Virtual Machine/
Enterprise Systems Architecture

Conversion Guide and Notebook

Version 2 Release 4.0

GC24-5839-01

Virtual Machine/
Enterprise Systems Architecture

Conversion Guide and Notebook

Version 2 Release 4.0

GC24-5839-01

Note:

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

Second Edition (July 1999)

This edition applies to Version 2, Release 4, Modification 0 of IBM® Virtual Machine/Enterprise Systems Architecture (VM/ESA®) (product number 5654-030) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces GC24-5839-00.

Contents

Notices

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

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

```
IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
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 to 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,
522 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 document 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 on 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 contains 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 contains 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 the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. 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.

Programming Interface Information

This book documents information NOT intended to be used as Programming Interfaces of VM/ESA.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

ACF/VTAM
Advanced Function Printing
BookManager
C/VM
CICS/ESA
DATABASE 2
DB2
DFSMS
DFSMS/VM
ECKD
Enterprise Systems Connection Architecture
ES/3090
ES/4381
ES/9000
ESA/390
ESCON
IBM
IBMLink
Language Environment
MQSeries
Multiprise
MVS
MVS/ESA
MVS/SP
MVS/XA
NetRexx
OfficeVision
OfficeVision/VM
OpenEdition
OS/2
OS/390
OS/400
PR/SM
Processor Resource/Systems Manager
PROFS
RACF
RETAIN
S/370
S/390
S/390 Parallel Enterprise Server
SQL/DS
System/370
VM/ESA
VTAM
3090
3890

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.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

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

Preface

This book provides information to help you convert (migrate) to VM/ESA Version 2 Release 4.0 from any of the following previous VM/ESA releases:

- VM/ESA Version 1 Release 1.5 370 Feature
- VM/ESA Version 1 Release 2.0
- VM/ESA Version 1 Release 2.1
- VM/ESA Version 1 Release 2.2
- VM/ESA Version 2 Release 1.0
- VM/ESA Version 2 Release 2.0
- VM/ESA Version 2 Release 3.0

Converting from Other VM Releases

This book does not directly address conversions from:

- VM/SP Release 5 or 6
- VM/SP HPO Release 5 or 6
- VM/XA SP Release 2.0 or 2.1
- VM/ESA Version 1 Release 1.0 (370 Feature)
- VM/ESA Version 1 Release 1.0 (ESA Feature)
- VM/ESA Version 1 Release 1.1

However, if you plan to convert from one of those releases, you can get the conversion information you need by first obtaining one of the following books (not supplied with VM/ESA Version 2 Release 4.0), which provide information about converting from those releases to VM/ESA Version 2 Release 1.0:

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

Then, use this book to determine the additional conversion considerations between VM/ESA Version 2 Release 1.0 and VM/ESA Version 2 Release 4.0.

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 system conversions. Parts of this book can also help application programmers.

What You Should Know before Reading This Book

This book assumes that you are familiar with the system from which you are converting and that you have a general understanding of VM/ESA Version 2 Release 4.0 from reading the *VM/ESA: General Information* book.

What This Book Contains

This book:

- Helps you plan for your conversion. It contains:
  - Information on what education you need to do a conversion
  - Tables for you to fill in to keep track of your system inventory
  - Help on what conversion strategy would work best for you
  - Information on what you should include in your conversion plan
- Provides instructions and information on certain conversion tasks, such as:
  - How to bring up your new system as a guest on the old system
  - How to convert your user directory
  - How to convert your segments and spool files
  - How you can use connectivity between your old and new systems
  - How to share data between systems
  - How to convert your system configuration files
  - How to install a backlevel CMS
  - How to convert your SFS servers
- Describes new functions added to VM/ESA since your release
- Describes differences between your system and VM/ESA Version 2 Release 4.0 in the areas of:
  - Installation
  - Service
  - Administration
  - System operation
  - Virtual machine operation
  - Application programming
  - Diagnosis
- Describes the changes in external interfaces (commands, messages, and so on) between your system and VM/ESA Version 2 Release 4.0

You will find little information in this book about exploiting the new and improved functions in the new system. IBM advises you to do your conversion first, then exploit new functions in the new system as a second step. If you make improvements to your system while converting, this adds to the size of the project and hampers your ability to return to your previous system. While converting, make note of improvements you can make and return to them after you have completed your conversion. One exception is the CP configurability function, which is new if you are converting from VM/ESA Version 1 Release 1.5 370 Feature. IBM recommends that you exploit this new function during your conversion.

You can find more information about exploiting the new functions of VM/ESA Version 2 Release 4.0 in the other VM/ESA books.

Where to Find More Information

See the bibliography on page 675.

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other VM/ESA documentation:

- Visit our home page at:

  `http://www.ibm.com/s390/vm`

  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:

Format	Address
Internet	pubrcf@vnet.ibm.com
IBM Mail	USIB2L8Z@IBMMAIL
IBMLink™	GDLVME(PUBRCF)

  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.

- Fill out the form at the back of this book and return it by mail, by fax, or by giving it to an IBM representative.

Part 1. Planning the Conversion

Chapter 1. Introduction

This book assumes that you have already had discussions with your systems engineer or your marketing representative about your hardware requirements, including your DASD, real storage, and performance needs.

Terminology and Symbols

The following sections describe some of the conversion terminology used throughout this book.

Definition of Conversion

In this book, the term “conversion” indicates the transfer and adjustment required to move to VM/ESA Version 2 Release 4.0 from a previous VM/ESA release. Rather than starting from scratch when you upgrade from your old release, you probably transfer and adjust a lot of your old information to use on the new. Information you may transfer includes:

- Application programs
- I/O configurations
- Memory configuration (saved segment definitions)
- Spool files

This task is often called migration, and the term migration is also sometimes used in this book.

How to Identify the Topics That Apply to Your Conversion

This book provides information for converting to VM/ESA Version 2 Release 4.0 (abbreviated as VM/ESA 2.4.0 in this book) from the following releases:

- VM/ESA Version 1 Release 1.5 370 Feature (VM/ESA 1.1.5 370 Feature)
- VM/ESA Version 1 Release 2.0 (VM/ESA 1.2.0)
- VM/ESA Version 1 Release 2.1 (VM/ESA 1.2.1)
- VM/ESA Version 1 Release 2.2 (VM/ESA 1.2.2)
- VM/ESA Version 2 Release 1.0 (VM/ESA 2.1.0)
- VM/ESA Version 2 Release 2.0 (VM/ESA 2.2.0)
- VM/ESA Version 2 Release 3.0 (VM/ESA 2.3.0)

Some topics in this book apply only to the conversion from a particular release. Other topics apply to conversions from more than one release. In Part 2 of this book (“Conversion Tasks and Considerations”), one or more of the following symbols may be included in brackets [ ] following a heading or at the beginning of an item in a list, or may be used as the label on a labeled box, to identify the conversions to which a topic applies:

ALL Applies to all the conversions documented in this book.

S370 Applies only to conversion from a System/370™ architecture release. This means a conversion from VM/ESA 1.1.5 370 Feature.

ESA Applies only to conversion from an ESA architecture release. This means a conversion from VM/ESA 1.2.0, VM/ESA 1.2.1, VM/ESA 1.2.2, VM/ESA 2.1.0, VM/ESA 2.2.0, or VM/ESA 2.3.0.

1.1.5 Applies to conversion from VM/ESA 1.1.5 370 Feature.

1.2.0 Applies to conversion from VM/ESA 1.2.0.

1.2.1 Applies to conversion from VM/ESA 1.2.1.

1.2.2 Applies to conversion from VM/ESA 1.2.2.

2.1.0 Applies to conversion from VM/ESA 2.1.0.

2.2.0 Applies to conversion from VM/ESA 2.2.0.

2.3.0 Applies to conversion from VM/ESA 2.3.0.

Therefore, if you are doing a conversion from VM/ESA 1.1.5 370 Feature, you should read all the topics in Part 2 marked with any of the following symbols: **ALL**, **S370**, or **1.1.5**. Similarly, if you are doing a conversion from VM/ESA 1.2.2, you should read all the topics in Part 2 marked with any of the following symbols: **ALL**, **ESA**, or **1.2.2**.

When one or more symbols are specified for a topic (such as a heading), the same set of symbols applies to all subtopics (subheadings) unless a new set of symbols is specified on a subtopic. For example, in the following heading sequence:

```
Installing CMS [1.1.5]
⋮
    Installing the CMS Named Saved System
⋮
    CMS Nucleus Changes
```

all the topics apply only to conversion from VM/ESA 1.1.5 370 Feature.

In most cases, if a new set of symbols is introduced on a subtopic, the appropriate symbols are specified on all the topics at that level to avoid confusion. Also, the new set of symbols introduced on a subtopic is always narrower than the one on the higher level topic, never broader. This means that the new set of symbols is always a subset of the symbols on the higher level topic. For example, consider the following heading sequence:

```
CPI Communications Changes [1.1.5, 1.2.0]
⋮
    TXTLIBs Required for Module Creation [1.1.5, 1.2.0]
⋮
    Event Management [1.1.5]
```

Part 3 of this book (“Compatibility Tables”) contains a separate chapter for each type of conversion. You need to read only the chapter that applies to your conversion. Each chapter contains a section for each component of VM/ESA that has changed between that VM/ESA release and VM/ESA 2.4.0. A section for the Programmable Operator Facility is also included, if appropriate. Within each section, the specific externals (commands, messages, and so on) that have changed are listed in tables.

Compatibility Terms Used in This Book

This book uses certain terms to convey the degree of compatibility between the external interfaces of your current system and those of the new system. External interfaces are commands, routines, utilities, DIAGNOSE codes, messages, macros, user directory control statements, and system services. The terms are:

Compatible The external interface exists in both systems and has equivalent syntax, functions, and responses.

Upwardly compatible The external interface exists in both systems, but there are differences in syntax, functions, or responses. Invocations and applications using the external interface as it is on your current system should continue to execute unchanged.

Incompatible The external interface exists in both systems, but differences in syntax, functions, or responses may cause some invocations or applications using the external interface as it is on your current system to execute incorrectly or to not execute at all.

VM/ESA 1.1.5 370 Feature only The external interface exists in VM/ESA 1.1.5 370 Feature but not in VM/ESA 2.4.0.

VM/ESA 1.2.0 only The external interface exists in VM/ESA 1.2.0 but not in VM/ESA 2.4.0.

VM/ESA 1.2.1 only The external interface exists in VM/ESA 1.2.1 but not in VM/ESA 2.4.0.

VM/ESA 1.2.2 only The external interface exists in VM/ESA 1.2.2 but not in VM/ESA 2.4.0.

VM/ESA 2.1.0 only The external interface exists in VM/ESA 2.1.0 but not in VM/ESA 2.4.0.

VM/ESA 2.2.0 only The external interface exists in VM/ESA 2.2.0 but not in VM/ESA 2.4.0.

VM/ESA 2.3.0 only The external interface exists in VM/ESA 2.3.0 but not in VM/ESA 2.4.0.

VM/ESA 2.4.0 only The external interface exists in VM/ESA 2.4.0 but not in your current system

Note: Changes in messages and return codes that result from entering CP commands are not listed as changes in the CP command compatibility tables.

Wherever you see "———" in compatibility tables in this book, it means that the external interface does not exist.

Examples of What Incompatible Means

For commands, routines, and macros: *Incompatible* means that some part of the command, routine, or macro has changed between your current release and the new system, and the change may require you to change your use of it. An incompatible command, routine, or macro may actually be upwardly compatible for you.

For example, if you look at the entry for the CMS ERASE command in the CMS commands compatibility table for VM/ESA 1.1.5 370 Feature (page 389), you see that a restriction no longer exists in the new system. In VM/ESA 1.1.5 370 Feature, you received an error message if you tried to erase a directory or directories that were open. On the new system, you do not receive this error message. If you were never concerned with this situation, then you do not need to change anything; the command is upwardly compatible for you. But if you were concerned about this situation and were expecting to get that error message in this particular situation, then ERASE is incompatible for you. You may have to change your application.

For messages: *Incompatible* means that there is some type of difference in the text of the message between your current release and the new system that may cause you to need to rewrite any application that depends on it.

For example, you may have received this CMS message on your system:

```
Disk A is accessed as read-only
```

With the new system, you may get the following CMS message for the same situation:

```
Filemode A is accessed as read-only
```

One text difference in a message that is not considered incompatible is appended text. When text is appended, the message is considered upwardly compatible. Another text difference that is not considered incompatible is a field in a message that may have a new value to support new function. This message would be considered upwardly compatible because you would not receive the new value unless you exploit new function.

Chapter 2. Starting Your Conversion Planning

Planning for conversion is the process of organizing the conversion project to produce a new system that meets your installation's requirements.

Planning should begin early, ideally as soon as you make the decision to order the new system.

Getting Help

If you need help during your conversion, contact your local support structure. Customers in the U.S. can also call 1-800-IBM-4-YOU.

Conversion Planning Steps

Key points you have to consider for conversion planning are:

1. **Educate yourself about the impact of major items that are handled differently in the new system.**

   See Chapter 3, “Educating Yourself and Your Conversion Staff” on page 9.

2. **Understand, know, establish, and/or update your hardware, software products, applications, local modifications, and tools.**

   See Chapter 4, “Preparing Hardware and Software Inventory” on page 13.

3. **Determine a strategy for your system conversion.**

   See Chapter 5, “Determining Which Conversion Strategies You Want to Use” on page 21.

4. **Document your conversion plan.**

   See Chapter 6, “Create a Detailed Conversion Plan” on page 49.

Chapter 3. Educating Yourself and Your Conversion Staff

To prepare for your conversion, you and your conversion staff may need to get educated in certain areas. This section discusses what types of education you might want to prepare yourself with, including IBM documentation and IBM education.

Other sources of information might include:

- User group presentations and conference proceedings, such as from SHARE, GUIDE, or SEAS.
- Flashes or EQUAL questions in RETAIN®, which you can get to through IBMLink.
- Red or orange books produced by IBM system centers.

For additional sources of information that can help with your conversion, consider talking to your local marketing support contact or visit the IBM VM operating system home page at http://www.ibm.com/s390/vm.

Understanding the Changes in the System

Some of the major changes for your conversion that you and your conversion staff should become familiar with include:

- Architecture
- I/O - software and hardware
- Installing and servicing your system
- Saved segment support
- Spool files
- Performance
- User directories
- Connectivity
- Changes for application programs and program products.

IBM Documentation That Can Help

A good place to start is with this book, which contains many details on the changes in the system since your current release.

Other books that may help you are listed with the topics below.

Architecture

- *VM/ESA: Enterprise Systems Architecture/Extended Configuration Principles of Operation*, for:
  - Information on data spaces and ESA/XC

I/O - Software and Hardware

- *VM/ESA: Planning and Administration*, for:
  - CP configurability
  - HCPRIO
- *VM/ESA: General Information*, for:
  - VM/ESA Processor Support Matrix and VM/ESA Device Support Matrix, which list the processors and devices that VM/ESA supports
- *ES/9000 and ES/3090 Input/Output Configuration Program User Guide*, for:
  - Information on IOCP, including setting up an IOCP for a logical partition (LPAR)

Installing and Servicing Your System

- This book:
  - Chapter 10, “Preinstallation and Installation Changes and Conversion Considerations” on page 209
  - Chapter 11, “Service Changes and Conversion Considerations” on page 249
- *VM/ESA: Installation Guide*
- *VM/ESA: VMSES/E Introduction and Reference*
- *VM/ESA: Service Guide*
- Product installation books
- Packaging installation books
- Program directory
- Memo to users

Saved Segment Support

- This book:
  - “Converting Spool Files and Saved Segments [ESA]” on page 91
- *VM/ESA: Planning and Administration*, for:
  - Creating and managing saved segments
- *VM/ESA: CMS Application Development Guide for Assembler*, for:
  - Using saved segments in assembler programs
- *VM/ESA: CMS Application Development Guide*, for:
  - Using saved segments in high-level-language programs
- *VM/ESA: Service Guide*, for:
  - Information on rebuilding saved segments

Spool Files

- This book:
  - “Converting Spool Files and Saved Segments [ESA]” on page 91
- *VM/ESA: CP Diagnosis Reference*, for
  - Information on management of local spooling operations
- *VM/ESA: CP Programming Services*, for:
  - SFBLOK information
- *VM/ESA: CP Command and Utility Reference*, for:
  - SPXTAPE and SPTAPE command information

Performance

- *VM/ESA: Performance*, for:
  - General performance and tuning information
- *VM/ESA: General Information*, for:
  - Performance improvements and considerations in VM/ESA 2.4.0
- *VM/ESA Performance Report* (see the IBM VM operating system home page at http://www.ibm.com/s390/vm)
- MONITOR LIST1403 file (shipped with VM/ESA), for:
  - Up-to-date monitor record changes

User Directories

- This book:
  - “How to Convert Your User Directory [ALL]” on page 107
- *VM/ESA: Planning and Administration*, for:
  - Planning for the user directory
  - User directory control statements
- *VM/ESA: CP Command and Utility Reference*, for:
  - DIRECTXA utility
- *VM/ESA: Running Guest Operating Systems*, for:
  - Sample directory entries for guests

Connectivity

- This book:
  - “Establishing Connectivity between First-Level and Second-Level Systems [ALL]” on page 117
  - Compatibility tables for changes to GCS commands and macros, AVS messages, APPC/VM and IUCV macros
- *VM/ESA: Connectivity Planning, Administration, and Operation*, for:
  - APPC communications
  - TSAF
  - AVS
  - ISFC
- *VM/ESA: Group Control System*, for:
  - GCS information
- *VM/ESA: CP Programming Services*, for:
  - APPCVM macros
  - IUCV macros
  - Converting programs from IUCV to APPC/VM
- *VM/ESA: Planning and Administration*, for:
  - Planning for SNA Console Communication Services (SNA/CCS)

Changes for Application Programs and Program Products

- This book:
  - Chapter 15, “Application Programming Changes and Conversion Considerations” on page 289
  - Compatibility tables for changes to REXX commands, APPC/VM macros, GCS commands and macros, CMS commands and macros, CP commands, and so on.
- *VM/ESA: REXX/EXEC Migration Tool for VM/ESA*, which comes packaged with the product library. The tool is shipped on a separate tape with the VM/ESA System DDR tapes. The kit, which includes both the book and the tool, can

also be ordered. This tool highlights incompatibilities in REXX and EXEC 2 execs.

- *Licensed Products Migration Matrix for VM/ESA*, which is available on the IBM VM operating system home page (http://www.ibm.com/s390/vm).
- *Non-IBM Solution Developers' Products*, which is available on the IBM VM home page.
- *IBM Operating Systems Supported as Guests of VM/ESA*, which is available in the *VM/ESA: General Information* book and on the IBM VM home page.

IBM Education that Can Help

Consider taking the following types of IBM courses:

- VM/ESA* from VM/XA* migration workshop
- VMSES/E for installation and service

For more information about courses, see the IBM VM operating system home page (http://www.ibm.com/s390/vm).

Chapter 4. Preparing Hardware and Software Inventory

You should begin your conversion with an accurate inventory of all the elements of your system, including hardware and software products. Gathering the information shown in the following tables can help you understand your hardware and software in preparation for your conversion. You may not be able to complete gathering this information at this time. You may need to gather it as you go along during the planning and preparation of your conversion. Appendix A, “Hardware and Software Inventory Tables” on page 623 contains empty tables that you can use to keep this information about your system.

Hardware Inventory

Use the following tables to help you understand your hardware inventory. Fill in the information as completely as you can. The more information you can gather about your installation, the easier it will be to plan your conversion. The following information may help you find this information:

- The VM processor support matrix
- Your IOCP listing file
- HCPRIO or SYSTEM CONFIG file
- Your user directory
- *VM/ESA: General Information* manual

It may also help for you to sketch the physical layout of your hardware configuration, including your processor and peripherals.

Processors

Following is an example of the type of information you should gather about your processors. You can use an empty table provided for you in “Processors” on page 623 to fill in this information about your processors. At the bottom of the table, there is a section called 'Column definitions.' This is an explanation of what should go into each column.

The VM processor support matrix may help you with some of this information.

Old	New	Type and model	Patch or EC level	PT	MSTO	XSTO	Modes	PTFs	REL	PC	EC	SI
√		4381-92		N	64MB		ESA/370	None	ESA1.2.0	8	0	Y
	√	9121		N	128MB	128MB	ESA/390™		ESA2.3.0, ESA1.2.0 (LPAR)	40	8	N

Column definitions:

Column	What it means
Old	Check this column if you currently use this processor in your old system.
New	Check this column if you plan to use this processor in your new system.
Patch/EC level	Patch or EC level required. The new system requires you to be at a certain Engineering Change level. See your IBM representative for help if your processor is not at the latest level.
PT	Can the processor be physically partitioned?
MSTO	The amount of main storage you have.
XSTO	The amount of extended storage (XSTORE) you have.
Modes	The processor modes supported by the processor, S/370™, ESA/370, or ESA/390. See the VM processor support matrix for this information.
PTFs	Service you need to apply to your current system due to the hardware you plan to use in the new system. It may be critical that you apply hardware service to your current system in case you have to back out.
REL	The supported VM releases that you are interested in.
PC	The number of parallel channels.
EC	The number of ESCON® channels.
SI	Does it have VMA/SIE assist?

DASD Used and System DASD Layout

Following is an example of the type of information you should gather about your DASD. You can use an empty table provided for you in “DASD Used and System DASD Layout” on page 624 to fill in this information about your DASD. At the bottom of the table, called 'Column definitions,' is an explanation of what should go into each column.

Your user directory may help you with this information.

Old	New	Volume label	Type and model	Dev Num	Used for	2.4.0?	SH?	ES?
√		ESARES	3380-1	123	System -- nucleus, system minidisks, directory, warm start etc. areas, TDISK space	Y	N	N
√	√	MDISKS	3380-1	125	User minidisks	Y	Y	N
	√	240RES	3390-1	122	System -- nucleus, system minidisks, directory, warm start, checkpoint, TDISK space	Y	N	Y
√	√	SYSSPO	3380-1	131	Spool space	Y	N	N
√	√	SYSPAG	3380-1	132	Page space	Y	N	N

Old	New	Volume label	Type and model	Dev Num	Used for	2.4.0?	SH?	ES?
√	√	SQLPK	3380-1	130	SQL database	Y	N	N

Column definitions:

Column **What it means**
Old Check this column if you currently use this DASD in your old system.
New Check this column if you plan to use this DASD in your new system.
Dev Num Real device number of the DASD.
2.4.0? Is this DASD supported in VM/ESA 2.4.0?
SH? Is this DASD shared? Shared DASD could be CSE volumes, volumes used with reserve/release, and so on.
ES? Is the DASD attached through an ESCON controller?

System DASD Layout

Following is an example of the type of information you should gather about your system DASD layout. You can use an empty table provided for you in “System DASD Layout” on page 625 to fill in this information about your system. At the bottom of the table, called 'Column definitions,' is an explanation of what should go into each column.

O N B	Volume	CP nuc	WARM	CKPT	TDISK	PARM	DRCT	Paging	Spool	System MDISKS	Other (what for)
O	ESARES	21-30	33-34	31-32	597-636		1-20	35-94	95-191	192-596 637-884	
B	MDISKS										User disks 1-884
N	240RES	18-26	33-34	31-32	35-68	218-251	1-17	158-217	69-158	218-1112	
B	SYSSPO								1-884		
B	SYSPAG							1-884			
B	SQLPK										SQL data 1-884

Column definitions:

O/N/B Specify 'O' if you use it in the old system, 'N' if you use it in the new system, 'B' for both old and new.
Volume Volume label.
CP nuc CP nucleus, start and end.
WARM Warmstart data area, start and end.
CKPT Checkpoint data area, start and end.
TDISK TDISK space, start and end.
PARM PARM space, start and end; PARM space is used with the CP configurability function (which is new if you are converting from VM/ESA 1.1.5 370 Feature). The *VM/ESA: Planning and Administration* book talks about the CP configurability function and PARM space.
DRCT User directory, start and end.
System MDISKS System minidisks and SFS space, start and end.
Other (what for) Note what it is for and the start and end.

Other Devices

Following is an example of the type of information you should gather about your other devices. You can use an empty table provided for you in “Other Devices” on page 627 to fill in this information. At the bottom of the table, called 'Column definitions,' is an explanation of what should go into each column.

For information on tape drives supported in the new system, see the *VM/ESA: General Information*.

Old	New	Device	Type and model	2.4.0?	Device number
√	√	Tape drive	3480	Y	231
√	√	Tape controller	3803-2	Y	200
√	√	Printer	3820	Y	112
√	√	Operator console	3278 (2)	Y	090, 091
