shared yes
```

**Note:** Unless SHARED YES is specified, CP assumes that the device is not shared.

---

## Sharing Data among CMS Users on Multiple Systems

You can use cross system extensions (CSE) to share access to minidisks among CMS users on different systems as if they were on the same system. Refer to *z/VM: Planning and Administration* for detailed information about CSE.

### Preparing for Cross-System Link

Activation of cross-system link requires that the volumes be defined in the CP\_OWNED or USER\_VOLUME\_LIST statements in the SYSTEM CONFIG file and switched online to all systems. Also, several other system configuration statements need to be added to the SYSTEM CONFIG file:

- XLINK\_SYSTEM\_EXCLUDE—to specify systems that CP is to exclude from the cross-system link.
- XLINK\_SYSTEM\_INCLUDE—to specify systems that CP is to include in the cross-system link.
- XLINK\_VOLUME\_EXCLUDE—to define DASD volumes that are to be excluded from the cross-system link operation.
- XLINK\_VOLUME\_INCLUDE—to define the DASD volumes to be included in the cross-system link operation.

The statements are described in *z/VM: Planning and Administration*.

### Preparing for Cross-System Spool

Activation of cross-system spool requires that the spooling volumes be defined to the control programs by the CP\_OWNED statement in the SYSTEM CONFIG files of all CSE systems in the complex. The spooling volumes must be switched online to those systems. Additional configuration statements may be needed, as follows:

- XSPOOL\_SYSTEM—to specify the systems that are to participate in cross-system commands and spooling operations.
- XSPOOL\_TRACE—to define the number of pages of storage that CP should allocate for the cross-system spool (XSPOOL) trace tables.
- XSPOOL\_XLIST\_INPUT—to specify virtual machines whose input spool files will not participate in cross-system spooling and cross-system message and query commands.
- XSPOOL\_XLIST\_OUTPUT—to specify virtual machines whose output spool files will not participate in cross-system spooling and cross-system message and query commands.

Finally, all spooling volumes must be CP-formatted for the new system. Once these conditions are met, cross-system spooling is activated by an explicit operator command and can be started on one system at a time until the entire complex is operating with shared spool.



## Migration Tasks

Any system not starting cross-system spool will continue to operate in the normal way without shared spool. However, once a complex has become operational, systems cannot be removed and then added again without consideration for the spool files that belong to different users and are now spread throughout the complex. These files are now owned by different systems and must be collected and redistributed if CSE operation is discontinued.

---

## Migrating Your User Directory

This section helps you migrate your old directory to the new system. Do not migrate the user directory until you finish installing the new system.

### Object Directory Compatibility

Object directory compatibility is supported when the object directory created with the DIRECTXA MODULE on the new system is used by **other currently supported releases of CP**. This support lets you use a single source directory in a mixed Cross System Extensions (CSE) environment, or when migrating to the new release of CP.

A copy of the new DIRECTXA MODULE should exist on each system that will share a single source directory and depend on object directory compatibility. A new copy of the DIRECTXA MODULE should be distributed to each of the systems whenever service is applied to the HCPDIR ASSEMBLE file on the new system, which is used to create the DIRECTXA MODULE.

### Getting Around Directory Differences

If you want to use the same source directory for your old and new systems:

- **Migrating from an Unsupported Release:** Avoid using any directory statements or options that were introduced after your current release. MIXED directory APARs do not exist for any subsequent releases. To include statements or options that were introduced after your release, you must maintain two source directories. The statements and options that are unsupported by your old system should be included only in the source directory for the new system.
- **Migrating from a Supported Release:** You can do either of the following:
  - Use the DIRECTXA MODULE on the new system to create the object directory for both systems, as described above.
  - Avoid using any directory statements or options that were introduced after your release. MIXED directory APARs do not exist for any subsequent releases.

### User Directory Control Statement Changes

For a list of control statement changes, refer to the control statements compatibility table (if any) for each of the releases since your old release.

### Default Volume Label Changes

The default volume labels have changed from your old release to the new release. The new prefix is 310. For example, the label for the system residence volume is 310RES.



## Preparing to Migrate Your User Directory

If you have a problem IPLing the new system, it may be because of changes you have made to the directory. If you suspect this, you can use the NODIRECT option when you bring up the new system. This brings up the system without accessing the user directory.

Before you can use NODIRECT, you need to know information about the source directory's minidisk, the CMS system disk, and the system residence volume. Enter this information now in Table 29 to help you remember it. It may be too late to get this information if you wait until you have directory problems. The first three entries in Table 29 are examples. Refer to the *z/VM: Planning and Administration* book for more information on loading z/VM 3.1.0. Refer to the *VM/ESA Installation and Service Sample Files* document packaged with z/VM for information on the IBM default location and sizes of minidisks.

Table 29. Information Needed To Use NODIRECT Option

	Minidisk or Volume Address	Volume Identifier	Starting Cylinder or Block	Number of Cylinders or Blocks Allocated
Minidisk containing source directory - <b>example</b>	2C2	310RES	430	5
CMS system disk - <b>example</b>	190	310RES	302	128
System residence volume - <b>example</b>	123	310RES	0	END
Minidisk containing source directory				
CMS system disk				
System residence volume				

## Using Two Source Directories

During your migration, you may choose to maintain either two source directories or one. If you use two directories, one on the old system and one on the new, you need to be very careful that any changes are reflected in both directories, if needed. Using two directories allows you to exploit the new functions of z/VM 3.1.0 without worrying about backing out.

## Considerations for Using a Directory Maintenance Program

If you have already installed a directory maintenance program on your z/VM 3.1.0 system, or if you are using only one directory, you may have to issue additional commands while migrating the source directory.

### Example Using IBM Directory Maintenance VM/ESA (DirMaint)

For example, several additional steps are needed if DirMaint is used:

1. Before changing the source directory, enter DIRM DISABLE. This ensures that DirMaint does not try to update the directory while you are converting it.
2. Enter DIRM USER BACKUP. This creates a CMS flat file of the current source directory named USER BACKUP and found on DirMaint's 1DB minidisk (G-disk).
3. Update USER BACKUP with the new directory statements.
4. After updating, copy USER BACKUP to DirMaint's 1DF minidisk (E-disk) and rename it to USER INPUT.
5. Erase any USER DIRECT E file and start DirMaint by autologging it or by running DVHBEGIN or DIRM RLDDATA.

## Migration Tasks

6. Enter DIRM ENABLE to allow DirMaint to resume making updates.

For complete information on how to make manual changes to the source directory when DirMaint is installed and running, see:

- *Directory Maintenance VM/ESA: Tailoring and Administration Guide*
- *Directory Maintenance VM/ESA: Command Reference*
- *IBM Directory Maintenance VM/ESA Program Directory*

## Steps for Migrating Your Source User Directory

Whether you use two directories or one during migration, you can migrate your source directory by merging information from the old directory into the new directory. The old directory is left unchanged and can be either discarded or used on the old system. To do this:

1. Make a copy of the source user directory on the old system with a different file name. Move the copy to the new system.
2. Edit the z/VM 3.1.0 directory and merge pieces from the old directory into it as the old directory entries are migrated. To migrate the old directory entries:

- a. Migrate each system user ID.

IBM supplies a sample directory with the VM/ESA product. In this directory, there are several system user IDs defined. For each system user ID that you used in your old system, verify that your directory entries are applicable in z/VM 3.1.0:

- If applicable, make the needed changes in the new directory's entry.
- If not applicable, leave the new directory's entry as is.

- b. For all other user IDs:

**Note:** This step may be staged or done in groups as you move the users to the new system.

- 1) Increase storage sizes as needed. The minimum storage size for a shared copy of CMS is 256KB. At least 20MB is needed to IPL a nonshared copy of CMS, for example to IPL the 190 minidisk.
- 2) Verify that AUTOONLY, NOLOG, NOPASS, or LBYONLY is not specified in the password field of the USER statement unless the user is to take advantage of the function the operand provides.
- 3) Specify the type of virtual machine you want your user IDs to have. You can specify an XA, ESA, or XC virtual machine.

**Note:** CMS is no longer supported in a 370 virtual machine.

- 4) Look for usage of incompatible directory control statements. See the user directory control statement compatibility tables in this book for the list of incompatible statements. Note any defaults that changed since your old release.
- 5) Resolve any incompatibilities by:
  - Removing directory control statements that are no longer supported.
  - Changing directory control statements with changed parameters.
  - Optionally, exploiting new directory control statements and parameters.
- 6) Move all migrated user IDs to the new directory.

3. Check the syntax of your new directory using the DIRECTXA EDIT command.  
For example:  

```
directxa user direct (edit
```
4. Fix any problems.
5. Put the new directory on-line using the DIRECTXA command, for example,  

```
directxa user direct
```

## What If I Cannot IPL Because of a Problem with the User Directory?

If you have a problem IPLing z/VM 3.1.0, you can try using the NODIRECT option. Because z/VM 3.1.0 does not support a stand-alone directory function, you must use the following steps to recover when you cannot IPL because of a problem with the user directory. The information recorded in Table 29 on page 79 is used in this procedure.

1. IPL and bring up the system using the NODIRECT option. This logs on the primary system operator.
2. Use the DEFINE MDISK command to obtain access to the minidisks containing the directory source file, the CMS system disk, and the CP system residence volume. You recorded the minidisks addresses, starting values, and sizes in Table 29 on page 79.

For example, using the sample information in Table 29 on page 79:

```
define mdisk as 2c2 430 5 310RES
define mdisk as 190 302 128 310RES
define mdisk as 123 0 END 310RES
```

3. IPL 190
4. Enter the ACCESS command to access the minidisk that contains the source directory.
5. Fix the problem with the directory.
6. Put the corrected directory on-line using the DIRECTXA command. For example:  

```
directxa user direct
```
7. Shutdown and re-IPL the system without using the NODIRECT option.

Refer to *z/VM: System Operation* for a complete step-by-step procedure for recovering the user directory after a problem during IPL.

---

## Migrating Your SFS File Pool Servers

You will probably want to have the *z/VM: CMS File Pool Planning, Administration, and Operation* book available as you do this task.

## Using Two System Images

For this type of migration, you have both the old system and the new system running at the same time. The new system can be installed on a separate physical processor, in a logical partition, or as a second level system on your old system.

You can do the migration in two ways:

- Move the entire user population for your SFS file pool servers all at once.
- Stage the move by grouping your users, for example, by department, by project, or whatever is appropriate.

## Migration Tasks

Staging the move may be time consuming. Also, if you are not careful, aliases and authorizations may be lost in the move.

You can automate the move somewhat by using the SFSTRANS EXEC, which is documented in the *z/VM: CMS File Pool Planning, Administration, and Operation* book, as a model. Note that SFSTRANS assumes that you have access to both the old and the new servers.

### Procedure

1. Use FILEPOOL BACKUP for each storage group to back up all the data on the old release.

For assistance, refer to the section on backing up the user data in the *z/VM: CMS File Pool Planning, Administration, and Operation* book. This backup can be used on your new server. It can also be used if you need to back out to your old release.

#### Notes:

- a. **[2.1.0]** Control data backup files created on your new system are not supported on the old system.
2. Shut down the SFS file pool servers on your old system.

Enter the STOP operator command. Do not use STOP IMMEDIATE. For example, from the server machine console, enter:

```
stop
```

Or, from a secondary user console, such as MAINT, enter:

```
#cp send vmserv3 stop  
#cp send vmserv4 stop
```

3. Install the new system.  
During the install, make sure the new CMS code gets loaded, but do not install or set up the SFS servers. (The CMS code contains the SFS server code.)
4. Set up a file pool server machine on the new system.

#### **If you are moving everyone at once:**

If you can, move the entire DASD pack that contains the SFS minidisks to the new system. Use the same minidisk locations and addresses defined in the directory entry for the server on the old system. If needed, update the directory entry as described in “Directory Entry Considerations and Changes” on page 85.

If you cannot move the entire DASD pack to the new processor, configure your directory entry for the new server so that it has identical minidisk addresses and sizes as the server on the old system. If you are moving the minidisks to a DASD of a different type, it may be impossible for the sizes to be exactly the same. In that case, make them slightly larger. Then, use the DFSMS COPY command or the DDR command to move the contents of the old server’s minidisks onto the new server’s minidisks. See the *z/VM: CMS File Pool Planning, Administration, and Operation* book for more information about how to prepare for and use DFSMS COPY.

#### **If you are staging the move by groups:**

Make sure the new file pool server machine has enough physical DASD space to hold the group of users that you want to move. See the section on generating a file pool and server in the *z/VM: CMS File Pool Planning, Administration, and Operation* book for details on how to do this.

5. **If you are staging the move by groups:**

Move the users in the group that you want to move to the new file pool. Consider modifying the SFSTRANS EXEC, which is shown in the *z/VM: CMS*

*File Pool Planning, Administration, and Operation* book, to automate this procedure for you. To move users to the new file pool, do the following:

- a. Enroll the users in the new file pool. Make sure the users have enough file blocks to contain their data.
- b. Re-create each user's directory structure in the new file pool.
- c. Copy or move the users' files from the old file pool to the new file pool.
- d. Remove the users' space from the old file pool.

If you can, use the DELETE USER command to delete a user from the old file pool. However, if a user needs to write to others' files in the old file pool, you can:

- Delete the user from the old file pool, then immediately re-enroll the user, or
- Erase the copied or moved files, and reduce the user's space to 0 using the MODIFY USER command.

**Notes:**

- 1) If you delete the user (using the DELETE USER command) from the old file pool, you may need to regrant authorizations. Authorizations that were granted to that user ID are deleted as well as any authorizations the user ID granted.
  - 2) Aliases that others have for the moved files are lost, and aliases that the moved user has in the old file pool are lost. If the users still need to share files, they have to access each others' directories. Aliases cannot refer to base files in other file pools.
6. Log on to the file pool server machine. Make sure it uses the new CMS code, which is typically:
    - CMS code: In the CMS segment or on the 190 minidisk, and
    - SFS code: On the 193 minidisk or in the CMSFILES segment.
  7. IPL the new CMS.
  8. Check the USERS startup parameter in the DMSPARMS file for your server and the MAXCONN value in the server's directory entry against the suggested values found in the *z/VM: CMS File Pool Planning, Administration, and Operation*. This can help you avoid potential virtual storage problems.
  9. If BACKUP is specified in the DMSPARMS file, issue FILESERV BACKUP to back up control data. You must do this before issuing FILESERV START or you will get message DMS3440E, and FILESERV START will not continue.
 

```
fileserv backup
```
  10. Start the SFS file pool servers.
 

If FILESERV START is not included in the PROFILE EXEC of the file pool server virtual machine, enter it from the operator's console:

```
fileserv start
```
  11. Immediately back up your server data. Use FILEPOOL BACKUP for each storage group to back up all the data on the old release.
 

**Note:** Control data created on your old release cannot be used on the new release Storage group (user data) backup files created on your old release can be used on the new release.
  12. If you encounter problems and need to back out to your old system, see "Converting an SFS File Pool Server Back to Your Old Release" on page 85.

### Using the 'Cut and Go' Strategy

The 'cut and go' strategy is to move immediately off the old system and right on to the new one. This approach is recommended only if your system is a guest system or is small or very straightforward.

#### Procedure

1. Use FILEPOOL BACKUP for each storage group to back up all the data on the old release.

For assistance, refer to the section on backing up the user data in the *z/VM: CMS File Pool Planning, Administration, and Operation* book. This backup can be used on your new server. It can also be used if you need to back out to your old release.

**Note:** Control data created on your old release cannot be used on the new release. Storage group (user data) backup files created on your old release can be used on the new release. Neither control data nor storage group backup files created on your new system are supported on the old system.

2. Review and make note of the start-up parameters, which are in the DMSPARMS file, for each server on your old system.
3. Shut down your old-system SFS file pool servers.

Enter the STOP operator command. Do not use STOP IMMEDIATE. For example, from the server machine console, enter:

```
stop
```

Or, from a secondary user console, such as MAINT, enter:

```
#cp send vmserv3 stop  
#cp send vmserv4 stop
```

4. Install the new system.  
During the install, make sure the new CMS code gets loaded (the CMS code contains the SFS server code), but do not install or set up the SFS servers.
5. Make sure the server virtual machine has access to the disks where the new CMS code resides, which is typically:
  - CMS code: In the CMS segment or on the 190 minidisk, and
  - SFS code: On the 193 minidisk or the CMSFILES segment.
6. IPL the new CMS.
7. Check the USERS startup parameter in the DMSPARMS file for your server and the MAXCONN value in the server's directory entry against the suggested values found in the *z/VM: CMS File Pool Planning, Administration, and Operation* book. This can help you avoid potential virtual storage problems.
8. If BACKUP is specified in the DMSPARMS file, issue FILESERV BACKUP to back up control data. You must do this before issuing FILESERV START or you will get message DMS3440E, and FILESERV START will not continue.

```
fileserv backup
```

9. Start the SFS file pool servers.

If FILESERV START is not included in the PROFILE EXEC of the file pool server virtual machine, enter it from the operator's console:

```
fileserv start
```

10. If you encounter problems and need to back out to your old system, see "Converting an SFS File Pool Server Back to Your Old Release" on page 85.



## Directory Entry Considerations and Changes

You may have to modify some of the CP directory entries for your SFS file pool servers. If you have not done so already, make the following changes:

- **USER directory control statement:**
  - Make sure the minimum and maximum virtual storage are at least 32MB.
- **OPTION directory control statement:**
  - Add the NOMDCFS operand to allow the server to use minidisk caching at a rate that is not limited by the Fair Share Limit.

**Note:** This is applicable only to SFS file pool servers, not CRR recovery servers.

- Add the QUICKDSP operand to allow the server to be added to the dispatch list immediately when it has work to do, without waiting in the eligible list.
- Add the SVMSTAT operand to specify that the virtual machine is a service virtual machine. This causes the server's monitor statistics to be reported separately from end-user virtual machines.
- Check your MAXCONN value in the server's directory entry against the suggested values found in *z/VM: CMS File Pool Planning, Administration, and Operation*. This can help you avoid potential virtual storage problems.
- **SHARE directory control statement:**
  - Add SHARE REL 1500 to place the server in a more favorable position in the dispatch queue.
- **MACHINE directory control statement:**
  - If not already done, replace 370 with XA or XC.  
SFS file pool servers should use XC to exploit data spaces. CRR recovery servers do not exploit data spaces and therefore should be set to XA.
- **XCONFIG directory control statement:**
  - If you designated XC in the MACHINE directory control statement, then add the following XCONFIG directory control statements:  
XCONFIG ADDRSPACE MAXNUMBER 100 TOTSIZE 8192G SHARE  
XCONFIG ACCESSLIST ALSIZE 1022

These statements have appropriate data space values for an SFS file pool server.

- **MINIOPT directory control statement:**
  - Add MINIOPT NOMDC to inhibit expanded storage caching for the following file pool minidisks:
    - For SFS file pool servers:
      - Control minidisk
      - SFS log minidisks
    - For CRR recovery servers:
      - All file pool minidisks

## Converting an SFS File Pool Server Back to Your Old Release

z/VM does not support the conversion of SFS file pools and servers back to a previous release. However, IBM does recognize that special situations exist where such a conversion may be desired. The following procedure is intended to provide some guidance. Please read this entire section before beginning the conversion.

## Migration Tasks

If you do the conversion incorrectly, some of the problems you may encounter include:

- Not being able to start the file pool server in your old system
- Loss of data

### Procedure

To avoid file pool server problems, you must perform the following steps, where required, in the order indicated:

1. Shut down the new SFS file pool server.

Stop file pool server multiple user mode processing by entering the STOP operator command. **Do not use STOP IMMEDIATE.** For example, from the server machine console enter:

```
stop
```

Or, from a secondary user console, such as MAINT, enter:

```
#cp send vmservu stop
```

If your SFS file pool server is also a CRR recovery server, this step ensures that all CRR logging activity completes normally. There are no additional migration considerations if the SFS file pool server is also a CRR recovery server. (Your SFS file pool server is a CRR recovery server if the CRR start-up parameter exists in its *serverid* DMSPARMS file.)

Also, if possible, do not change the LU name value on the LUNAME start-up parameter. See *z/VM: CMS File Pool Planning, Administration, and Operation* for LUNAME start-up parameter considerations.

2. Convert your system back to the old release.
3. On the old system:
  - a. If you are using SFS control backup, which means you used the BACKUP start-up parameter, execute the FILESERV BACKUP command to create a new control backup file. For example:

```
fileserv backup
```

**Note:** Control backup files created by one VM release cannot be processed by a different (earlier or later) release.

- b. Start the old-system SFS file pool server for multiple user mode processing. If FILESERV START is not included in the PROFILE EXEC of the file pool server virtual machine, enter it:

```
fileserv start
```

Your SFS file pool conversion back to the old release is now complete.

---

## Establishing Connectivity between Your New and Old Systems

Establishing connectivity between your new system and your old system allows you to move data files from the old system to the new system. If you have installed the new system on a separate processor or in another logical partition on the same processor as the old system, you can use real channels to establish connectivity between the two systems. If you have installed the new system as a guest of the old system, you can use virtual channels to establish connectivity between the two systems.



For example, you can set up a virtual channel between the RSCS licensed program installed on the old system and RSCS installed on the new system. A virtual channel for using PVM, VTAM®, TSAF, or AVS could be set up in a similar manner.

To set up RSCS, you make a virtual channel and add the proper definitions in the configuration and control files. RSCS relies on GCS for supervising services. The RSCS virtual machine must be defined as part of the GCS group so that it can share a common storage area for information exchange, multitasking services, and general I/O services.

Figure 4 shows the relationship between the old (first-level) and new (second-level) RSCS machines.

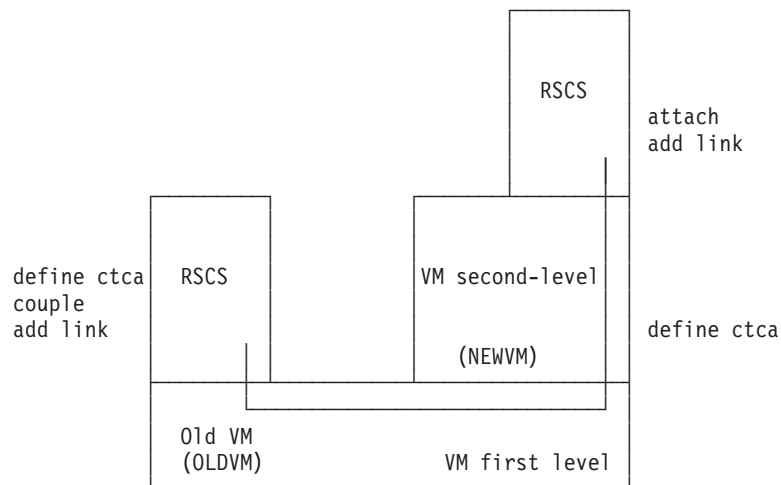


Figure 4. Connectivity between First- and Second-Level RSCS Machines

## Procedures

### On the First-level System

1. Define a link between the first-level RSCS virtual machine and the user ID that will be running the second-level system.

In this example, the user ID that will be running the second-level system is NEWVMESA.

The virtual address of the channel that is used for the link is 312.

- a. From **both** the first-level RSCS virtual machine and the NEWVMESA user ID, define the channel using the CP DEFINE command or the SPECIAL user directory control statement.

For example, using the CP DEFINE command, you can enter:

```
define ctca 312
```

Or, in the user directory entries for NEWVMESA and the RSCS virtual machine, add the following statement:

```
SPECIAL CTCA 312
```

- b. Ensure that the channel you specified using DEFINE CTCA is defined in the second-level system's SYSTEM CONFIG file.

## Migration Tasks

In our example, we are using 312 as the address of the channel that is used for the link.

2. From either the RSCS machine or the NEWVMESA user ID, couple the addressed links to enable communications over the virtual channel. For example, from the first-level RSCS virtual machine, you can enter:  

```
couple 312 newvmesa 312
```
3. Bring up the second-level system, including the second-level RSCS machine.

### On the Second-level System

1. From the second-level RSCS virtual machine, attach the channel address defined by NEWVMESA. For example, enter:  

```
attach 312 rscs 312
```

---

## How to Back Out

You may have to back out of the migration because of either system problems or problems with application programs.

**Note:** To back out spool files, use the SPXTAPE command.

### Backing Out Because of System Problems

How you back out depends on whether you are using separate residence volumes for the old system and the new system or the same residence volume for both.

#### Using Separate Residence Volumes

If you run into a problem with z/VM 3.1.0 that forces you to back out, and you have not yet exploited functions specific to z/VM 3.1.0 or migrated any CMS users, the backout will not affect your end users. However, if you have to back out after migrating some CMS users, you will greatly affect those users. Once users convert their applications to run on the z/VM 3.1.0 CMS, they may not be able to run these applications on the old CMS. Therefore, you should not migrate users to z/VM 3.1.0 until you are satisfied with its stability.

Throughout the migration, you should maintain your old system residence volume and CP-owned packs. Then, if you need to back out, you can easily do so.

If you must back out from z/VM 3.1.0 and IPL an old system, do the following:

1. Dump the spool files with the SPXTAPE DUMP command.
2. Shut down the z/VM 3.1.0 system.
3. Re-IML the processor to the appropriate mode, if needed.
4. IPL the old system.
5. Load the spool files with the SPXTAPE LOAD command.
6. Enable the terminals.

#### Using the Same Residence Volume

In preparation for a possible backout:

1. Before going to the new system, use stand-alone DDR to dump the nucleus:  