√	√	Terminals	3179-G1s (150)	Y	
√	√	Terminals	3290-220 (250)	Y	A01-AFA
√	√	Display control unit	3174	Y	A00
√	√	DASD controller	3880-1	Y	500
	√	DASD controller	3990-1	Y	600

Column definitions:

Column	**What it means**
Old	Check this column if you currently use this device in your old system.
New	Check this column if you plan to use this device in your new system.
2.4.0?	Is this device supported by VM/ESA 2.4.0?

Software Inventory

Use the following tables to help you understand your software inventory. Fill in the information as completely as you can. The more information you can gather about your installation, the easier it will be to plan your conversion.

The following may help you find this information:

- *Licensed Products Migration Matrix for VM/ESA* (on the VM home page)
- *Non-IBM Solution Developers' Products* (on the VM home page)
- *VM/ESA: General Information* manual.

Included are other sources of information that may help you fill in the tables.

Guests

Following is an example of the type of information you should gather about your other guest systems. You can use an empty table provided for you in “Guests” on page 628 to fill in this information. At the bottom of the table, called 'Column definitions,' is an explanation of what should go into each column.

To find out where guests are supported, see the Guest Support Matrix in *VM/ESA: General Information*.

Old	New	Guest name and release level	Type of guest (V=R, V=F, V=V)	Type of VM it can run in (370, XA, ESA)
√	√	VSE/SP 3 or later	V=R	370

Column definitions:

Column **What it means**
Old Check this column if the guest is on your old system.
New Check this column if you plan to use this guest in your new system.

IBM Licensed Products or Vendor Products

Following is an example of the type of information you should gather about your licensed products and vendor products. You can use an empty table provided for you in "IBM Licensed Products or Vendor Products" on page 629 to fill in this information. At the bottom of the table, called 'Column definitions,' is an explanation of what should go into each column.

For information on IBM licensed programs supported in the new system, see the *Licensed Products Migration Matrix for VM/ESA*. For information on vendor products supported in the new system, see the *Non-IBM Solution Developers' Products* list.

OLD	NEW	Name and release level	KP	US	REL	VM	16M	BCK	RES	RUN	DEP
√	√	OfficeVision® 1.2 or later	Y	F	ESA1.2.0, ESA2.3.0	370, XA, XC	N	N	MD	segment	PTF UL83596
√	√	ICKDSF R14 or later	Y	S	ESA1.2.0, ESA2.3.0	370, XA	N		MD	module	
√	√	DITTO 3.2.0 or later	M	R	ESA1.2.0, ESA2.3.0	370, XA	N		MD	segment	PTF UN18382 (which requires US94938)
√	√	ACF/VTAM® 3.4 or later	Y	F	ESA1.2.0, ESA2.3.0	370, XA, XC	N		MD	segment	

Column definitions:

Column **What it means**
OLD Check this column if you currently use this product in your old system.
NEW Check this column if you plan to use this product in your new system.
KP Keep — Must you keep this product (Yes, No, or Maybe)?
US Usage — How much is the product used (Frequently, Sometimes, or Rarely)?
REL VM releases it can run on that you are interested in. Note that some products or parts of products that are not supported may actually run sufficiently on the new system to use during conversion. If you use an unsupported product or release of a product, IBM will not provide service for that product or for problems in VM/ESA brought about by use of the product.
VM Virtual machine it can run in (370, XA, XC).
16M Can it run above the 16MB line?
BCK Can it run in a backlevel CMS on the new system?
RES Where does it reside, SFS or minidisk (MD)?
RUN Where does it run, segment, module, nucleus, or elsewhere?
DEP Dependencies, such as h/w, PTFs, pre-reqs, co-reqs.

System Applications

Following is an example of the type of information you should gather about your system applications. You can use an empty table provided for you in “System Applications” on page 631 to fill in this information. At the bottom of the table, called 'Column definitions,' is an explanation of what should go into each column.

Some of the information you may not be able to answer at this time.

OLD	NEW	Name and release level	KP	US	REL	VM	16M	BCK	RES	RUN	Time	DEP
√	√	TIMECDXA 3.1	Y	F	ESA1.2.0 ESA2.3.0?	XA, XC	Y	N	MD	seg	1wk (to test with new COBOL)	COBOL
√	√	SCHEDULE 2.0	Y	S	ESA1.2.0 ESA2.3.0?	XA, XC	Y	Y	MD	mod	1wk (to test with new SQL)	REXX compiler SQL/DS™ or DB2® for VM
√	√	PAYROLL	Y	S	ESA1.2.0 ESA2.3.0?	370	N	Y	MD	mod	1wk (to test)	COBOL SQL/DS or DB2 for VM
√	√	MIDTERM	Y	S	ESA1.2.0 ESA2.3.0?	370	N	Y	MD	mod	1wk (to test)	REXX compiler SQL/DS or DB2 for VM

Column definitions:

Column	**What it means**
OLD	Check this column if you currently use this program in your old system.
NEW	Check this column if you plan to use this program in your new system.
KP	Keep — Must you keep this program (Yes, No, or Maybe)?
US	Usage — Is the program used Frequently, Sometimes, or Rarely?
REL	VM releases it can run on that you are interested in.
VM	Virtual machine it can run in (370, XA, XC).
16M	Runs above the 16MB line?
BCK	Runs in a backlevel CMS on the new system?
RES	Where it resides, SFS or minidisk (MD).
RUN	Where it runs, segment or nucleus.
Time	Estimated time to make changes.
DEP	Dependencies, such as h/w, PTFs, pre-reqs, co-reqs.

Your Users' Applications

You can ask your users who write their own applications to fill in a table like the following one for each application they currently maintain themselves.

You can use a full-page table provided for you in “Your Users' Applications” on page 633 to fill in this information. At the bottom of the table, called 'Column definitions,' is an explanation of what should go into each column.

OLD	NEW	Name and release level	KP	US	REL	VM	16M	BCK	RES	RUN	Time	DEP

Column definitions:

Column	What it means
OLD	Check this column if you currently use this program in your old system.
NEW	Check this column if you plan to use this program in your new system.
KP	Keep — Must you keep this program (Yes, No, or Maybe)?
US	Usage — Is the program used Frequently, Sometimes, or Rarely?
REL	VM releases it can run on that you are interested in.
VM	Virtual machine it can run in (370, XA, XC).
16M	Runs above the 16MB line?
BCK	Runs in a backlevel CMS on the new system?
RES	Where it resides, SFS or minidisk (MD).
RUN	Where it runs, segment, module, or elsewhere.
Time	Estimated time to make changes.
DEP	Dependencies, such as h/w, PTFs, pre-reqs, co-reqs.

Local Mods

Following is an example of the type of information you should gather about your local modifications to VM. You can use an empty table provided for you in "Local Mods" on page 634 to fill in this information. At the bottom of the table, called 'Column definitions,' is an explanation of what should go into each column.

Name	Description	Modules or control blocks it affects	ESA	KP	US	Time
ASCII terminal translation tables	Specifies translation of ASCII terminal codes to corresponding EBCDIC codes and vice versa. **Note:** In the new system, the TRANSLATE_TABLE statement in the SYSTEM CONFIG file specifies file names of translation tables located on the system parm disk. This function can replace the local modification.	HCPTBN	Y	N	F	——

Column definitions:

Column	What it means
ESA	Can the modification be replaced by function in the new system?
KP	Keep — Must you keep this modification (Yes, No, or Maybe)?
US	Usage — Is the modification used Frequently, Sometimes, or Rarely?
Time	Estimated time to make changes.

Chapter 5. Determining Which Conversion Strategies You Want to Use

This chapter describes several conversion strategies and discusses the decisions you have to make. You can use it as a guide to help you determine which strategies are best for you and your environment.

The conversion strategy is broken into three parts:

1. System configuration
2. System application, program product, and system data
3. User.

But, these parts are not entirely separate entities. As you determine each part of your strategy, remember to consider the other parts as well.

As you determine a strategy, remember that you should first convert to the new system before exploiting its new functions. This helps if you need to backout your system, your applications, or your users from the new system. While converting, make note of improvements that you can make later, and return to them after you have completed the move to the new system.

The strategies discussed in this chapter are:

- **System configuration conversion strategies:**

  Based on the type and amount of hardware you have, the characteristics of your old system, and the requirements for your new system, you determine which system configuration strategy would work best for you. The system configuration conversion strategies discussed in this chapter are:

  - One processor with the new system as a guest of the old
  - Two processors -- with the old system on one processor and the new system on the other
  - One processor logically partitioned into two -- with the old system in one partition and the new system in the other
  - Cut and go -- which means to move directly from the old system to the new system.

- The strategy you choose for moving your applications and program products is probably closely tied with the strategy you chose for moving your users. Based on a combination of the characteristics of the applications and program products in your old system and the users that access them, you can determine which strategies work best for which applications and user groups. The strategies discussed in this chapter for converting programs and users are:

  - **System application, program product, and system data conversion strategies:**

    - On the old system before moving them to the new system
    - On the new system using the new CMS
    - On the new system using a backlevel CMS.

- **User conversion strategies:**
  - Staged by organization, by user type, or by application or program product
  - Everybody all at once.

For each part of the conversion strategy, this chapter:

1. Provides a work sheet that helps you take an initial evaluation of which strategies are best for you
2. Discusses the advantages and disadvantages of each strategy
3. Gives an overview of what each strategy would entail.

Determining Which Part of the Conversion Strategy to Plan First

The part of your strategy (system configuration, application or product, or user) you determine to work on first depends on what drives your conversion. To help you understand what drives your conversion, this section has you:

- Think about the general characteristics of your current system
- Analyze what drives your conversion.

Note: Determining which part of your conversion strategy to work on first does not mean that your conversion would actually take place in that order. For example, you may decide that your users are the driving force of your conversion, so you should determine your user conversion strategy first. But, when it comes to actually doing the conversion, you will not, of course, convert your users before converting your system configuration.

General Characteristics of Your Environment

Think about the general characteristics of your environment that are important to the conversion. You can use empty tables provided in “Tables for Information for System Configuration Strategies” on page 635 to write down this sort of information. Look at the samples in the two following tables and in “Examples of Determining What Order to Plan Your Strategies” on page 24 to get ideas of what to think about and how to fill in these tables. Also, use the information you filled in or thought about in Chapter 4, “Preparing Hardware and Software Inventory.”

“Tables for Information for System Configuration Strategies” on page 635 has a full-page table like the one shown below that you can use to note any general characteristics about your hardware that you think are important to the conversion strategy. Think about such things as:

- Can your processors be logically or physically partitioned?
- How much I/O (channels and DASD) do you have available?

Table 1. Sample of processors and I/O characteristics information

Processors and I/O characteristics
9121 -- currently logically partitioned into two ESA1.1 is in one partition; XA is in the other
A total of twelve 3380s and eight 3390s are available for both systems
A total of eight 3480 tapes drives, eight 3174 display control units, and one 4248 line printer are available for both systems

Table 165 on page 636 is a full-page table like the one below that you can use to note general characteristics about your users and their applications or products that are important to the conversion strategy. Try to group your users. Use the second column to indicate what you think the priority or the order is that the groups should be moved.

Table 2. Sample of user and application information

User and application/product group characteristics	Priority or order for conversion
Building 222 -- about 275 OfficeVision® users who use several accounting applications	3
Technical group -- about 100 users They have their own mathematical/scientific applications. They want to take advantage of data spaces for lots of read-only data.	2
Executives -- 3 users. They use OfficeVision. To make sure that these individuals are happy with the money they just spent on this new system, it must be as reliable as possible. So, these users will be moved last.	4
System support staff -- 5 users. That's us. We will move first to test the new system.	1

Think about What the Driving Force of Your Conversion Is

Based on the general characteristics that you jotted down in the tables above, think about what the driving force of your conversion should be. Look at “Examples of Determining What Order to Plan Your Strategies” on page 24 to get ideas of what to think about.

Fill in the first column of Table 3 based on your analysis of what the driving force for your conversion is. If there is no one thing that drives your conversion, or if your conversion is driven equally by all three aspects (system, applications, and users), you can (1) start with the system configuration strategy, (2) do your application/program product strategy next, (3) then work on your user strategy.

Use the references in Table 3 to determine where in this chapter to go to figure out your strategy. Go to those parts of this chapter in the order you specified in the priority column.

Table 3. Where to find information on the parts of conversion strategies

Priority (you fill in):	If your conversion is driven by:	See the following section:
	Your system configuration	“System Configuration Conversion Strategies” on page 26
	Your applications or program products	“System Application, Program Product, and System Data Conversion Strategies” on page 34
	Your users	“Strategies for Converting Your Users” on page 41

Examples of Determining What Order to Plan Your Strategies

The following sections show examples of how people who coordinate a conversion might determine the order that they should plan the parts of their conversion strategy. The order is based on what the driving forces of the conversion are.

Note: The priorities listed in these examples are just examples and are not meant to be 'recommended.' You have to determine your own priorities based on your environment.

Example 1 - User Groups and Applications Are Clearly Defined

Consider Katy's environment, which has the following characteristics:

Processors and I/O characteristics
9021 -- can be physically or logically partitioned
10 extra DASD volumes available
Enough I/O for two systems

User and application/product group characteristics	**Priority or order for conversion**
Group 1: Users who primarily use CATIA.**	4
Group 2: Users who primarily use several financial-type applications and program products.	3
Group 3: A group of scientific users who do a lot of calculations.	1
Group 4: Another big group of users who use a wide variety of applications and program products.	2

Because of the flexibility of Katy's system configuration, she could use any of the system configuration conversion strategies. Her users and their applications are clearly defined, and she considers them the driving force of her overall conversion strategy. So, she decides to figure out her overall conversion strategy in the following order:

1. User conversion strategy.

   Her users are clearly defined, and she feels she must figure out how to move them first.

2. Application conversion strategy.

   After she decides on the order that she would move her users, she needs to determine a strategy for moving the system applications and the program products they use. She decides to use a different application conversion strategy for each user group.

3. System configuration conversion strategy.

   Now she can pick a system conversion strategy that would work effectively with the user and application strategies she chose.

Example 2 - VM Used Primarily to Run Guest Operating Systems

Consider Jon's environment, which has the following characteristics:

Processors and I/O characteristics
9121 -- can be logically partitioned
5 extra DASD volumes available
Enough I/O for two systems

Jon himself is the only user directly on VM. Jon's installation uses VM to run VSE and MVS™ guests.

User and application/product group characteristics	**Priority or order for conversion**
VSE and MVS guests and users.	2
Jon, as the system programmer, personally uses a few applications and program products directly on VM.	1

Jon decides that, because his program and user conversion considerations are straightforward, his system configuration is the driving force of his conversion.

Example 3 - Limited Hardware

Consider Jeanne's environment, which has the following characteristics:

Processors and I/O characteristics
9221
No extra DASD volumes available*
Not enough I/O for two systems
Note: * Jeanne needs a few DASD volumes available during conversion. She will need to figure out how she wants to handle this situation. She may decide to purchase some, to borrow some temporarily from another system she supports, or to lease some.

Jeanne's users access a wide variety of applications and program products. She can categorize her users into two groups.

User and application/product group characteristics	**Priority or order for conversion**
Group 1: Local users -- this is her biggest group of users. She could subgroup them by: • Department, • The building that they are located in, or • The controller they are attached to.	2
Group 2: Remote users -- this is a small group of users.	1

Because Jeanne's hardware is limited, she decides that she is driven by her system configuration. She decides to figure out her overall conversion strategy in the following order:

1. System configuration conversion strategy.

   Her hardware is limited, so she feels she needs to figure out which system configuration strategy to use first.

2. Application and program product conversion strategy.

   Then she wants to decide how she should move the applications and program products.

3. User conversion strategy.

   Once she decides a strategy for converting applications and program products, she can figure out how she wants to subgroup her users for the move to the system.

System Configuration Conversion Strategies

The following are several system configuration conversion strategies that you can choose from; which one you choose depends on your needs:

1. One processor -- new system as a guest
2. Two processors (or one processor physically partitioned into two processors)
3. One processor with LPAR
4. Cut and go -- move directly from the old system to the new.

Note: There may be other strategies or combinations of strategies that would also work, but this chapter talks mostly about these four. “Example of Combining Strategies —— 'New System as a Guest' and 'Cut and Go'” on page 34 talks about combining the 'one processor with the new system as a guest on the old system' strategy with the 'cut and go' strategy.

Do the following to help you determine your system configuration conversion strategy:

- Use the worksheet in “Initial Evaluation of Your System Environment” on page 27 to help you make an initial evaluation of which system configuration conversion strategy may be best for your environment.
- Review the advantages and disadvantages of each strategy for converting your system configuration as discussed in “Advantages and Disadvantages of System Configuration Conversion Strategies” on page 28.
- Review the overviews of what each strategy would entail; these are described in “Overviews of What Each System Configuration Conversion Strategy Would Entail” on page 31.

When you have completed determining your system configuration conversion strategy, determine your other conversion strategies (if you have not done so already), in the order you previously decided:

- Converting your applications, program products, and system data (see page 34)
- Converting your users (see page 41)

Initial Evaluation of Your System Environment

Use the worksheet below to do an initial evaluation of your system environment. Keep in mind the items that you find are important to you when you review the advantages and disadvantages of each system configuration strategy (“Advantages and Disadvantages of System Configuration Conversion Strategies” on page 28).

To fill out this worksheet:

1. In the first column, rate how important the environmental characteristic is to you. For example:
   - `'very'`
   - `'some'` (probably cannot ignore, but not as important)
   - `'none'` or leave it blank to indicate that it is not important.
2. Look at the strategies that have diamonds in their columns. These are the strategies that would probably work best if the characteristic is important to you. Open diamonds (◊) indicate strategies that are pretty good for that particular characteristic. Closed diamonds (♦) indicate strategies that are best if the characteristic is important.

Table 4 (Page 1 of 2). Worksheet for selecting a system configuration conversion strategy. Open diamonds (◊) indicate strategies that may be good for that particular characteristic. Closed diamonds (♦) indicate strategies that are best if the characteristic is important.

Importance	Environmental Characteristic	Guest	Two	LPAR	Cut
	My system requires easy backout of programs, data, or users.	◊	◊	◊	
	My system requires good performance for programs and users still on the old system.	◊	♦	◊	
	I do NOT have enough I/O to support two systems.	◊			◊
	I do not have two processors available or cannot physically partition my processor into two.	♦		♦	◊
	My processor does not have LPAR capability.	♦	♦		◊
	I must have the absolute best system performance during conversion.		♦		◊
	I do not have or do not want to use extra hardware connections for communications between my old and new systems.	♦			
	I have extra hardware connections that I could use for communications between my old and new systems.		♦	♦	
	My new system or my old system requires good reliability.		♦	♦	
	I have experience with running VM as a guest of VM.	♦			
	I have no experience with LPAR.	◊	♦		
	I have lots of experience with LPAR.			♦	
	My environment is both a V=V production guest and a CMS interactive environment.	♦	♦	♦	
	My environment is both a preferred production guest and a CMS interactive environment.	♦	♦		
	My environment is primarily a V=V production guest environment.		♦	♦	♦

Table 4 (Page 2 of 2). Worksheet for selecting a system configuration conversion strategy. Open diamonds (◊) indicate strategies that may be good for that particular characteristic. Closed diamonds (♦) indicate strategies that are best if the characteristic is important.

Importance	Environmental Characteristic	Guest	Two	LPAR	Cut
	My environment is primarily a preferred production guest environment.		♦		♦
	My environment is small with few users, applications, and program products.				♦

Advantages and Disadvantages of System Configuration Conversion Strategies

Which conversion strategy you choose depends on the following considerations:

- Effects on application and user strategies
- Hardware you have available
- System performance you need during conversion
- Ease of connectivity between the new and old systems
- Reliability
- Guest or CMS interactive environment
- Complexity -- skills required
- Time frame or schedule needed

Effects on application and user strategies: Performance and the ability to backout are the major effects on application or user strategies.

Hardware you have available: For example, with some conversion strategies you cannot share system DASD volumes and other I/O devices. In these situations, you must have enough for both your old and your new systems. If you do not have a lot of I/O devices, you may need to use a strategy where you can share I/O.

Or, for example, some processors do not have logical partitioning (LPAR) capability. LPAR refers to the Processor Resource/Systems Manager™ (PR/SM™) hardware feature in LPAR mode. (The other PR/SM mode is basic mode.) If your processor does not have the PR/SM feature, you have to choose a scenario that does not use LPAR. For information about which processors support LPAR mode, see the VM/ESA Processor Support Matrix in the *VM/ESA: General Information* book or on the IBM VM operating system home page (http://www.ibm.com/s390/vm).

System performance needed during conversion: Some of your applications or program products may be sensitive to the system's performance. Some conversion strategies may negatively impact the performance of your old or your new system. If high performance is a necessity, you should consider those strategies that allow the best performance.

Ease of connectivity between the new and old systems: Some conversion strategies require real hardware connections to communicate between your new and your old systems.

Reliability: Some strategies have the potential to be less reliable during the conversion. Reliability of the old system could be impacted by the amount you change it in preparation for conversion. Reliability in the new system could be impacted by moving a lot from the old system without careful testing.

Guest or CMS Interactive Environment: Some strategies work better if your environment is mostly a production guest environment. For example, running multiple guests on VM in a logical partition may not perform well because logical partitions do not provide all the performance assists provided when running the guests on VM natively on the processor. Other strategies work better if your environment is mostly a CMS interactive environment.

Complexity -- skills required: All strategies are quite complex and require experienced VM system programmer skills. Each strategy, however, requires different types of skills.

Time frame or schedule needed: Some strategies can be used if you require a very aggressive conversion schedule. These scenarios are typically much more risky, and may end up taking longer in the long run. But, if you want to take the risk, they may allow you to meet your very aggressive schedule.

Table 5 (Page 1 of 3). Advantages and disadvantages of each system strategy

Strategy	Considerations	Advantages and Disadvantages
One processor -- new system as a guest	Application and user strategies you choose	Acceptable for any strategy. Backout of applications, program products, system data, or users, if required, may be easiest with this strategy. Some performance for applications and users still on the old system may be impacted.
	H/W you have available	Some additional DASD may be required, but, mostly you can share I/O between the old and new systems.