```
ddr dump nuc
```
2. When you go to the new system, use the same CP-owned volumes, warmstart area, and checkpoint area. Also, do not overwrite where the old system's nucleus was; keep that space available.

Then, if you have to back out, do the following:

1. Use the SPXTAPE DUMP command to save the spool files, if you want to.
2. IPL the stand-alone DDR program.
3. Use DDR to restore the nucleus to the system residence pack.
4. IPL the system residence pack.
5. Use the SPXTAPE LOAD command to restore any saved spool files.

## Backing Out Individual Users Because of Problems with Application Programs

When the system is stable, you can begin to migrate CMS users. To prepare for a possible backout, either dual-path the code or keep copies of the old (pre-migration) versions of your applications.

### User Directory Concerns

If a user or group of users has problems running applications on z/VM 3.1.0 CMS, you will need to backout these users to the old release. To plan for this, keep a user directory entry in an old-level system for each z/VM 3.1.0 user until you are satisfied that the users are running smoothly. You can use either an old-level guest or a separate old-level system as the backout system. This will be extra work for you but will have the least impact on the end users if they run into problems.

If you are using two source user directories, you must remember to reflect changes made in one user directory to the other user directory.

---

## Installing a Backlevel CMS

IBM provides limited support for multiple levels of CMS on z/VM 3.1.0. You can continue to use your production CMS with the CP component of z/VM 3.1.0, and then gradually migrate users and applications to the new CMS. Previous releases of the CMS component will be supported for a limited time following the general availability of z/VM 3.1.0.

There is no intent to retrofit new function onto old releases of CMS. Attempts to use new function on backlevel releases of CMS are unsupported and the results are undefined. New function is defined as any device support, new CMS, or new CP functions introduced in later releases of the operating system and not retrofitted to the old CMS through the APAR service stream.

IBM will help with problem determination in these mixed environments and will take APARs for problems in older releases of CMS. However, when IBM discontinues service of an old VM release, support for the corresponding CMS under this offering also ends.

### Notes:

1. Throughout this discussion, backlevel CMS refers to an older level of CMS, still in service, running on z/VM 3.1.0 CP.
2. z/VM 3.1.0 CMS does not support back levels of CP.

## Before You Install a Backlevel CMS

While you are still running on your old system, there is certain information you should record that will help when you install your backlevel CMS. Record this information in Table 30 on page 91.

- Record the MDISK statement that defines the 190 minidisk on your old system. This can be used as a reference when defining the minidisk to hold the backlevel CMS on z/VM 3.1.0.

## Migration Tasks

- Record the number of cylinders needed for your old 190 disk. You can get this information from the MDISK statement that defines your old 190 disk, or you can enter:  

```
q virtual 190
```
- Determine the type of DASD your old CMS will reside on when you install it on your z/VM 3.1.0 system. While installing a backlevel CMS, you will have to copy the contents of your 190 minidisk from the old system to your z/VM 3.1.0 system.
  - If you use the same type of DASD to hold the backlevel CMS on your z/VM 3.1.0 system as you used on your old system, you can use DDR to copy the files over. This is the preferred method.
  - If the DASD type is not the same, you can use the VMFPLC2 command. This method is not as easy as using DDR because you have to rebuild the CMS nucleus.

Record the type of DASD you will use and how you will copy files over to z/VM 3.1.0.

- If you are using the VMFPLC2 command to copy files, you have to format the z/VM 3.1.0 minidisk that holds the backlevel CMS using the FORMAT command with the RECOMP option. The RECOMP option changes the number of cylinders on a disk that are available to you. By using the RECOMP option on the FORMAT command you can leave a number of cylinders on the minidisk available for the CMS nucleus. If you do not FORMAT the minidisk with RECOMP, you cannot build the CMS nucleus.

To figure out how many cylinders to recomp, from your old system enter:

```
query virtual 190
```

An example of what this command returns is:

```
DASD 0190 3380 SYGEMC R/0    72 CYL  ON DASD
```

Now enter:

```
query disk s
```

An example of what this command returns is:

```
LABEL  VDEV M  STAT  CYL TYPE BLKSIZE  FILES  BLKS USED-(%)  BLKS LEFT  BLK TOTAL
CMSOLD 190  S   R/0   66 3380 4096      0          6-0      4794      4800
Ready;
```

The difference in the number of cylinders displayed by these two commands is the amount of space needed to hold the CMS nucleus. The number of cylinders shown in the QUERY DISK response is the number of cylinders needed to hold the contents for the 190 disk and is also the number of cylinders to recomp. In this example, 66 cylinders are required on a 3380 DASD. By recomping 66 cylinders, six cylinders are available for the CMS nucleus.

If you are using the VMFPLC2 method because your DASD types are not the same, the number of cylinders you have to recomp may not be exactly the number calculated above. This is because the number of pages per cylinder varies from DASD type to DASD type. To make sure you are recomping a sufficient number of cylinders:

1. Figure out how many pages per cylinder there are for each DASD type you are using. You can find this information in the DASD storage capacity tables in the *z/VM: Planning and Administration* book. For example, a 3380-E has 150 pages per cylinder and a 3390-1 has 180 pages per cylinder.

2. Multiply the number of cylinders needed to hold the contents of your old 190 disk (66 in this example) by the pages per cylinder for each DASD.

**For 3380 model E:**

$$66 * 150 = 9900 \text{ pages}$$

**For 3390-1:**  $66 * 180 = 11880 \text{ pages}$

This shows you that you need 9900 pages of storage for the contents of the old 190 disk. If you recomp 66 cylinders of 3390-1 DASD, this gives you 11880 pages of storage. Therefore, 66 cylinders is a sufficient number of cylinders to recomp.

3. Multiply the number of cylinders needed to hold the CMS nucleus (six in the previous example) by the pages per cylinder for each DASD. For example,

**For 3380 model E:**

$$6 * 150 = 900 \text{ pages}$$

**For 3390-1:**  $6 * 180 = 1080 \text{ pages}$

If you reserve six cylinders on a 3390-1 DASD, it is plenty of space to hold the CMS nucleus that you stored in six cylinders on a 3380 model E.

If you calculated the number of cylinders to recomp on a 3390-1 DASD to be 66 cylinders, you would have to recomp 80 cylinders on a 3380-E DASD to store the same amount of data. This is because there are fewer pages per cylinders on a 3380-E.

Record the number of cylinders you need to recomp in Table 30.

Table 30. Information Needed Before Installing a Backlevel CMS

What	Example Information	Your Information
MDISK statement for old VM system 190 disk	MDISK 190 3380 125 72 CMSRES RR ALL	
Number of cylinders for old VM system 190 disk	72	
DASD type for old VM system CMS	3380	
DASD type for backlevel CMS on new system	3390	
Method you will use to copy files (DDR or VMFPLC2)	VMFPLC2	
If using VMFPLC2, number of cylinders to recomp	66	

## Steps for Installing a Backlevel CMS

When you are done installing the old CMS, the environment will look like this:

## Migration Tasks

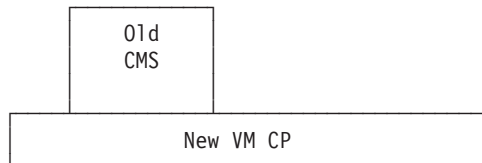


Figure 5. Backlevel CMS Running on a z/VM 3.1.0 CP

The steps for installing the old CMS on the z/VM 3.1.0 CP are as follows:

1. Log on to your z/VM 3.1.0 MAINT user ID.
2. Create a minidisk where the backlevel CMS will reside. You can use the MDISK directory control statement from your old system as a reference. You recorded this in Table 30 on page 91.

The MDISK directory control statement in your z/VM 3.1.0 user directory for the backlevel CMS may look like this:

```
MDISK 590 3380 125 72 310RES MR ALL WRITE MULTIPLE
```

- The 590 minidisk eventually becomes the 190 disk for the backlevel CMS. The 590 minidisk can reside on any DASD volume.
- In this example, 72 cylinders are reserved for the 590 disk starting at cylinder 125. You recorded the number of cylinders needed by your old 190 disk in Table 30 on page 91.
  - If you are using the DDR method later in the procedure, you must make your backlevel CMS minidisk on z/VM 3.1.0 exactly the same size as your old CMS 190 minidisk.
  - If you are using the VMFPLC2 method, the backlevel CMS minidisk can be the same size or bigger than your old CMS 190 minidisk.
- Once you know how many cylinders you need for the 590 disk, you have to figure out where to put the minidisk in your z/VM 3.1.0 configuration. Use the DISKMAP command to find out where to put the backlevel CMS minidisk. In the previous MDISK statement, the starting cylinder is 125. This was obtained by entering:

```
diskmap user direct
```

For more information on DISKMAP, see the *z/VM: CP Command and Utility Reference*.

The output file from this command is USER DISKMAP. This file shows you the space you have available. If you cannot find enough contiguous space, you may have to shift other minidisk locations and data around to make room for the backlevel CMS.

**Note:** If you move a minidisk location, make sure you also move the data on that minidisk. If you decide to shift locations of minidisks used for the Shared File System, there may be additional considerations. See the *z/VM: CMS File Pool Planning, Administration, and Operation* book for details on replacing the SFS and CRR file pool minidisks.

3. Put the directory with the new statement online by issuing:

```
directxa user direct
```
4. Make sure your MAINT user ID is running an XA virtual machine. To check the virtual machine mode, you can use the QUERY SET command. You can use the SET MACHINE XA command or the MACHINE directory control statement to set MAINT's virtual machine to XA.

5. Link the minidisk where the backlevel CMS will reside by entering:  

```
link maint 590 590 MR
```
6. Format the 590 disk and access it using an unused file mode by entering:  

```
format 590 k
DMSFOR603R Format will erase all files on disk k(590).
Do you wish to continue? Enter 1 (YES) or 0 (NO).
1
DMSFOR605R Enter disk label:
CMSOLD
```

### Note to VMFPLC2 Users

If you plan to use the VMFPLC2 method to copy files to z/VM 3.1.0, skip to step 10 on page 94.

If you are using the DDR method, continue with step 7.

7. Copy the contents of your old 190 minidisk to the backlevel CMS minidisk on z/VM 3.1.0 590. The following steps show how to DDR the old 190 contents to the z/VM 3.1.0 590 minidisk.
  - a. Log on to your old-system MAINT user ID.
  - b. Attach a tape to the MAINT user ID on your old system:  

```
attach 494 to maint as 181
```
  - c. Dump the old-system files:  

```
ddr
sysprint cons
input 190 3380
output 181 3480
dump all
```
  - d. Log on to your z/VM 3.1.0 MAINT user ID.
  - e. Attach the tape to the MAINT user ID on the z/VM 3.1.0 system:  

```
attach 494 to maint as 181
```
  - f. Load the files off the tape onto the z/VM 3.1.0 590 minidisk:  

```
ddr
sysprint cons
input 181 3480
output 590 3380
restore all
```

Now you have the old VM system 190 CMS files on your z/VM 3.1.0 590 minidisk.

8. Define the 190 minidisk that contains z/VM 3.1.0 CMS with another address; for example:  

```
define 190 as 90
```
9. Define the 590 minidisk with the old-system CMS on it as your 190; for example:  

```
def 590 190
```

### Note to DDR Users

DDR users can now skip to step 17 on page 95.

If you are using the VMFPLC2 method, continue with step 10.



## Migration Tasks

10. While you are still on your z/VM 3.1.0 MAINT user ID, use the FORMAT command with the RECOMP option to change the available cylinders on 590 for disk-resident files. In the previous example, 66 cylinders needed to be recomped. (See Table 30 on page 91.) Remember, this leaves six cylinders for the backlevel CMS nucleus. Use the following FORMAT command:

```
format 590 k 66 (recomp
```

The response from this command would be:

LABEL	VDEV	M	STAT	CYL	TYPE	BLKSIZE	FILES	BLKS USED-(%)	BLKS LEFT	BLK TOTAL
CMSOLD	590	K	R/W	66	3390	4096	0	6-0	4794	4800

Ready;

11. Copy the contents of the old-system CMS on your 190 minidisk to the backlevel CMS minidisk on your z/VM 3.1.0 590. The following steps show how to move the old-system 190 contents to the z/VM 3.1.0 590 minidisk using VMFPLC2.
  - a. Log on to your old-system MAINT user ID.
  - b. Create a copy of the CMS nucleus and put it in MAINT's reader by entering:

```
spool punch *  
spool printer *  
vmfload cmsload dmsvm
```

These commands put the load deck in MAINT's reader.
  - c. Receive the load deck as a CMS file on your 191 disk. Make sure you record the file name and file type. The example described here uses the file name CMSNUC FILE A.
  - d. Attach a tape to the MAINT user ID on your old system:

```
attach 494 to maint as 181
```
  - e. Access the old-system 190 disk as something other than S. If you access it as S, only files with a file mode of S2 are accessible.

```
access 190 k
```
  - f. Dump the old-system 190 files and the new copy of the CMS nucleus on the A-disk to tape and write two tape marks:

```
vmfplc2 dump * * k  
vmfplc2 wtm  
vmfplc2 dump cmsnuc file a  
vmfplc2 wtm 2  
detach 181
```
  - g. Log on to your z/VM 3.1.0 MAINT user ID.
  - h. Attach the tape to the MAINT user ID on your z/VM 3.1.0 system:

```
attach 494 to maint as 181
```
  - i. Access the z/VM 3.1.0 590 disk and load the old-system 190 files off the tape and onto the 590 minidisk:

```
access 590 k  
vmfplc2 load * * k
```

Now you have the old-system CMS files on your 590 minidisk.
  - j. Load the CMSNUC FILE on to your 191 A disk:

```
vmfplc2 load cmsnuc file a
```
  - k. Detach the tape:

```
detach 181
```
12. Build your backlevel CMS nucleus on your z/VM 3.1.0 system.



- a. Enter the following commands:

```
spool punch *
spool printer *
spool reader hold
```

- b. Punch the file containing the CMS load deck, CMSNUC FILE, to the MAINT user ID. Make sure you use the NOHEADER option so a header is not included in the file.