	Performance during conversion	May be impacted. Use V=R guest to minimally impact performance.
	Connectivity	Requires software products, such as TCP/IP, RSCS, PVM, or VTAM®, but no real hardware connections are required.
	Reliability	The guest system reliability depends on the reliability of the first level system.
	Skills required	Requires experienced VM system programmer skills and knowledge of running VM as a guest of VM.
	Guest or CMS interactive environment	Running your guests third level will impact performance.
	Time frame or schedule required	Requires a lot of time.

Table 5 (Page 2 of 3). Advantages and disadvantages of each system strategy

Strategy	Considerations	Advantages and Disadvantages
Two processors (or one processor physically partitioned into two)	Application and user strategies you choose	Acceptable for any strategy. Backout of applications, program products, system data, or users, if required, may be hard with this strategy. If you have shared data, such as a shared directory, this strategy may provide easier backout.
	H/W you have available	Dedicated I/O is required for each system, including system DASD, tapes, printers, terminals.
	Performance during conversion	No impact.
	Connectivity	Requires real hardware connections as well as connectivity software products, such as TCP/IP, RSCS, PVM, or VTAM.
	Reliability	Good -- reliability on the new system does not impact reliability on the old system, and vice versa.
	Skills required	Requires experienced VM system programmer skills. Also requires experience with hardware configurations.
	Guest or CMS interactive environment	Acceptable for both.
	Time frame or schedule required	Probably requires the longest amount of time.
One processor with LPAR	Application and user strategies you choose	Acceptable for any strategy. Backout of applications, program products, system data, or users, if required, may be hard with this strategy. If you have shared data, such as a shared directory, this strategy may provide easier backout.
	H/W you have available	Requires LPAR capability. Also, requires dedicated I/O for each system, including system DASD, tapes, printers, terminals.
	Performance during conversion	Some impact. Impact to guests of VM in an LPAR may be considerable.
	Connectivity	Requires real hardware connections as well as connectivity software products, such as TCP/IP, RSCS, PVM, or VTAM.
	Reliability	Good -- reliability on the new system does not impact reliability on the old system, and vice versa.
	Skills required	Requires experienced VM system programmer skills. Also requires experience with LPAR and hardware configurations.
	Guest or CMS interactive environment	Acceptable for CMS interactive. In general, guest operating systems other than CMS running in VM in an LPAR are supported only for test and development. Preferred guests and multiple guests on VM in an LPAR may have performance impacts.
	Time frame or schedule required	Requires a considerable amount of time; depends on the level of connectivity needed between the old and new systems. If you need easy connectivity, you spend more effort setting it up, but it is convenient during the conversion.

Table 5 (Page 3 of 3). Advantages and disadvantages of each system strategy

Strategy	Considerations	Advantages and Disadvantages
Cut and go -- move directly from the old release to the new release	Application and user strategies you choose	Backout of applications, program products, system data, or users, if required, is very difficult with this strategy.
	H/W you have available	No impact.
	Performance during conversion	No impact.
	Connectivity	Not applicable.
	Reliability	Can be risky -- you must have a good backout plan.
	Skills required	Requires experienced VM system programmer skills.
	Guest or CMS interactive environment	Recommended for a pure guest environment.
	Time frame or schedule required	Quick.

Overviews of What Each System Configuration Conversion Strategy Would Entail

As you see in the previous section, the advantages and disadvantages are different among the system conversion strategies. However, the overall conversion approaches for the following strategies are similar:

- One processor with the new system as a guest on the old system
- Two processors
- One processor physically or logically partitioned into two.

The 'cut and go' strategy is described in “Conversion Approach for Cut and Go” on page 33.

Conversion Approach Using One or Two Processors

The following procedure shows the conversion approach using one or two processors.

1. Make sure you have enough processor and storage capacity.

   If your old system is processor-constrained, you may need to upgrade your equipment. Also consider whether you have sufficient storage (real and expanded) and DASD. Make sure you understand the performance considerations for new hardware.

2. Install and test a base VM/ESA Version 2 Release 4.0 system.

   Use this base system to test and become familiar with the new system.

   For 'one processor with the new system as a guest':

   Install and IPL the new system as a V=R preferred guest on the old system.

   For 'two processors or one processor physically or logically partitioned':

   Install and IPL a base VM/ESA Version 2 Release 4.0 system. IPL this system on a separate processor, on one side of a physically partitioned processor, or in a logical partition.

3. Reconfigure the new system in preparation for production. For example:

   a. Add any extra DASD you need (if they are defined in your IOCP already and if you use the new CP configurability function you do not have to re-IPL).
   b. Increase storage for spool space, paging space, temporary disk space, and so on, as needed for your new production system.

4. Test the new system.

   Before you convert any CMS users, you should sufficiently test the new system. Once users convert their applications and begin to exploit the new system's CMS, you will have a harder time backing out. To ensure that the system is reliable, you may want to run the new system for a period of time before you move users.

   During this time, you can:

   - Train the data processing and operations staff
   - Install and test selected program products
   - Install users' converted applications that are critical
   - Convert your system applications
   - Convert the programmable operator and other automated functions.

5. Save the old system's spool files and desired system data files, and restore them on the new system, if needed.

   For example, use the SPTAPE command (use the SPXTAPE command when migrating from VM/ESA 1.2.2 or later) to save the old system's spool files and system data files on tape and to move them to the new system.

   Note: There are several ways to move files, such as spool files, between your old and new systems. For example, you could use tape, communications connections, or download and upload the files to and from a personal computer.

6. Move users and their applications to the new system.

   How exactly you do this depends on the application and user conversion strategies that you choose.

7. Store your old system on tape for back up.

8. **For 'one processor with the new system as a guest':**

   a. IPL your new system first level.
   b. Bring up your old system as a guest on the new system, if you want.

   Note: You can immediately migrate spool files by using the same DASD allocation for spool space on the new system as your old system had.

9. Remove your old system -- if you want:

   For 'one processor with the new system as a guest':

   a. Make sure all users and data are on the new system.
   b. Shut down your old system.
   c. Make the DASD you used for the guest available.

   For 'two processors or one processor physically or logically partitioned':

   a. Make sure all users and data are moved to the new system.
   b. Shut down the new system and your old systems.
   c. Reconfigure the processor so that there is no more partitioning.

d. Make the DASD you used for the your old system available.
e. Bring up VM/ESA Version 2 Release 4.0 as the production system.

Considerations of Logical Partitioning

ES/9000® processors support logical partitioning. To be able to logically partition an ES/3090™, you must have the Processor Resource/Systems Manager (PR/SM) feature operating in logically partitioned (LPAR) mode. PR/SM is a hardware feature that allows you to run separate systems, such as old and new systems, in independent logical partitions that are active on the same processor complex at the same time.

Note these considerations:

- The CPUID that appears in most EREP records is not the same as the CPUID of the physical processor on which the error occurred (because there is more than one partition per physical processor in LPAR mode).
- The address-limit-checking facility is not available in a logical partition, and therefore CCW translation is always required. As a result, the new system's SET CCWTRANS OFF and SET NOTRANS ON commands have no effect in LPAR mode.
- The start interpretive-execution assist is not available in a logical partition, and the new system's SET IOASSIST ON command has no effect in LPAR mode. This is why:
  - VM/ESA 2.4.0 is supported in LPAR mode for development, test, and CMS production only, not for guest production.
  - V=R guests are supported only for test and development.

Conversion Approach for Cut and Go

You probably would want to consider this strategy as an option if your environment:

- Is a guest-only environment
- Is simple; has few applications that all users work on, does not have local mods, and so on.

Note: The approach shown here would need to change if you have an old processor that you need to replace. You would have to combine the 'two processor' and the 'cut and go' approaches.

A 'cut and go' conversion approach using a single processor might look like this:

1. Make sure you have enough processor and storage capacity.

   If your existing system is constrained, you may need to upgrade your equipment.

2. Back up your old system.

   a. Move any data left on your old system to your new system.
   b. Store your old system on tape for back up.

3. Bring up and test a VM/ESA Version 2 Release 4.0 system.

   Note: Throughout this step you must work when your users are not on the system, possibly during the night or on a weekend. You have to re-IPL your old system for production work until you are satisfied with the new system's reliability. (This is sometimes called a flip-flop conversion.)

   a. Bring down your old system.

   b. Reconfigure your I/O if needed.
   c. Re-IML your processor in ESA/370 or ESA/390 mode.
   d. Install a base VM/ESA Version 2 Release 4.0 system on your existing processor.
   e. Use this base VM/ESA Version 2 Release 4.0 system to test and become familiar with the new system.

4. Save the old systems' spool files and desired system data files, and restore them on the new system, if needed.

   For example, use the SPTAPE command (use the SPXTAPE command when migrating from VM/ESA 1.2.2 or later) to save your old system's spool files and system data files on tape and to move them to the new system.

5. Move guests, users, and applications to the new system.

   How exactly you do this depends on your overall conversion strategy.

Example of Combining Strategies —— 'New System as a Guest' and 'Cut and Go'

If your system is mostly guests, then you might choose the 'cut and go' strategy. If you would feel safer doing some testing of the new system before doing the cut and go, you could combine the 'new system as a guest' strategy with the 'cut and go' strategy. Your conversion approach might be the following:

1. Install the new system as a guest on the old system.

2. Cut over to the new system:

   a. Move your guests and applications to the new system.
   b. Move your users to the new system.

3. Bring your new system up first level. You can re-IPL using the address of the DASD you installed your second level system on. See “Converting Spool Files and Saved Segments [ESA]” on page 91 for information on moving spool files.

4. Free the DASD that was running your old system.

System Application, Program Product, and System Data Conversion Strategies

Do the following to help you determine your conversion strategy for your applications, program products, and system data:

1. Use the worksheet in “Initial Evaluation of Which Application and Program Product Conversion Strategy Works Best for You” on page 35 to help you determine which strategy would be best for your environment.

2. Review the advantages of each approach (“Advantages and Disadvantages of System Application, Program Product, and System Data Conversion Strategies” on page 36):

   a. On the old system before moving them to the new system
   b. On the new system using the new CMS
   c. On the new system using a backlevel CMS.

   Note: A combination of some of the above strategies may actually be the best strategy for you. You must understand your user population and how they use their applications.

3. Review the overviews of each approach in “Overview of What Application, Program Product, and Data Conversion Strategies Would Entail” on page 39 to understand what they would entail.

When you have completed determining your strategy for converting your applications, program products, and system data, determine your other conversion strategies (if you have not done so already), in the order you previously decided:

- Converting your system configuration (see page 26)
- Converting your users (see page 41)

Initial Evaluation of Which Application and Program Product Conversion Strategy Works Best for You

Use the worksheet below to do an initial evaluation of what application and program product strategies might work best for you. You may want to fill out a separate worksheet for each of the user and application/product groups that you listed in Table 165 on page 636.

Keep in mind the items that you find are important to you when you review the advantages and disadvantages of each application and program product conversion strategy (“Advantages and Disadvantages of System Application, Program Product, and System Data Conversion Strategies” on page 36).

To fill out the worksheet:

1. In the first column, rate how important the characteristic is to you. For example:
   - 'very'
   - 'some'
   - 'little' (cannot ignore, but not as important)
   - 'none' or leave it blank to indicate that it is not important.
2. Look at the strategies that have diamonds in their columns. These are the strategies that would probably work best if the characteristic is important to you. Open diamonds (◊) indicate strategies that are pretty good for that particular characteristic. Closed diamonds (♦) indicate strategies that are best if the characteristic is important.

Table 6 (Page 1 of 2). Worksheet for selecting an application/program product conversion strategy. Open diamonds (◊) indicate strategies that may be good for that particular characteristic. Closed diamonds (♦) indicate strategies that are best if the characteristic is important.

Importance	Characteristic	On old sys	Back CMS	On new sys
	I would like to be able to test on the new system immediately.		◊	♦
	I would like my users to get accustomed to changes while still on the old system.	♦		
	I want my users still on the old system to see no changes in how programs run.		♦	♦
	I would like to be able to test as much as I can on the old system, even if I cannot fully test it yet.	♦	◊	

Table 6 (Page 2 of 2). Worksheet for selecting an application/program product conversion strategy. Open diamonds (◊) indicate strategies that may be good for that particular characteristic. Closed diamonds (♦) indicate strategies that are best if the characteristic is important.

Importance	Characteristic	On old sys	Back CMS	On new sys
	I have some applications that I need or would like to keep on an old CMS. I don't want to convert these, but I want the users to move to the new system.		♦	
	I would like to test the new facilities of the new system (such as XC virtual machines) immediately.		◊	♦
	I would like to test program products that newly exploit XA, XC, or running above the 16MB line.			♦
	I would like to convert my applications but be sure that they still work on the old system.	♦	◊	
	I do not want to apply any PTFs to my old system to support the conversion. (I want to keep the current level of reliability in my old system.)	◊		♦
	I do not mind risking the reliability of my applications by moving them to the new system to convert them there.			♦
	I do not mind installing products and converting applications twice. (I would rather spread the conversion out over time and have it be more gradual.)	♦	◊	
	I want to install products and convert applications all at once right on the new system.			♦
	I would like to stage my conversion -- CP first, then CMS.		♦	
	I have a flexible time frame. I would rather spread the conversion of my applications out over time.	♦	♦	
	I need to or want to make my conversion to the new system as quickly as possible. I would rather spend the time on the new system working out all the bugs.		◊	♦

Advantages and Disadvantages of System Application, Program Product, and System Data Conversion Strategies

Which conversion approach you choose depends on the following considerations:

- Which system strategy you choose
- User impact
- Where your applications, program products, or data can work
- Reliability required
- Work load to convert applications and install program products
- The time frame or schedule required

Which system strategy you choose: Some application or data conversion strategies might not work efficiently with the system conversion strategy you select in the previous section. If you have already decided on a system strategy and find this to be the case, you may need to go back to “System Configuration Conversion Strategies” on page 26 to re-evaluate which strategy would work best for you.

User impact: Some strategies may cause more disruption for the user.

Where your applications and program products run and where data resides: The conversion approach you choose depends on whether your environment can allow applications, program products, or data to be easily converted to a level that runs in the new system.

Another consideration for applications and program products is whether they can work:

- In an XA or XC virtual machine
- Above the 16MB line.
- In an XA virtual machine with 370ACCOM feature on for 370 applications.

If you have not yet figured out where some of your applications, program products, or data would work, you may want to experiment in the new system by just trying things out. See “Overview of What Application, Program Product, and Data Conversion Strategies Would Entail” on page 39 for information on testing applications on the new system.

Reliability required: The conversion approach you choose also depends on how critical it is that applications or program products are reliable in the old or the new environment.

Work load to convert applications or install program products: You need to consider how much effort you can spend to convert your applications or install program products. For example, some strategies work better if you can afford the effort it would take to make your applications work in both the old system and the new system.

The time frame or schedule required: You need to consider how much time you can spend to convert your applications. Some approaches allow you to cut over quickly, then you can exploit or upgrade as you need or want.

Application, Program Product, Data Conversion Strategy	Considerations	Advantages and Disadvantages
Convert on the old system before moving to the new	Which system strategy you choose	Acceptable for all system strategies. If you have not created your new system when you are converting your old applications, you will not be able to test whether it runs on the new system.
	User impact	If you get the users accustomed to the changes while they are still on the old system, there is less impact when they move to the new system. When users move, the bugs may not be completely ironed out if you do not test applications on the new system beforehand.
	Where your applications and program products run	Test may be limited. If you have not created your new system, you can only verify if the converted application still works on the old system.
	Reliability required	You may have to apply PTFs to your old system to support the new levels of program products or to bring your CMS up to the most current level.
	Work load to convert applications and install products	You may end up converting applications or installing program products twice -- once on the old system, then again on the new system. You can do some conversion before bringing up the new system. Testing needs to be done in the new system.
	Time frame or schedule required	You need a flexible time frame; this strategy may take longer.
Convert on the new system using the new CMS	Which system strategy you choose	You can use this with all system strategies.
	User impact	When users first move to the new system, they may encounter many changes.
	Where your applications and program products run	There is an impact if there are dependencies on things that have changed (such as, changed command interfaces or changed control blocks). You can test whether your applications work in new features of the system, such as in XC virtual machines.
	Reliability required	You do not impact the reliability of the old system because you are not required to apply PTFs; but reliability may be risky in the new system.
	Work load to convert applications or install products	You should have to convert applications or install program products only once -- in the new system. Your test effort is in the new system.
	Time frame or schedule required	Your move to the new system is quicker, but you may spend more time working out problems in the new system.

Application, Program Product, Data Conversion Strategy	Considerations	Advantages and Disadvantages
Convert on the new system using backlevel CMS	Which system strategy you choose	Not applicable for cut and go.
	User impact	If users have to switch between the old CMS and the new CMS to run different applications, this may be an inconvenience.
	Where your applications and program products run	You can keep the old applications running in production while testing and converting on the new CMS. If your applications have CP dependencies, they may not run in the backlevel CMS.
	Reliability required	You may have to apply PTFs to your old CMS to run it as a backlevel CMS on the new system.
	Work load to convert applications and install products	Allows for a staged conversion, CP first, then CMS. You avoid many re-installs or re-conversions.
	Time frame or schedule required	You can spread out the application conversion time.

Overview of What Application, Program Product, and Data Conversion Strategies Would Entail

You may want to use a different strategy for each application or program product. Your overall approach may look like this:

1. Understand where your applications and program products work.
2. Decide how you want them to work in the new system.
3. Analyze what changes they will need.
4. Convert them using one of the strategies:
   - On your new system using your new CMS
   - On your new system using a backlevel CMS
   - On your old system.

An approach you might use is:

1. Determine which applications and program products are needed in the new system. Keep in mind other products that your applications or program products require to run.

   Look at what you filled out in Chapter 4, “Preparing Hardware and Software Inventory” on page 13. Also, if you find more information, fill it in.

   If there are applications or products no longer used at your installation, get rid of them or archive them. But, be careful about destroying them. Some users may not know they are dependant on certain applications or products. Also, some applications get used very seldom but are critical to the business.

2. Understand where your programs products run. Determine which program products can:

   a. Run in both the old and the new system
   b. Run in a backlevel CMS on the new system
   c. Run only in the old system.

You can use the *Licensed Products Migration Matrix for VM/ESA or WHAT IS SUPPORTED WHERE for VM* and the *Vendor Products List* to determine where a program product is supported.

In some instances, although a product is not supported, it may still run sufficiently for you to use in the new system directly or in a backlevel CMS during your conversion. Note that if you use a program product in an unsupported VM environment, IBM will not provide service for that product or for problems in VM/ESA brought about by use of the product.

The *Licensed Products Migration Matrix for VM/ESA or WHAT IS SUPPORTED WHERE for VM* lists IBM licensed products with very detailed support information about each, such as:

- What VM products and releases they are supported in
- Which types of virtual machines they are supported in (370, XA, or XC)
- PTFs required to support them.

The *Vendor Products List* is not as detailed as the *Licensed Products Migration Matrix for VM/ESA or WHAT IS SUPPORTED WHERE for VM*. It lists vendor products and their release numbers that run in the new system.

3. Understand and keep notes by filling out more details in the tables in Chapter 4, “Preparing Hardware and Software Inventory” on page 13 about where your applications can run and where data associated with the applications can reside. You can do the following, keeping track of where each application can run:

   a. Move copies of your applications onto your new CMS. Try running them in the following environments:

      1) In an XC virtual machine above the 16MB line.
      2) In an XA virtual machine above the 16MB line.
      3) In an XC virtual machine below the 16MB line.
      4) In an XA virtual machine below the 16MB line.
      5) In an XA virtual machine with 370ACCOM feature on for 370 applications.

      You should decide later where you actually want to run them based on your overall conversion strategy.

   b. For the applications that do not run on the new CMS at all, running them in a backlevel CMS running on the new system.

      Note: Applications that have CP or other non-CMS dependencies may not run in a VM/ESA 2.4.0 or a backlevel CMS.

4. For applications that do not run in the new CMS, determine what changes you would need to make. You can use the VM/ESA MIGR tool to help you understand where the incompatibilities are in your REXX and EXEC 2 programs.

5. Decide how and when you want to convert each application.

   The ideal would be for all applications to be able to run in an XC or an XA virtual machine above the 16MB line. If this is not possible for you to do immediately, consider how, when, and with which group of users you should convert each application.

   Decide on the following possibilities:

- Converting the application to run in an XA machine with 370ACCOM feature on. Later you can convert it to run in an XA or XC virtual machine.
- Converting the application to run in an XA or XC virtual machine.
- Dual pathing the application so it works in both the old and new systems.
- Converting the application in your old system before you move, or converting it in the new system.

6. Apply the PTFs to your old VM system that you need:
   - To allow your old CMS to run in the new system.
   - For application conversion.