```
punch cmsnuc file a (noh
```

An example of a response from the previous PUNCH command is:

```
RDR FILE 0002 SENT FROM MAINT    PUN WAS 0002 RECS 025K CPY 001 A NOHOLD NOKEEP
```

You need the file number of the CMS nucleus for the next step. The file number in this example is 0002.

- c. If the CMS nucleus is not the first file in your reader, order your reader so that the CMS nucleus is the first file processed; for example:

```
order rdr 0002
```

13. Define the 190 minidisk that contains z/VM 3.1.0 CMS with another address; for example:

```
define 190 as 90
```

14. Define the 590 minidisk with the old-system CMS on it as your 190. For example:

```
def 590 190
```

15. IPL the load deck from MAINT's reader:

```
ipl 00c c1
```

16. When you are sure you are done with the CMSNUC FILE, you may want to erase it; it can be rather big. Keep the tape with the old-system 190 files and the CMSNUC FILE for backup.

**Note to Both DDR and VMFPLC2 Users**

Use the following steps to put CMS into a named saved system.

17. Determine the DEFSYS command to use to define CMSOLD as a named saved system. You can probably just use your old system's DEFSYS command. You may want to add this DEFSYS command to an exec where you define all the other saved segments and saved systems. The sample exec in Figure 6 on page 96 defines various CMS named saved systems.

18. Issue the DEFSYS command or use the sample exec in Figure 6 on page 96 to define the CMSOLD named saved system; for example:

```
defnss cmsold
```

19. Use the QUERY NSS MAP command to check that CMSOLD exists.

20. IPL the 190 minidisk with CMSOLD on it:

```
ipl 190 parm savesys cmsold
```

## Defining Various Levels of CMS

Use the following sample exec to define different levels of CMS in a named saved system.

## Migration Tasks

```
/******  
/*  
/* An EXEC that will define the various levels of CMS          */  
/* NSSs for our new system.                                   */  
/*  
/* You can pass in the following arguments:                   */  
/* o CMSNEW - for the new CMS                                */  
/* o CMSOLD  - for the old CMS                               */  
/* o ALL    - for both the old CMS and the new CMS          */  
/*  
/******  
  
Arg parms  
  
allnss = 'CMSNEW CMSOLD'  
  
/* DEFSYS command to define the new CMS */  
nss.1 ="CP DEFSYS CMSNEW  0-D EW 20-23 EW F00-13FF SR MINSIZE=256K ",  
      "MACHMODE XA,XC,ESA PARMREGS=0-15"  
  
/* DEFSYS command to define an old CMS */  
nss.2 ="CP DEFSYS CMSOLD  0-D EW 20-23 EW F00-12FF SR MINSIZE=256K ",  
      "MACHMODE XA,XC"  
  
addrcmd = 'Address Command '  
  
/* If DEFNSS is called with 'ALL' or blank then */  
/* define both the new CMS and the backlevel CMS */  
If parms = '' | parms = 'ALL'  
  Then Do  
    Do i = 1 to WORDS(allnss)  
      Interpret addrCmd nss.i  
    End  
  End  
End
```

Figure 6. DEFNSS EXEC (Part 1 of 2)

```

/* Else determine the number of parameters passed */
/* in. Make sure the parameters are named saved */
/* systems defined as either nss.1 or nss.2 */
/* (that is, CMSNEW or CMSOLD). If a parameter is */
/* not CMSNEW or CMSOLD then issue an error message. */
Else Do
  badnss = ''
  errors = 'NO'
  Do i = 1 to WORDS(parms)
    nssno = FIND(allnss,WORD(parms,i))
    If nssno = 0
      Then Do
        Errors = 'YES'
        badnss = badnss ' ' WORD(parms,i)
      End
    /* If a valid named saved system is passed in */
    /* then issue the appropriate DEFSYS command. */
    Else
      Interpret addrCmd nss.nssno
    End
  End
  /* If the named saved system is not CMSNEW or CMSOLD */
  /* then issue an error message. */
  If errors = 'YES' Then
    Signal Errinv

Exit 0

Errinv:

  Say "The following NSSs were not defined because a definition has"
  Say " not been set up for the NSS': " badnss
  Say ''
  Say "A list of all valid NSS' follows:"
  Say ' ' allnss
Exit 99

```

Figure 6. DEFNSS EXEC (Part 2 of 2)

## Swapping Between the Backlevel CMS and z/VM 3.1.0 CMS

If you want to be able to switch between the z/VM 3.1.0 CMS and the backlevel CMS, consider using the following execs that link to the different system disks. You can make these execs available to your users. The users then need to do a SET MACH XA or SET MACH 370 and IPL the corresponding CMS segment (IPL CMSNEW or IPL CMSOLD).

```

/* This exec swaps a backlevel CMS with a */
/* new CMS. */

'EXECIO 0 CP (STRING DETACH 190'
'CP LINK MAINT 190 190 RR'

```

Figure 7. SWAP2NEW EXEC

The user should then do a SET MACH XA and IPL CMSNEW.

## Migration Tasks

```
/* This exec swaps a new CMS with a      */
/* backlevel CMS.                        */

      'EXECIO 0 CP (STRING DETACH 190'
      'CP LINK MAINT 590 190 RR'
```

Figure 8. SWAP2OLD EXEC

The user should then issue:

```
set mach xa
ipl cmsold
```

---

## Migrating REXX and EXEC 2 Application Programs

The REXX/EXEC Migration Tool for VM/ESA (VM/ESA MIGR) helps you convert your REXX and EXEC 2 source files. The areas where VM/ESA MIGR can give you assistance are:

- Estimating the conversion effort necessary.
- Identifying changes that have to be made in your programs.
- Applying the necessary changes in the program. VM/ESA MIGR only assists you by pointing out the changes needed; it does not change the files. VM/ESA MIGR creates an interactive environment that assists you in finding and making changes to your execs due to incompatibilities or changes in commands and options. VM/ESA MIGR can also present you with information about these changes through Help panels.

See the *VM/ESA: REXX/EXEC Migration Tool for VM/ESA* book for information on using this tool.

---

## Appendix A. CMS Pipelines Message Cross-Reference [2.3.0]

All CMS Pipelines messages have been renamed and renumbered from a DMS prefix to an FPL prefix. All of the FPL message numbers are consistent with those from CMS/TSO Pipelines.

This appendix contains two message cross-references: one from DMS prefix to FPL prefix, the other from FPL prefix to DMS prefix. Many messages also have different text, but those changes are not identified here. See the descriptions of individual FPL messages in *z/VM: System Messages and Codes*.

**Note:** FPL messages not included in these cross-references have no DMS prefix equivalent.

---

### DMS to FPL Message Cross-Reference

The following is a cross-reference from the old DMS message numbers to the corresponding new FPL message numbers:

DMS2571E-FPL337E	DMS2608E-FPL654E	DMS2646I-FPL717I	DMS2685E-FPL040E
DMS2572E-FPL338E	DMS2609E-FPL655E	DMS2647I-FPL718I	DMS2686E-FPL041E
DMS2573E-FPL339E	DMS2610E-FPL656E	DMS2648I-FPL719I	DMS2687E-FPL042E
DMS2574I-FPL340I	DMS2611E-FPL657E	DMS2649I-FPL720I	DMS2688E-FPL043E
DMS2575I-FPL341I	DMS2612E-FPL660E	DMS2650E-FPL000E	DMS2689E-FPL044E
DMS2576I-FPL342I	DMS2613E-FPL662E	DMS2651I-FPL001I	DMS2690W-FPL045W
DMS2577E-FPL343E	DMS2614E-FPL664E	DMS2652I-FPL002I	DMS2691E-FPL046E
DMS2578I-FPL344I	DMS2615E-FPL1032E	DMS2653I-FPL003I	DMS2692E-FPL047E
DMS2579E-FPL345E	DMS2616E-FPL1033E	DMS2654I-FPL004I	DMS2693E-FPL048E
DMS2580E-FPL346E	DMS2618E-FPL1036E	DMS2655E-FPL010E	DMS2694E-FPL049E
DMS2581E-FPL347E	DMS2619E-FPL686E	DMS2656E-FPL011E	DMS2695E-FPL050E
DMS2582I-FPL348I	DMS2620E-FPL1038E	DMS2657E-FPL012E	DMS2696E-FPL051E
DMS2583E-FPL1015E	DMS2621E-FPL1039E	DMS2658I-FPL721I	DMS2697E-FPL052E
DMS2584E-FPL573E	DMS2622E-FPL1040E	DMS2659E-FPL014E	DMS2698E-FPL053E
DMS2585E-FPL575E	DMS2623E-FPL1041E	DMS2660E-FPL015E	DMS2699E-FPL054E
DMS2586I-FPL369I	DMS2624E-FPL1049E	DMS2661E-FPL016E	DMS2700I-FPL1111I
DMS2587E-FPL576E	DMS2625E-FPL680E	DMS2662E-FPL017E	DMS2701E-FPL056E
DMS2588E-FPL611E	DMS2626E-FPL663E	DMS2663E-FPL018E	DMS2702I-FPL725I
DMS2589I-FPL612I	DMS2627E-FPL694E	DMS2664W-FPL019W	DMS2703E-FPL058E
DMS2590E-FPL613E	DMS2628E-FPL1124E	DMS2665I-FPL020I	DMS2704E-FPL059E
DMS2591E-FPL614E	DMS2629E-FPL700E	DMS2666E-FPL021E	DMS2705E-FPL060E
DMS2592E-FPL615E	DMS2630E-FPL701E	DMS2668E-FPL023E	DMS2706E-FPL061E
DMS2593E-FPL616E	DMS2631I-FPL702I	DMS2669W-FPL024W	DMS2707E-FPL062E
DMS2594E-FPL617E	DMS2632I-FPL703I	DMS2670E-FPL1100E	DMS2708E-FPL063E
DMS2595W-FPL620W	DMS2633E-FPL704E	DMS2672E-FPL027E	DMS2709E-FPL064E
DMS2596E-FPL627E	DMS2634E-FPL705E	DMS2673I-FPL028I	DMS2710E-FPL065E
DMS2597E-FPL635E	DMS2635E-FPL706E	DMS2674E-FPL029E	DMS2711E-FPL066E
DMS2598E-FPL1019E	DMS2636E-FPL707E	DMS2675I-FPL030I	DMS2712E-FPL067E
DMS2599E-FPL371E	DMS2637E-FPL708E	DMS2676I-FPL031I	DMS2713E-FPL068E
DMS2600W-FPL564W	DMS2638E-FPL709E	DMS2677I-FPL1110I	DMS2714E-FPL069E
DMS2601E-FPL685E	DMS2639E-FPL710E	DMS2678I-FPL033I	DMS2715E-FPL070E
DMS2602E-FPL639E	DMS2640E-FPL711E	DMS2679I-FPL034I	DMS2716E-FPL071E
DMS2603E-FPL642E	DMS2641E-FPL712E	DMS2680I-FPL035I	DMS2717E-FPL072E
DMS2604E-FPL307E	DMS2642E-FPL713E	DMS2681I-FPL036I	DMS2718E-FPL073E
DMS2605E-FPL651E	DMS2643E-FPL714E	DMS2682I-FPL037I	DMS2719E-FPL074E
DMS2606E-FPL652E	DMS2644E-FPL715E	DMS2683I-FPL038I	DMS2720E-FPL075E
DMS2607E-FPL653E	DMS2645E-FPL716E	DMS2684I-FPL039I	DMS2721I-FPL076I

## Pipelines Message Cross-Reference

DMS2722I-FPL077I	DMS2782E-FPL139E	DMS2841E-FPL219E	DMS2901E-FPL352E
DMS2723E-FPL078E	DMS2783E-FPL140E	DMS2842E-FPL220E	DMS2902E-FPL354E
DMS2724E-FPL079E	DMS2784E-FPL141E	DMS2843E-FPL222E	DMS2903I-FPL355I
DMS2725E-FPL080E	DMS2785E-FPL142E	DMS2844E-FPL223E	DMS2904I-FPL356I
DMS2726E-FPL081E	DMS2786E-FPL143E	DMS2845E-FPL224E	DMS2905E-FPL357E
DMS2727E-FPL082E	DMS2787E-FPL144E	DMS2846E-FPL225E	DMS2906E-FPL358E
DMS2728E-FPL083E	DMS2788I-FPL145I	DMS2847E-FPL226E	DMS2907E-FPL359E
DMS2729E-FPL084E	DMS2789E-FPL146E	DMS2848E-FPL227E	DMS2908E-FPL360E
DMS2730E-FPL085E	DMS2790E-FPL147E	DMS2849E-FPL229E	DMS2909I-FPL361I
DMS2731I-FPL086I	DMS2791E-FPL148E	DMS2850E-FPL230E	DMS2910E-FPL362E
DMS2732E-FPL087E	DMS2792E-FPL150E	DMS2851E-FPL231E	DMS2911E-FPL363E
DMS2733E-FPL088E	DMS2793E-FPL151E	DMS2852E-FPL232E	DMS2912E-FPL364E
DMS2734E-FPL089E	DMS2794E-FPL152E	DMS2853E-FPL233E	DMS2913E-FPL365E
DMS2735E-FPL090E	DMS2795E-FPL154E	DMS2854E-FPL234E	DMS2914E-FPL366E
DMS2736E-FPL091E	DMS2795E-FPL732E	DMS2855E-FPL235E	DMS2915E-FPL741E
DMS2737E-FPL092E	DMS2796E-FPL155E	DMS2856E-FPL236E	DMS2916E-FPL368E
DMS2738E-FPL093E	DMS2797E-FPL156E	DMS2857E-FPL237E	DMS2917E-FPL742E
DMS2739E-FPL094E	DMS2798E-FPL157E	DMS2858E-FPL238E	DMS2918E-FPL370E
DMS2740E-FPL095E	DMS2799E-FPL159E	DMS2859E-FPL241E	DMS2919I-FPL743I
DMS2741E-FPL096E	DMS2800E-FPL161E	DMS2860W-FPL245W	DMS2920I-FPL744I
DMS2742E-FPL097E	DMS2801E-FPL162E	DMS2861E-FPL253E	DMS2921E-FPL373E
DMS2743E-FPL098E	DMS2802E-FPL163E	DMS2862I-FPL256I	DMS2922E-FPL380E
DMS2744E-FPL099E	DMS2803E-FPL164E	DMS2863E-FPL257E	DMS2925E-FPL391E
DMS2745E-FPL100E	DMS2804E-FPL165E	DMS2864E-FPL261E	DMS2926E-FPL392E
DMS2746E-FPL101E	DMS2805E-FPL166E	DMS2865E-FPL264E	DMS2927E-FPL393E
DMS2747E-FPL102E	DMS2806E-FPL733E	DMS2866E-FPL279E	DMS2928E-FPL745E
DMS2748E-FPL103E	DMS2807E-FPL169E	DMS2867E-FPL280E	DMS2929E-FPL400E
DMS2749E-FPL104E	DMS2808E-FPL170E	DMS2868W-FPL281W	DMS2930E-FPL401E
DMS2750E-FPL105E	DMS2809E-FPL172E	DMS2869E-FPL282E	DMS2931I-FPL402I
DMS2751E-FPL107E	DMS2810E-FPL173E	DMS2870W-FPL283W	DMS2932E-FPL746E
DMS2752E-FPL108E	DMS2811E-FPL174E	DMS2871E-FPL284E	DMS2933E-FPL747E
DMS2753E-FPL109E	DMS2812E-FPL734E	DMS2871E-FPL737E	DMS2934E-FPL405E
DMS2754E-FPL110E	DMS2813E-FPL735E	DMS2872E-FPL287E	DMS2934E-FPL748E
DMS2755E-FPL111E	DMS2814I-FPL177I	DMS2873E-FPL738E	DMS2935E-FPL406E
DMS2756E-FPL112E	DMS2815E-FPL178E	DMS2874E-FPL289E	DMS2936E-FPL407E
DMS2757E-FPL113E	DMS2816E-FPL179E	DMS2875E-FPL290E	DMS2937I-FPL412I
DMS2758E-FPL114E	DMS2817E-FPL180E	DMS2876E-FPL291E	DMS2938E-FPL409E
DMS2759E-FPL115E	DMS2818E-FPL181E	DMS2877E-FPL292E	DMS2939E-FPL410E
DMS2760E-FPL116E	DMS2819W-FPL182W	DMS2878I-FPL293I	DMS2940I-FPL411I
DMS2761E-FPL117E	DMS2820E-FPL183E	DMS2879E-FPL297E	DMS2941E-FPL420E
DMS2762E-FPL118E	DMS2821E-FPL184E	DMS2880I-FPL298I	DMS2942I-FPL413I
DMS2763E-FPL119E	DMS2822E-FPL185E	DMS2881E-FPL740E	DMS2943E-FPL749E
DMS2764E-FPL120E	DMS2823I-FPL186I	DMS2882E-FPL301E	DMS2944E-FPL750E
DMS2765I-FPL726I	DMS2824E-FPL187E	DMS2883E-FPL302E	DMS2946E-FPL752E
DMS2766E-FPL122E	DMS2825I-FPL189I	DMS2884E-FPL303E	DMS2947E-FPL753E
DMS2767I-FPL727I	DMS2826E-FPL190E	DMS2885E-FPL304E	DMS2948E-FPL780E
DMS2768E-FPL124E	DMS2827E-FPL191E	DMS2886E-FPL305E	DMS2949E-FPL509E
DMS2769E-FPL125E	DMS2828I-FPL192I	DMS2887E-FPL308E	DMS2950E-FPL510E
DMS2770E-FPL126E	DMS2829E-FPL193E	DMS2888E-FPL309E	DMS2951E-FPL511E
DMS2771E-FPL127E	DMS2830E-FPL194E	DMS2889E-FPL310E	DMS2952E-FPL512E
DMS2772E-FPL128E	DMS2831E-FPL195E	DMS2890E-FPL311E	DMS2953E-FPL513E
DMS2773E-FPL129E	DMS2832E-FPL196E	DMS2891E-FPL313E	DMS2954E-FPL514E
DMS2774E-FPL131E	DMS2833E-FPL197E	DMS2892E-FPL314E	DMS2955E-FPL515E
DMS2775E-FPL132E	DMS2834E-FPL198E	DMS2893E-FPL315E	DMS2956E-FPL516E
DMS2776E-FPL133E	DMS2835E-FPL209E	DMS2894E-FPL317E	DMS2957E-FPL517E
DMS2777E-FPL134E	DMS2836E-FPL211E	DMS2895E-FPL318E	DMS2958E-FPL518E
DMS2778I-FPL728I	DMS2837E-FPL212E	DMS2896E-FPL319E	DMS2959E-FPL530E
DMS2779I-FPL729I	DMS2838E-FPL214E	DMS2898E-FPL333E	DMS2960E-FPL531E
DMS2780E-FPL137E	DMS2839E-FPL215E	DMS2899E-FPL334E	DMS2961E-FPL532E
DMS2781E-FPL138E	DMS2840E-FPL736E	DMS2900E-FPL350E	DMS2962E-FPL533E

## Pipelines Message Cross-Reference

DMS2963E-FPL534E	DMS2971E-FPL542E	DMS2980I-FPL552I	DMS2989E-FPL561E
DMS2964E-FPL535E	DMS2972E-FPL543E	DMS2981I-FPL1113I	DMS2990E-FPL562E
DMS2965E-FPL536E	DMS2973I-FPL544I	DMS2982E-FPL554E	DMS2991E-FPL306E
DMS2966I-FPL537I	DMS2974E-FPL545E	DMS2983I-FPL555I	DMS2992E-FPL650E
DMS2967I-FPL538I	DMS2975E-FPL546E	DMS2984E-FPL556E	DMS2993E-FPL569E
DMS2968E-FPL539E	DMS2976E-FPL547E	DMS2986I-FPL1114I	DMS2994E-FPL571E
DMS2969E-FPL540E	DMS2977I-FPL548I	DMS2987I-FPL1115I	DMS2998E-FPL335E
DMS2970E-FPL541E	DMS2979I-FPL1112I	DMS2988I-FPL560I	DMS2999E-FPL336E

---

## FPL to DMS Message Cross-Reference

The following is a cross-reference from the new FPL message numbers to the corresponding old DMS message numbers:

FPL000E-DMS2650E	FPL052E-DMS2697E	FPL096E-DMS2741E	FPL144E-DMS2787E
FPL001I-DMS2651I	FPL053E-DMS2698E	FPL097E-DMS2742E	FPL145I-DMS2788I
FPL002I-DMS2652I	FPL054E-DMS2699E	FPL098E-DMS2743E	FPL146E-DMS2789E
FPL003I-DMS2653I	FPL056E-DMS2701E	FPL099E-DMS2744E	FPL147E-DMS2790E
FPL004I-DMS2654I	FPL058E-DMS2703E	FPL100E-DMS2745E	FPL148E-DMS2791E
FPL010E-DMS2655E	FPL059E-DMS2704E	FPL101E-DMS2746E	FPL150E-DMS2792E
FPL011E-DMS2656E	FPL060E-DMS2705E	FPL102E-DMS2747E	FPL151E-DMS2793E
FPL012E-DMS2657E	FPL061E-DMS2706E	FPL103E-DMS2748E	FPL152E-DMS2794E
FPL014E-DMS2659E	FPL062E-DMS2707E	FPL104E-DMS2749E	FPL154E-DMS2795E
FPL015E-DMS2660E	FPL063E-DMS2708E	FPL105E-DMS2750E	FPL155E-DMS2796E
FPL016E-DMS2661E	FPL064E-DMS2709E	FPL107E-DMS2751E	FPL156E-DMS2797E
FPL017E-DMS2662E	FPL065E-DMS2710E	FPL108E-DMS2752E	FPL157E-DMS2798E
FPL018E-DMS2663E	FPL066E-DMS2711E	FPL109E-DMS2753E	FPL159E-DMS2799E
FPL019W-DMS2664W	FPL067E-DMS2712E	FPL110E-DMS2754E	FPL161E-DMS2800E
FPL020I-DMS2665I	FPL068E-DMS2713E	FPL111E-DMS2755E	FPL162E-DMS2801E
FPL021E-DMS2666E	FPL069E-DMS2714E	FPL112E-DMS2756E	FPL163E-DMS2802E
FPL023E-DMS2668E	FPL070E-DMS2715E	FPL113E-DMS2757E	FPL164E-DMS2803E
FPL024W-DMS2669W	FPL071E-DMS2716E	FPL114E-DMS2758E	FPL165E-DMS2804E
FPL027E-DMS2672E	FPL072E-DMS2717E	FPL115E-DMS2759E	FPL166E-DMS2805E
FPL028I-DMS2673I	FPL073E-DMS2718E	FPL116E-DMS2760E	FPL169E-DMS2807E
FPL029E-DMS2674E	FPL074E-DMS2719E	FPL117E-DMS2761E	FPL170E-DMS2808E
FPL030I-DMS2675I	FPL075E-DMS2720E	FPL118E-DMS2762E	FPL172E-DMS2809E
FPL031I-DMS2676I	FPL076I-DMS2721I	FPL119E-DMS2763E	FPL173E-DMS2810E
FPL033I-DMS2678I	FPL077I-DMS2722I	FPL120E-DMS2764E	FPL174E-DMS2811E
FPL034I-DMS2679I	FPL078E-DMS2723E	FPL122E-DMS2766E	FPL177I-DMS2814I
FPL035I-DMS2680I	FPL079E-DMS2724E	FPL124E-DMS2768E	FPL178E-DMS2815E
FPL036I-DMS2681I	FPL080E-DMS2725E	FPL125E-DMS2769E	FPL179E-DMS2816E
FPL037I-DMS2682I	FPL081E-DMS2726E	FPL126E-DMS2770E	FPL180E-DMS2817E
FPL038I-DMS2683I	FPL082E-DMS2727E	FPL127E-DMS2771E	FPL181E-DMS2818E
FPL039I-DMS2684I	FPL083E-DMS2728E	FPL128E-DMS2772E	FPL182W-DMS2819W
FPL040E-DMS2685E	FPL084E-DMS2729E	FPL129E-DMS2773E	FPL183E-DMS2820E
FPL041E-DMS2686E	FPL085E-DMS2730E	FPL131E-DMS2774E	FPL184E-DMS2821E
FPL042E-DMS2687E	FPL086I-DMS2731I	FPL132E-DMS2775E	FPL185E-DMS2822E
FPL043E-DMS2688E	FPL087E-DMS2732E	FPL133E-DMS2776E	FPL186I-DMS2823I
FPL044E-DMS2689E	FPL088E-DMS2733E	FPL134E-DMS2777E	FPL187E-DMS2824E
FPL045W-DMS2690W	FPL089E-DMS2734E	FPL137E-DMS2780E	FPL189I-DMS2825I
FPL046E-DMS2691E	FPL090E-DMS2735E	FPL138E-DMS2781E	FPL190E-DMS2826E
FPL047E-DMS2692E	FPL091E-DMS2736E	FPL139E-DMS2782E	FPL191E-DMS2827E
FPL048E-DMS2693E	FPL092E-DMS2737E	FPL140E-DMS2783E	FPL192I-DMS2828I
FPL049E-DMS2694E	FPL093E-DMS2738E	FPL141E-DMS2784E	FPL193E-DMS2829E
FPL050E-DMS2695E	FPL094E-DMS2739E	FPL142E-DMS2785E	FPL194E-DMS2830E
FPL051E-DMS2696E	FPL095E-DMS2740E	FPL143E-DMS2786E	FPL195E-DMS2831E



## Pipelines Message Cross-Reference

FPL196E-DMS2832E	FPL315E-DMS2893E	FPL513E-DMS2953E	FPL663E-DMS2626E
FPL197E-DMS2833E	FPL317E-DMS2894E	FPL514E-DMS2954E	FPL664E-DMS2614E
FPL198E-DMS2834E	FPL318E-DMS2895E	FPL515E-DMS2955E	FPL680E-DMS2625E
FPL209E-DMS2835E	FPL319E-DMS2896E	FPL516E-DMS2956E	FPL685E-DMS2601E
FPL211E-DMS2836E	FPL333E-DMS2898E	FPL517E-DMS2957E	FPL686E-DMS2619E
FPL212E-DMS2837E	FPL334E-DMS2899E	FPL518E-DMS2958E	FPL694E-DMS2627E
FPL214E-DMS2838E	FPL335E-DMS2998E	FPL530E-DMS2959E	FPL700E-DMS2629E
FPL215E-DMS2839E	FPL336E-DMS2999E	FPL531E-DMS2960E	FPL701E-DMS2630E
FPL219E-DMS2841E	FPL337E-DMS2571E	FPL532E-DMS2961E	FPL702I-DMS2631I
FPL220E-DMS2842E	FPL338E-DMS2572E	FPL533E-DMS2962E	FPL703I-DMS2632I
FPL222E-DMS2843E	FPL339E-DMS2573E	FPL534E-DMS2963E	FPL704E-DMS2633E
FPL223E-DMS2844E	FPL340I-DMS2574I	FPL535E-DMS2964E	FPL705E-DMS2634E
FPL224E-DMS2845E	FPL341I-DMS2575I	FPL536E-DMS2965E	FPL706E-DMS2635E
FPL225E-DMS2846E	FPL342I-DMS2576I	FPL537I-DMS2966I	FPL707E-DMS2636E
FPL226E-DMS2847E	FPL343E-DMS2577E	FPL538I-DMS2967I	FPL708E-DMS2637E
FPL227E-DMS2848E	FPL344I-DMS2578I	FPL539E-DMS2968E	FPL709E-DMS2638E
FPL229E-DMS2849E	FPL345E-DMS2579E	FPL540E-DMS2969E	FPL710E-DMS2639E
FPL230E-DMS2850E	FPL346E-DMS2580E	FPL541E-DMS2970E	FPL711E-DMS2640E
FPL231E-DMS2851E	FPL347E-DMS2581E	FPL542E-DMS2971E	FPL712E-DMS2641E
FPL232E-DMS2852E	FPL348I-DMS2582I	FPL543E-DMS2972E	FPL713E-DMS2642E
FPL233E-DMS2853E	FPL350E-DMS2900E	FPL544I-DMS2973I	FPL714E-DMS2643E
FPL234E-DMS2854E	FPL352E-DMS2901E	FPL545E-DMS2974E	FPL715E-DMS2644E
FPL235E-DMS2855E	FPL354E-DMS2902E	FPL546E-DMS2975E	FPL716E-DMS2645E
FPL236E-DMS2856E	FPL355I-DMS2903I	FPL547E-DMS2976E	FPL717I-DMS2646I
FPL237E-DMS2857E	FPL356I-DMS2904I	FPL548I-DMS2977I	FPL718I-DMS2647I
FPL238E-DMS2858E	FPL357E-DMS2905E	FPL552I-DMS2980I	FPL719I-DMS2648I
FPL241E-DMS2859E	FPL358E-DMS2906E	FPL554E-DMS2982E	FPL720I-DMS2649I
FPL245W-DMS2860W	FPL359E-DMS2907E	FPL555I-DMS2983I	FPL721I-DMS2658I
FPL253E-DMS2861E	FPL360E-DMS2908E	FPL556E-DMS2984E	FPL725I-DMS2702I
FPL256I-DMS2862I	FPL361I-DMS2909I	FPL560I-DMS2988I	FPL726I-DMS2765I
FPL257E-DMS2863E	FPL362E-DMS2910E	FPL561E-DMS2989E	FPL727I-DMS2767I
FPL261E-DMS2864E	FPL363E-DMS2911E	FPL562E-DMS2990E	FPL728I-DMS2778I
FPL264E-DMS2865E	FPL364E-DMS2912E	FPL564W-DMS2600W	FPL729I-DMS2779I
FPL279E-DMS2866E	FPL365E-DMS2913E	FPL569E-DMS2993E	FPL732E-DMS2795E
FPL280E-DMS2867E	FPL366E-DMS2914E	FPL571E-DMS2994E	FPL733E-DMS2806E
FPL281W-DMS2868W	FPL368E-DMS2916E	FPL573E-DMS2584E	FPL734E-DMS2812E
FPL282E-DMS2869E	FPL369I-DMS2586I	FPL575E-DMS2585E	FPL735E-DMS2813E
FPL283W-DMS2870W	FPL370E-DMS2918E	FPL576E-DMS2587E	FPL736E-DMS2840E
FPL284E-DMS2871E	FPL371E-DMS2599E	FPL611E-DMS2588E	FPL737E-DMS2871E
FPL287E-DMS2872E	FPL373E-DMS2921E	FPL612I-DMS2589I	FPL738E-DMS2873E
FPL289E-DMS2874E	FPL380E-DMS2922E	FPL613E-DMS2590E	FPL740E-DMS2881E
FPL290E-DMS2875E	FPL391E-DMS2925E	FPL614E-DMS2591E	FPL741E-DMS2915E
FPL291E-DMS2876E	FPL392E-DMS2926E	FPL615E-DMS2592E	FPL742E-DMS2917E
FPL292E-DMS2877E	FPL393E-DMS2927E	FPL616E-DMS2593E	FPL743I-DMS2919I
FPL293I-DMS2878I	FPL400E-DMS2929E	FPL617E-DMS2594E	FPL744I-DMS2920I
FPL297E-DMS2879E	FPL401E-DMS2930E	FPL620W-DMS2595W	FPL745E-DMS2928E
FPL298I-DMS2880I	FPL402I-DMS2931I	FPL627E-DMS2596E	FPL746E-DMS2932E
FPL301E-DMS2882E	FPL405E-DMS2934E	FPL635E-DMS2597E	FPL747E-DMS2933E
FPL302E-DMS2883E	FPL406E-DMS2935E	FPL639E-DMS2602E	FPL748E-DMS2934E
FPL303E-DMS2884E	FPL407E-DMS2936E	FPL642E-DMS2603E	FPL749E-DMS2943E
FPL304E-DMS2885E	FPL409E-DMS2938E	FPL650E-DMS2992E	FPL750E-DMS2944E
FPL305E-DMS2886E	FPL410E-DMS2939E	FPL651E-DMS2605E	FPL752E-DMS2946E
FPL306E-DMS2991E	FPL411I-DMS2940I	FPL652E-DMS2606E	FPL753E-DMS2947E
FPL307E-DMS2604E	FPL412I-DMS2937I	FPL653E-DMS2607E	FPL780E-DMS2948E
FPL308E-DMS2887E	FPL413I-DMS2942I	FPL654E-DMS2608E	FPL1015E-DMS2583E
FPL309E-DMS2888E	FPL420E-DMS2941E	FPL655E-DMS2609E	FPL1019E-DMS2598E
FPL310E-DMS2889E	FPL509E-DMS2949E	FPL656E-DMS2610E	FPL1032E-DMS2615E
FPL311E-DMS2890E	FPL510E-DMS2950E	FPL657E-DMS2611E	FPL1033E-DMS2616E
FPL313E-DMS2891E	FPL511E-DMS2951E	FPL660E-DMS2612E	FPL1036E-DMS2618E
FPL314E-DMS2892E	FPL512E-DMS2952E	FPL662E-DMS2613E	FPL1038E-DMS2620E



## Pipelines Message Cross-Reference

FPL1039E-DMS2621E  
FPL1040E-DMS2622E  
FPL1041E-DMS2623E

FPL1049E-DMS2624E  
FPL1100E-DMS2670E  
FPL1110I-DMS2677I

FPL1111I-DMS2700I  
FPL1112I-DMS2979I  
FPL1113I-DMS2981I

FPL1114I-DMS2986I  
FPL1115I-DMS2987I  
FPL1124E-DMS2628E

## Pipelines Message Cross-Reference

---

## Appendix B. Sample Utilities for CP Configurability

This appendix provides reference information for the following sample utility programs supplied with z/VM:

- HCPTRIO
- HCPTSYS

These programs are shipped on the z/VM System DDR. They have a file type of SAMPEXEC and are not fully supported. To use these sample utilities, you must rename HCPTRIO SAMPEXEC and HCPTSYS SAMPEXEC to HCPTRIO EXEC and HCPTSYS EXEC, respectively.

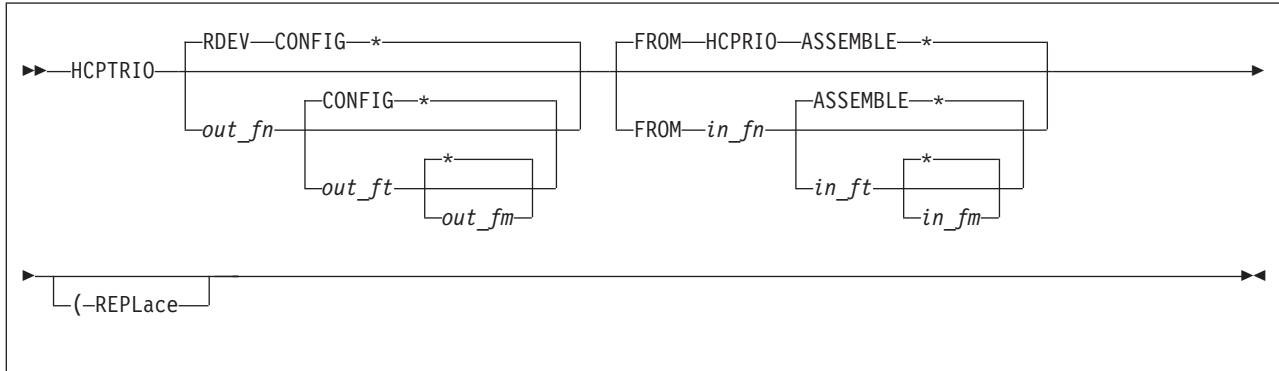
There are other utilities that you might find useful when working with the CP system configuration:

- CPSYNTAX verifies the syntax of a SYSTEM CONFIG file. CPSYNTAX is described in *z/VM: CP Command and Utility Reference*.
- SALIPL installs a copy of the Stand-Alone Loader (SAL) in cylinder zero or, in the case of FBA DASD, in blocks 5 to 31 of a disk. SALIPL is described in *z/VM: CP Command and Utility Reference*.

The SALIPL utility can run under CMS or it can be loaded by SAL to run stand alone. There is also a version of SALIPL that can be IPLed stand alone from tape or card readers. For information on using SALIPL as a stand-alone utility, refer to *z/VM: System Operation*.

- The sample utility program, HCPDCON, creates a system configuration file from a running VM system. For more information, see *z/VM: Planning and Administration*.

## HCPTRIO



### Authorization

Privilege Class: G

### Purpose

Use HCPTRIO to generate a system configuration file (SYSTEM CONFIG) from an HCPTRIO ASSEMBLE file. The generated file will contain statements for those real devices that do not answer sense ID requests or that do not return enough information.

### Operands

*out\_fn*

is the name of the CMS file to be generated. The default is RDEV.

*out\_ft*

is the type of the file to be generated. The default is CONFIG.

*out\_fm*

is the CMS file mode on which the file is to reside. The file mode must be accessed in write mode. An asterisk (\*), the default, causes the HCPTRIO to use the first file mode accessed R/W.

**FROM**

tells HCPTRIO that there is a specific source file it should use.

*in\_fn*

is the name of the CMS file that contains the input. The default is HCPTRIO.

*in\_ft*

is the type of the CMS file that contains the input. The default is ASSEMBLE.

*in\_fm*

is the file mode on which the file resides. An asterisk (\*), the default, causes HCPTRIO to use the first file mode on which the file resides.

**REPLace**

indicates that if the output file already exists, it should be replaced.

### Usage Notes

1. HCPTRIO uses a work file having a file type of "\$" followed by the first seven characters of the file type of the input file. If such a file already exists,

HCPTRIO erases it. After writing the work file, HCPTRIO erases the original configuration file and renames the work file.

Example:

```
hcprio rdev config a from hcprio assemble b (repl
```

In this case, HCPTRIO writes a work file named RDEV \$CONFIG A, erases the old RDEV CONFIG A, and renames RDEV \$CONFIG A to RDEV CONFIG A.

## Return Codes

- 0** Successful execution
- 4** Disk not accessed or not accessed R/W
- 12** Output file already exists and the REPLACE option was not specified
- 16** Input file to translate was not found
- 20** I/O error in creating output file or in reading input file
- 24** Invalid option or too many options

## Examples

### Example 1

```
hcprio cambvm3 config a from hcprio assemble d
```

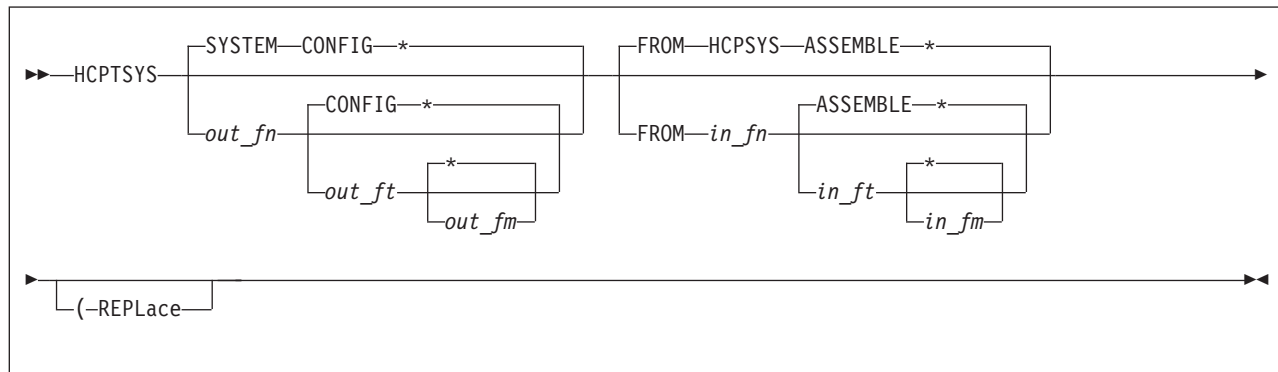
In this example, HCPTRIO uses the file HCPRIO ASSEMBLE D to generate a configuration file named CAMBVM3 CONFIG A.

### Example 2

```
hcprio
```

In this example, HCPTRIO reads the first file it finds in the CMS search order that is named HCPRIO ASSEMBLE. It creates a configuration file named RDEV CONFIG on the first file mode letter in the CMS search order that is accessed R/W.

## HCPTSYS



### Authorization

Privilege Class: G

### Purpose

Use HCPTSYS to generate a system configuration file (SYSTEM CONFIG) from an HCPSYS ASSEMBLE file.

### Operands

*out\_fn*

is the name of the CMS file to be generated. The default is SYSTEM.

*out\_ft*

is the type of the file to be generated. The default is CONFIG.

*out\_fm*

is the CMS file mode on which the file is to reside. The file mode must be accessed in write mode. An asterisk (\*), the default, causes the HCPTSYS to use the first file mode accessed R/W.

**FROM**

tells HCPTSYS that there is a specific source file it should use.

*in\_fn*

is the name of the CMS file that contains the input. The default is HCPSYS.

*in\_ft*

is the type of the CMS file that contains the input. The default is ASSEMBLE.

*in\_fm*

is the file mode on which the file resides. An asterisk (\*), the default, causes HCPTSYS to use the first file mode on which the file resides.

**REPLace**

indicates that if the output file already exists, it should be replaced.

### Usage Notes

1. HCPTSYS uses a work file having a file type of "\$" followed by the first seven characters of the file type of the input file. If such a file already exists, HCPTSYS erases it. After writing the work file, HCPTSYS erases the original configuration file and renames the work file.

Example:

```
hcptsys system config a from hcpsys assemble b (repl
```

In this case, HCPTRIO writes a work file named SYSTEM \$CONFIG A, erases the old SYSTEM CONFIG A, and renames SYSTEM \$CONFIG A to SYSTEM CONFIG A.

## Return Codes

- 0** Successful execution
- 4** Disk not accessed or not accessed R/W
- 12** Output file already exists and the REPLACE option was not specified
- 16** Input file to translate was not found
- 20** I/O error in creating output file or in reading input file
- 24** Invalid option or too many options

## Examples

### Example 1

```
hcptsys cambvm3 config a from hcpsys assemble d
```

In this example, HCPTRIO uses the file HCPSYS ASSEMBLE D to generate a configuration file named CAMBVM3 CONFIG A.

### Example 2

```
hcptsys
```

In this example, HCPTSYS reads the first file it finds in the CMS search order that is named HCPSYS ASSEMBLE. It creates a configuration file named SYSTEM CONFIG on the first file mode letter in the CMS search order that is accessed R/W.





---

## Notices

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

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

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10594-1785  
U.S.A.

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

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

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

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

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

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

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

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

IBM Corporation  
Mail Station P300  
2455 South Road  
Poughkeepsie, NY 12601-5400  
U.S.A.  
Attention: Information Request

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

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

BookManager	NetRexx
C/VM	OpenEdition
DB2	OpenExtensions
DFSMS/VM	OS/2
Enterprise Storage Server	OS/390
ESCON	RACF
FICON	S/390
IBM	S/390 Parallel Enterprise Server
IBMLink	VM/ESA
Language Environment	VTAM
MQSeries	z/Architecture
Multiprise	z/VM
MVS	

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

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



---

## Glossary

For a list of VM terms and their definitions, see the glossary in the online HELP Facility. For example, to display the definition of “cms”, enter:

```
help glossary cms
```

You will enter the HELP Facility’s online glossary file and the definition of “cms” will be displayed as the current line. When you are in the glossary file, you can also search for other terms.

If you are unfamiliar with the HELP Facility, you can enter:

```
help
```

to display the main HELP Menu, or enter:

```
help cms help
```

for information about the HELP command.

For more information about the HELP Facility, see the *z/VM: CMS User’s Guide*. For more about the HELP command, see the *z/VM: CMS Command Reference*.



---

## Bibliography

This bibliography lists the publications that provide information about your z/VM system. The z/VM library includes z/VM base publications, publications for additional facilities included with z/VM, and publications for z/VM optional features. For abstracts of z/VM publications and information about current editions and available publication formats, see *z/VM: General Information*.

---

### z/VM Base Publications

#### Evaluation

- *z/VM: Licensed Program Specifications*, GC24-5943
- *z/VM: General Information*, GC24-5944

#### Installation and Service

- *z/VM: Installation Guide*, GC24-5945
- *z/VM: Service Guide*, GC24-5946
- *z/VM: VMSES/E Introduction and Reference*, GC24-5947

#### Planning and Administration

- *z/VM: Planning and Administration*, SC24-5948
- *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-5949
- *z/VM: Migration Guide*, GC24-5928
- *VM/ESA: REXX/EXEC Migration Tool for VM/ESA*, GC24-5752
- *z/VM: Running Guest Operating Systems*, SC24-5950
- *VM/ESA: Connectivity Planning, Administration, and Operation*, SC24-5756
- *z/VM: Group Control System*, SC24-5951
- *z/VM: Performance*, SC24-5952

#### Customization

- *z/VM: CP Exit Customization*, SC24-5953

#### Operation

- *z/VM: System Operation*, SC24-5954
- *z/VM: Virtual Machine Operation*, SC24-5955

### Application Programming

- *z/VM: CP Programming Services*, SC24-5956
- *z/VM: CMS Application Development Guide*, SC24-5957
- *z/VM: CMS Application Development Guide for Assembler*, SC24-5958
- *z/VM: CMS Callable Services Reference*, SC24-5959
- *z/VM: CMS Macros and Functions Reference*, SC24-5960
- *z/VM: CMS Application Multitasking*, SC24-5961
- *VM/ESA: REXX/VM Primer*, SC24-5598
- *z/VM: REXX/VM User's Guide*, SC24-5962
- *z/VM: REXX/VM Reference*, SC24-5963
- *z/VM: OpenExtensions POSIX Conformance Document*, GC24-5976
- *z/VM: OpenExtensions User's Guide*, SC24-5977
- *z/VM: OpenExtensions Command Reference*, SC24-5978
- *z/VM: OpenExtensions Advanced Application Programming Tools*, SC24-5979
- *z/VM: OpenExtensions Callable Services Reference*, SC24-5980
- *z/VM: Reusable Server Kernel Programmer's Guide and Reference*, SC24-5964
- *z/VM: Enterprise Systems Architecture/Extended Configuration Principles of Operation*, SC24-5965
- *C for VM/ESA: Library Reference*, SC23-3908
- *OS/390: DFSMS Program Management*, SC27-0806
- *z/VM: Program Management Binder for CMS*, SC24-5934
- *Debug Tool User's Guide and Reference*, SC09-2137
- *External Security Interface (RACROUTE) Macro Reference for MVS and VM*, GC28-1366
- *VM/ESA: Programmer's Guide to the Server-Requester Programming Interface for VM*, SC24-5455
- *VM/ESA: CPI Communications User's Guide*, SC24-5595
- *Common Programming Interface Communications Reference*, SC26-4399
- *Common Programming Interface Resource Recovery Reference*, SC31-6821

## End Use

- *z/VM: CP Command and Utility Reference*, SC24-5967
- *VM/ESA: CMS Primer*, SC24-5458
- *z/VM: CMS User's Guide*, SC24-5968
- *z/VM: CMS Command Reference*, SC24-5969
- *z/VM: CMS Pipelines User's Guide*, SC24-5970
- *z/VM: CMS Pipelines Reference*, SC24-5971
- *CMS/TSO Pipelines: Author's Edition*, SL26-0018
- *z/VM: XEDIT User's Guide*, SC24-5972
- *z/VM: XEDIT Command and Macro Reference*, SC24-5973
- *z/VM: Quick Reference*, SC24-5986

## Diagnosis

- *z/VM: System Messages and Codes*, GC24-5974
- *z/VM: Diagnosis Guide*, GC24-5975
- *z/VM: VM Dump Tool*, GC24-5887
- *z/VM: Dump Viewing Facility*, GC24-5966

---

## Publications for Additional Facilities

### DFSMS/VM<sup>®</sup>

- *VM/ESA: DFSMS/VM Function Level 221 Planning Guide*, GC35-0121
- *VM/ESA: DFSMS/VM Function Level 221 Installation and Customization*, SC26-4704
- *VM/ESA: DFSMS/VM Function Level 221 Storage Administration Guide and Reference*, SH35-0111
- *VM/ESA: DFSMS/VM Function Level 221 Removable Media Services User's Guide and Reference*, SC35-0141
- *VM/ESA: DFSMS/VM Function Level 221 Messages and Codes*, SC26-4707
- *VM/ESA: DFSMS/VM Function Level 221 Diagnosis Guide*, LY27-9589

### OSA/SF

- *S/390: Planning for the S/390 Open Systems Adapter (OSA-1, OSA-2) Feature*, GC23-3870
- *VM/ESA: Open Systems Adapter Support Facility User's Guide for OSA-2*, SC28-1992
- *S/390: Open Systems Adapter-Express Customer's Guide and Reference*, SA22-7403

## Language Environment<sup>®</sup>

- *Language Environment for OS/390 & VM: Concepts Guide*, GC28-1945
- *Language Environment for OS/390 & VM: Migration Guide*, SC28-1944
- *Language Environment for OS/390 & VM: Programming Guide*, SC28-1939
- *Language Environment for OS/390 & VM: Programming Reference*, SC28-1940
- *Language Environment for OS/390 & VM: Writing Interlanguage Communication Applications*, SC28-1943
- *Language Environment for OS/390 & VM: Debugging Guide and Run-Time Messages*, SC28-1942

---

## Publications for Optional Features

### CMS Utilities Feature

- *VM/ESA: CMS Utilities Feature*, SC24-5535

### TCP/IP Feature for z/VM

- *z/VM: TCP/IP Level 3A0 Planning and Customization*, SC24-5981
- *z/VM: TCP/IP Level 3A0 User's Guide*, SC24-5982
- *z/VM: TCP/IP Level 3A0 Programmer's Reference*, SC24-5983
- *z/VM: TCP/IP Level 3A0 Messages and Codes*, GC24-5984
- *z/VM: TCP/IP Level 3A0 Diagnosis Guide*, GC24-5985

### OpenEdition<sup>®</sup> DCE Feature for VM/ESA<sup>®</sup>

- *OpenEdition DCE for VM/ESA: Introducing the OpenEdition Distributed Computing Environment*, SC24-5735
- *OpenEdition DCE for VM/ESA: Planning*, SC24-5737
- *OpenEdition DCE for VM/ESA: Configuring and Getting Started*, SC24-5734
- *OpenEdition DCE for VM/ESA: Administration Guide*, SC24-5730
- *OpenEdition DCE for VM/ESA: Administration Reference*, SC24-5731
- *OpenEdition DCE for VM/ESA: Application Development Guide*, SC24-5732



- *OpenEdition DCE for VM/ESA: Application Development Reference*, SC24-5733
- *OpenEdition DCE for VM/ESA: User's Guide*, SC24-5738
- *OpenEdition DCE for VM/ESA: Messages and Codes*, SC24-5736

## LANRES/VM

- *LAN Resource Extension and Services/VM: Licensed Program Specifications*, GC24-5617
- *LAN Resource Extension and Services/VM: General Information*, GC24-5618
- *LAN Resource Extension and Services/VM: Guide and Reference*, SC24-5622

---

## CD-ROM

The following CD-ROM contains all the IBM libraries that are available in IBM BookManager® format for current VM system products and current IBM licensed programs that run on VM. It also contains PDF versions of z/VM publications and publications for some related IBM licensed programs.

- *Online Library Omnibus Edition: VM Collection*, SK2T-2067

**Note:** Only unlicensed publications are included.



---

# Index

## Special Characters

SPPF file names 5

## Numerics

32-bit CP image 11  
370-mode virtual machines 10  
3990 Model 6 Multi-Path Lock Facility support 13  
64-bit CP image 11  
64-bit support 11

## A

Active Disk Table (ADT) 26  
administration, changes related to 6  
allocation of real storage for segment tables 8  
APPCVM macro functions 43  
application development and deployment, changes related to 17  
ATTACH command 31  
AUDIT command 49  
AUDIT enhancement 7  
AUTOLOG command 31

## B

backing out the new system 88  
backlevel CMS, installing 89  
BEGIN command 31  
BFS root no longer case sensitive 24  
BFSQUERY stage 51  
binder/loader support, CMS 24  
BPXYERNO macro 59  
BPXYOPNF macro 59  
BPXYVM5 macro 59

## C

CHARACTER\_DEFAULTS system configuration statement 29  
close (BPX1CLO) routine 57  
CLOSE macro (OS Simulation) 60  
cluster bus channel support 17  
CMS, support for multiple levels of 89  
CMS callable services 52  
CMS commands 44  
CMS compatibility-interface macros 59  
CMS compatibility-interface routines 58  
CMS control blocks, viewing and printing 26  
CMS file pool administration and operator commands 49  
CMS general-interface macros 58  
CMS IPL heading 8  
CMS level, querying 18  
CMS messages 61  
CMS Migration Utility Feature no longer available 20

CMS multitasking routines 57  
CMS OS Simulation macros 60  
CMS OS Simulation supervisor calls 60  
CMS Pipelines messages 51  
CMS Pipelines stages, subcommands, and macros 51  
CMS productivity aids 7  
CMSDESK command 44  
CMSFLAG function 61  
CMSLEVEL macro 58  
connectivity, establishing between old and new systems 86  
COPYFILE command 44  
COUPLE command 31  
CP commands 30  
CP control blocks, viewing and printing 26  
CP exit modifications 11  
CP exits, dynamic 10  
CP images 11  
CP macros 43  
CP messages 43  
CP nucleus 11  
CP utilities 41  
CPLISTFILE command 31  
CPXLOAD system configuration statement 30  
CREATE DIRECTORY command 44  
CRYPTO directory control statement 30  
CSE, using to share data 77  
CSL routines that have dates as output, calling 18  
CSLLIST command 45  
CSLMAP command 45

## D

DACBGEN EXEC 26  
data compression, shell commands for 22  
data spaces CP macros 43  
DATE function 61  
DATECONVERT stage 51  
DateTimeSubtract routine 57  
DCB macro (OS Simulation) 60  
DEFAULTS command 45  
DEFINE CHPID / PATH command 31  
DEFINE command 31  
DEFINE CRYPTO command 31  
DEFINE CTCA command 31  
DEFINE CU / CNTLUNIT command 31  
DEFINE DEVICE / IODEVICE command 32  
DEFINE STORAGE command 32  
DEFNUC macro 60  
DELETE CHPID / PATH command 32  
DELETE command 32  
DELETE CU / CNTLUNIT command 32  
DELETE DEVICE / IODEVICE command 32  
DELETE RDEVICE command 32

DELETE USER command 49  
DETACH command 32  
DIAG / DIAGRC function 61  
DIAGNOSE code X'00' 42  
DIAGNOSE code X'04' 42  
DIAGNOSE code X'14' 42  
DIAGNOSE code X'210' 43  
DIAGNOSE code X'270' 43  
DIAGNOSE code X'68' 42  
DIAGNOSE code X'7C' 42  
DIAGNOSE code X'84' 42  
DIAGNOSE code X'98' 42  
DIAGNOSE code X'BC' 42  
DIAGNOSE code X'D8' 42  
DIAGNOSE codes 41  
diagnosis, changes related to 25  
DIAL command 32  
DIRBUFF macro 58  
directory, migrating 78  
directory control statements 30  
DIRECTXA utility 41  
DIRLIST command 45  
DISCONNECT command 32  
DISPLAY command 32  
DISPLAY ESA/XC Storage command 32  
DISPLAY Guest Storage command 32  
DISPLAY Host Storage command 33  
DISPLAY Linkage Stack command 33  
DISPLAY PSW command 33  
DISPLAY Registers command 33  
DMSCBLK routine 52  
DMSCDLBK routine 52  
DMSCLOSE routine 52  
DMSCRDIR routine 52  
DMSCRFIL routine 52  
DMSCROB routine 52  
DMSDEUSR routine 53  
DMSENUUSR routine 53  
DMSERP routine 53  
DMSEXIDI routine 53  
DMSEXIFI routine 53  
DMSEXIST routine 54  
DMSGETDA routine 54  
DMSGETDF routine 55  
DMSGETDI routine 55  
DMSGETDS routine 56  
DMSGETDX routine 56  
DMSOPBLK routine 56  
DMSOPDBK routine 56  
DMSOPEN routine 56  
DMSQEFL macro 58  
DMSQEFL routine 56  
DMSQSFSL routine 56  
DMSRDCAT routine 57  
DMSTRUNC routine 57  
DMSTVS routine 58  
DUMP command 33  
DUMP ESA/XC Storage command 33  
DUMP Guest Storage command 33  
DUMP Host Storage command 33  
DUMP Linkage Stack command 33

DUMP Registers command 33  
Dump Viewing Facility functions 64  
DUMPSCAN command 64  
dynamic CP exits 10  
dynamic link library (DLL) support 19

## E

electronic delivery of service 3  
ERASE command 45  
euro support 27  
EXEC 2 applications, migrating 98  
EXSBUFF macro 59  
Extended-TOD-clock support 17

## F

fcntl (BPX1FCT) routine 57  
FEATURES system configuration statement 30  
fibre connection (FICON) channel support 17  
file pool administration and operator commands 49  
file pool server, converting back to old release 85  
file pool servers, migrating 81  
File Status Table (FST) 25  
FILE subcommand (XEDIT) 50  
FILEDEF command 45  
FILELIST command 45  
FILEPOOL RELOAD command 49  
FILEPOOL UNLOAD command 49  
Flash Copy support 17  
fork (BPX1FRK) processing 22  
FSSTATE macro 59  
FSTD macro 59

## G

GCS commands and macros 62  
GCS pathlength reduced 19  
GCSLEVEL macro 62  
GENCPBLS EXEC 63  
GET macro (OS Simulation) 60  
GET subcommand (XEDIT) 50  
GETMAIN macro 62  
GLOBAL command 45  
glossary information 115  
guest coupling simulation support 13  
guest cryptographic support 14  
guest support, changes related to 12  
GUI Facility changes 12

## H

hardware architecture and facilities, changes related to 16  
HCPRIO, converting to system configuration file 65  
HCPSPYS, converting to system configuration file 65  
HCPTKDEF macro 43  
HELPIINST saved segment 7  
High Level Assembler 11

## I

IDENTIFY command 46  
IEEE Floating Point support 15

incompatible change, definition of 2  
INDICATE command 33  
INDICATE QUEUES command 34  
INDICATE SPACES command 34  
INDICATE USER command 34  
installation, changes related to 3  
integrated cluster bus channel support 17  
IPL command 34  
IPLing with the NODIRECT option 10  
ITNVTSTR EXEC 3, 63  
IUCV functions 43

## J

Java support 16

## L

Language Environment included with VM 19  
LISTDS command 46  
LISTFILE command 46  
LKED command 46  
LOAD command 46  
LOAD subcommand (XEDIT) 50  
LOADMOD command 46  
local modification of replacement-maintained parts 3  
LOCATE (Storage) command 34  
LOCATE CMDBK command 34  
LOCATE command 34  
LOCATE DGNDBK command 34  
LOCATE FILID command 34  
LOCATE FRAMETBL command 34  
LOCATE LDEV command 34  
LOCATE VDEV command 34  
LOCATE VSMBK command 34  
LOCATEVM command 34  
LOCK command 34  
logical device limit 8  
LOGON / LOGIN command 35

## M

MACLIST command 46  
MDISK directory control statement 30  
MODIFY CHPID / PATH command 35  
MODIFY CU / CNTLUNIT command 35  
MODIFY DEVICE / IODEVICE command 35  
mount (BPX1MNT) routine 57  
MOVEFILE command 47  
MQSeries Client for VM/ESA 16  
Multi-Path Lock Facility support 13  
multiple levels of CMS 89

## N

NETDATA command 47  
NetRexx support 16  
networking support, changes related to 16  
NODIRECT option 10

non-relocatable modules loaded between 16-20MB 20  
NOTE command 47  
NUCXLOAD command 47

## O

object directory compatibility 78  
OPEN macro (OS Simulation) 60  
OpenEdition enhancements (2.3.0) 21  
OpenEdition enhancements (2.4.0) 23  
OpenEdition renamed to OpenExtensions 24  
OpenExtensions callable services 57  
OpenExtensions commands 49  
OpenExtensions macros 59  
OpenExtensions open files limit 24  
OPENVM commands 49  
OPENVM DEBUG command 50  
OPENVM MOUNT command 50  
OPENVM OWNER command 50  
OPENVM QUERY MOUNT command 50  
OPENVM RUN command 50  
openvmf (BPX1VM5) routine 57  
operation, changes related to 6  
OPTION directory control statement 30  
OS Simulation enhancements (2.4.0) 23  
OS simulation enhancements (3.1.0) 24  
OS Simulation macros 60  
OS Simulation supervisor calls 60  
OS Simulation support for Tape Library Dataserver 19  
OSA Express Gigabit Ethernet support 14  
OSA/SF 17

## P

Parallel Access Volumes (PAV) support 15  
PARSE instruction 61  
PEEK command 47  
PIPE command 47  
Pipelines code bases merged 20  
Pipelines messages 51  
Pipelines stages, subcommands, and macros 51  
PIPEPVR macro 51  
PPF file names 5  
product enablement support 10  
pseudo timer extended 18  
PTF, removing 5  
PUT macro (OS Simulation) 60  
PUT subcommand (XEDIT) 50  
PUTD subcommand (XEDIT) 50

## Q

QDIO facility support 14  
QUERY (Virtual Device) command 38  
QUERY CACHEFW command 35  
QUERY CHPID command 35  
QUERY CMSLEVEL command 47  
QUERY CMSREL command 47  
QUERY command (CMS) 47  
QUERY command (CP) 35

QUERY CPLEVEL command 35  
 QUERY CPLOAD command 36  
 QUERY CRYPTO command 36  
 QUERY DASD command 36  
 QUERY DASDFW command 36  
 QUERY EXITS command 36  
 QUERY FILEDEF command 48  
 QUERY FRAMES command 36  
 QUERY GCSLEVEL command 62  
 QUERY GRAF command 36  
 QUERY IMG command 36  
 QUERY LDEVS command 36  
 QUERY MODDATE command 62  
 QUERY NAMES command 36  
 QUERY NLS command 37  
 QUERY NSS command 37  
 QUERY READER / PRINTER / PUNCH  
 command 37  
 QUERY SPACES command 37  
 QUERY stage 51  
 QUERY STORAGE command 37  
 QUERY TIME command 37  
 QUERY TOKEN command 38  
 QUERY TRACE command 38  
 QUERY TRFILES command 38  
 QUERY UCR command 38  
 QUERY USERID command 38  
 QUERY USERS command 38  
 QUERY VIRTUAL ALL command 38  
 QUERY VIRTUAL CONSOLE  
 command 38  
 QUERY VIRTUAL CRYPTO  
 command 38  
 QUERY VIRTUAL CTCA command 38  
 QUERY VIRTUAL GRAF command 38  
 QUERY VIRTUAL OSA command 39  
 QUERY VIRTUAL STORAGE  
 command 39

## R

RDRLIST command 48  
 read (BPX1RED) routine 58  
 READER stage 51  
 RECEIVE command 48  
 Removable Media Services (RMS) free  
 drive support 7  
 replacement-maintained parts, local  
 modification of 3  
 reserve/release considerations for  
 VSE 72  
 REXX applications, migrating 98  
 REXX Sockets API 18  
 REXX/VM instructions and functions 61  
 RIO370 11

## S

S/390 Service Update Facility 4  
 SALIPL utility 41  
 SAVE subcommand (XEDIT) 50  
 saved segments, migrating 66  
 segment tables, allocation of real storage  
 for 8  
 SEND command 39  
 SENDFILE command 48  
 service, changes related to 3

service, electronic delivery of 3  
 Service Update Facility (SUF) 4  
 SET command (CMS) 48  
 SET command (CP) 39  
 SET CPTRACE command 39  
 SET CRYPTO command 39  
 SET DUMP command 39  
 SET MACHINE command 39  
 SET MDCACHE command 39  
 SET RDEVICE command 39  
 SFS file pool server, converting back to  
 old release 85  
 SFS file pool servers, migrating 81  
 sharing data among CMS users on  
 multiple systems 77  
 sharing data between your old system  
 and new system 72  
 SHUTDOWN command 39  
 SPECIAL directory control statement 30  
 spool files, migrating 66  
 SPTAPE command 39  
 STORE (Registers) command 40  
 STORE command 39  
 STORE ESA/XC Storage command 39  
 STORE Guest Storage command 40  
 STORE Host Storage command 40  
 STORE PSW command 40  
 STORE STATUS command 40  
 SVC 19 (OPEN) (OS Simulation) 60  
 SVC 20 (CLOSE) (OS Simulation) 60  
 SYSTEM command 40  
 system configuration file statements 29  
 system diagnosis, changes related to 25  
 SYSTEM\_RESIDENCE system  
 configuration statement 30

## T

TAPE command 48  
 tape FICON and RAS support 25  
 Tape Library Dataserver, OS Simulation  
 support for 19  
 tape support enhancements 15  
 TAPECTL macro 59  
 TCP/IP awareness in CP 8  
 TELL command 48  
 TIME macro (GCS) 62  
 TIME macro (OS Simulation) 60  
 TRACE command 40  
 TRACE mnemonic1 command 40  
 TRACE TABLE command 40  
 TRSOURCE ID command 40

## U

UNLOCK command 41  
 upwardly compatible change, definition  
 of 2  
 USER\_DEFAULTS system configuration  
 statement 30  
 user directory, migrating 78  
 USER directory control statement 30  
 user directory control statements 30

## V

VARY (Real Device) command 41  
 VM data spaces CP macros 43

VM Dump Tool 26  
 VMCF, full MP capability for 19  
 VMDUMP command 41  
 VMFAPPLY EXEC 63  
 VMFASM EXEC 63  
 VMFBLD EXEC 63  
 VMFENRPT EXEC 4  
 VMFEXUPD EXEC 63  
 VMFHASM EXEC 63  
 VMFHLASM EXEC 63  
 VMFINS DELETE command 63  
 VMFINS EXEC 63  
 VMFINS INSTALL command 63  
 VMFINS MIGRATE command 63  
 VMFMRDSK EXEC 63  
 VMFNLS EXEC 63  
 VMFPLC2 command 48  
 VMFREC EXEC 63  
 VMFREM EXEC 5, 63  
 VMFREPL EXEC 63  
 VMFSETUP EXEC 63  
 VMFSIM EXEC 63  
 VMFSIM GETLVL command 63  
 VMFSUFIN EXEC 4, 64  
 VMFSUFTB EXEC 4  
 VMFVIEW EXEC 64  
 VMLINK command 49  
 VMLINK improvements 12  
 VMSES/E enhancements (2.3.0) 4  
 VMSES/E execs 62  
 VMSES/E messages 64

## W

write (BPX1WRT) routine 58

## X

XAUTOLOG command 41  
 XEDIT subcommand (XEDIT) 50  
 XEDIT subcommands 50

## Y

Year 2000 support (2.2.0) 6  
 Year 2000 support (2.3.0) 7

## Z

z/Architecture support 16



---

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

z/VM  
Migration Guide  
Version 3 Release 1.0

Publication No. GC24-5928-00

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

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

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

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

Please tell us how we can improve this book:

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

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

---

Name

---

Address

---

Company or Organization

---

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation  
Information Development  
Department G60G  
1701 North Street  
Endicott, New York  
13760-5553



Fold and Tape

Please do not staple

Fold and Tape







File Number: S370/S390-20  
Program Number: 5654-A17



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

GC24-5928-00