7. Convert and move your applications based on what you decided in step 5 on page 40.
   a. Bring up your backlevel CMS in your new system, if appropriate. Move applications to your backlevel CMS that you want there.

   b. On the old system, convert and test any applications that you wanted to. When you are ready, move them to the new system. Test them on the new system.

   c. Move the applications that you want converted on the new system to the new system. Convert and test them on the new system.
8. Move users to the new system. How you do this depends on the user conversion strategy you choose.

Strategies for Converting Your Users

The following are several strategies for converting users you can choose from; which one you choose depends on your needs:

1. Staged by organization, by user type, or by application or product
2. Everybody all at once.

Note: There may be other strategies that would also work, but this chapter talks mostly about these.

Do the following to help you determine your conversion strategy for your users:

1. Use the worksheet in “An Initial Evaluation of Which User Conversion Strategy Might Work Best for You” on page 42 to take an initial evaluation of which user conversion strategy would work best for your environment.
2. Review the advantages and disadvantages of each approach (“Advantages and Disadvantages of Strategies for Converting Users” on page 43):
   a. Staged by organization, by user type, or by application or product
   b. Everybody all at once.
3. Review the overviews of each approach in “Overviews of What Conversion Strategies for Users Would Entail” on page 45 to understand what they would entail.

When you have completed determining your strategy for converting your users, determine your other conversion strategies (if you have not done so already), in the order you previously decided:

- Converting your system configuration (see page 26)
- Converting your applications, program products, and system data (see page 34)

An Initial Evaluation of Which User Conversion Strategy Might Work Best for You

Use the worksheet below to do an initial evaluation of what user conversion strategy might work best for you. You may want to fill out a separate worksheet for each of the user and application/product groups that you listed in Table 165 on page 636.

Keep in mind the items that you find are important to you when you review the advantages and disadvantages of each application and program product conversion strategy ("Advantages and Disadvantages of Strategies for Converting Users" on page 43).

To fill out the worksheet:

1. In the first column, rate how important the characteristic is to you. For example:
   - `'very'`
   - `'some'`
   - `'little'` (cannot ignore, but not as important)
   - `'none'` or leave it blank to indicate that it is not important.
2. Look at the strategies that have diamonds in their columns. These are the strategies that would probably work best if the characteristic is important to you. Closed diamonds (♦) indicate strategies that are best if the characteristic is important.

Table 7 (Page 1 of 2). Worksheet for selecting a user conversion strategy. Closed diamonds (♦) indicate strategies that are best if the characteristic is important.

Importance	Characteristic	Stage	All at once
	I would like to minimize the impact to my overall user community by having a few at a time experience the new system. The first few that move can help me iron out things I had forgotten.	♦	
	I have a small straightforward user population.		♦
	My VM is used primarily to support guest operating systems.		♦
	The reliability of some of my applications or products is critical. I'd rather have the reliability of the new system ironed out by less critical applications or products and their users testing the new environment before I move my critical ones. Or, I'd rather get the conversion of my critical applications or products over with first.	♦	
	I would rather have reliability (of the system or converted applications) not impact my entire user population at once.	♦	
	I would prefer to convert my users very quickly.		♦

Table 7 (Page 2 of 2). Worksheet for selecting a user conversion strategy. Closed diamonds (♦) indicate strategies that are best if the characteristic is important.

Importance	Characteristic	Stage	All at once
	I do not mind having to maintain two levels of users——some on the new system, some on the old. (I'd rather do this to do a gradual safe conversion of my users.)	♦	
	I would rather spread the effort of handling user problems out over time.	♦	
	I do not want to maintain two levels of users. (I'd rather take the reliability risks.)		♦

Advantages and Disadvantages of Strategies for Converting Users

Which conversion approach you choose depends on the following considerations:

1. Which system and application conversion strategies you choose
2. What the users do and what applications they use
3. Security
4. Reliability
5. User impact
6. The time frame or schedule required
7. Work load

Which system and application conversion strategies you choose: Some system strategies can impact performance of users and their applications and the ability to backout easily. Application and user conversion strategies should be considered together based on your users and their applications.

What the users do and what applications they use: If you can determine what your users do and which applications they use, it may help you to determine how to stage the conversion of your users. For example, if you have a set of users that run specific applications, it may be beneficial to convert these users together.

Security: If your environment requires careful consideration of security, then you may find that some strategies help you better maintain security during conversion.

Reliability: Some of your end users may require a certain level of reliability. You may want to consider keeping these users together. Also, the reliability of the system may have a different level of impact depending on the user conversion strategy you choose.

User impact: Depending on your user environment, some strategies might help minimize the impact of the conversion on your users.

The time frame or schedule required: Some strategies allow for a gradual, careful conversion of users. In other cases, the time frame required for your conversion may not allow for a gradual conversion of users.

Work load: The overall total work load may be similar for all the strategies. Some strategies, however, may require a heavier work load up front while others require a heavier work load later in the conversion.

Strategy	Considerations	Advantages and Disadvantages
Staged by organization, by user type, or by application	Which system and application conversion strategies you choose	Some system strategies can impact performance of users and their applications and the ability to backout easily. **New system as a guest** Backout of applications, products, data, and users is easiest with this strategy. Performance is not optimum. **Two processors or LPAR** Harder to backout. **Cut and go** Not applicable.
	What the users do and the applications they use	You can decide on the breakdown of your groups by characteristics such as, department, location, or the applications they use. Or, you do not have to group them by any specific characteristics. You can have the first group be people who use few applications to start with, and then build up. Or, you can have the first group be people who use many applications common to the rest of user community. They would iron out many of the bugs of many of the converted applications for you.
	Security	You can move groups by their security classification, or you can move groups using the same secure applications and data.
	Reliability	You only affect one group of users at a time, and they can help test the reliability of the new system. If reliability of certain applications is important, you can stage your conversion such that these applications are moved to the new system later, after reliability of the new system has been tested. If reliability of an application becomes a problem, you can backout just those users who use that application.
	User impact	Impact is on a smaller group at any one time. The first to move may be impacted most.
	Time frame or schedule required	Allows for a gradual and relatively safe conversion.
	Your work load	The effort to maintain two levels of users may be significant. For example, you may have to maintain two directories. However, you can gradually 'break in' the system, and the effort to handle user problems during the conversion is spread out over time. If a backout is required during conversion, you can backout a group at a time.

Strategy	Considerations	Advantages and Disadvantages
Everybody all at once	Which system and application conversion strategies you choose	Some system strategies can impact performance of users and their applications and the ability to backout easily. **New system as a guest** Backout of applications, products, data, and users is easiest with this strategy. Performance is not optimum. **Two processors or LPAR** Harder to backout. **Cut and go** Not applicable.
	What the users do and the applications they use	It is not necessary to classify the users and their applications with this strategy. This strategy probably makes the most sense if you have a small user community or you use VM to support guests.
	Security	No impact -- same as the old system.
	Reliability	Any problems with reliability impacts many users at once.
	User impact	May have a high impact on all users.
	Time frame or schedule required	Allows for a very quick conversion.
	Your work load	The effort to move the users may be less to start with. For example, you do not have maintain two directories. However, user support during the conversion time may be considerable. Also, if a backout is required during conversion, it would be necessary to re-do everything.

Overviews of What Conversion Strategies for Users Would Entail

Before you can determine a strategy for converting users, you must understand:

- How they could be grouped
- What applications and products they use
- What the priority should be for each group to move to the new system.

Once you understand who your groups of users are, what applications and products they use, and when they should or can be moved, you can determine which conversion strategy would be most effective for your environment: (1) everybody all at once, or (2) staged by organization, user type, or application.

Note: If you have some users who cannot go to the new system because they need functions that are not available, you may need to maintain your new system as a guest and complete the conversion at a later release of VM/ESA.

Staging the Conversion by Organization, by User Type, or by Application or Program Product

In this situation, you may have a large number of users who can be subgrouped. For example, you may be able to group a set of your users because they:

- All use the same application or program product
- Are in the same department
- Work on the same project
- Work in the same building

- Work at a remote location
- Have the same size virtual machine
- Are connected to the same controller.

Whatever your reason for grouping these people, your approach may look like this:

1. **Prepare the applications and products for the first group for use on the new system.**
2. **Move the first group of users to the new system.**

   Have them convert and test their own personal application programs on the new CMS.
3. **Once the first group is stable, begin moving the next group, and so on.**

   For example, going back to "Example 1 - User Groups and Applications Are Clearly Defined" on page 24, the users were grouped and prioritized like this:

Group	Characteristic	Priority / order
1	Primarily use CATIA	4
2	Primarily use several financial-type applications and program products	3
3	A group of scientific users who do a lot of calculations.	1
4	Another big group of users who use a wide variety of applications and program products	2

Note: The priorities are just examples and are not meant to be 'recommended.' You would have to determine your own priorities based on your environment.

The approach Katy (the system programmer in this example) would use for this example would be:

a. She converts the applications used in group 3, the scientific users.
b. She moves group 3 to the new system.
c. She converts all the applications and program products used by group 4, the miscellaneous group.
d. She moves group 4. (She could decide to stage group 4's conversion. For example, she could decide to move them by department. Then she could move a subset of group 4 users at a time. The first subset could be the pilot users. They could test the system applications and program products on the new system. Then, as the bugs get fixed, she could move more subsets of users from group 4.)
e. She converts all the financial applications and program products used by group 2, the financial people.
f. She moves group 2 to the new system.
g. She converts CATIA last.
h. She moves group 1, the CATIA users, to the new system.

Everybody All at Once

This approach should be considered only if:

- You have a small amount of users doing the same thing or using few applications, or
- Your VM system is used for running guest operating systems.

Your approach could be:

1. Get all the applications and program products you need working on the new system.
2. Move all of your users to the new system.

Chapter 6. Create a Detailed Conversion Plan

A conversion plan is the key to a successful move to the new system. Your plan should provide a detailed map of the activities needed to complete a successful conversion.

This chapter first talks about developing a statement of purpose and a schedule for your conversion. The rest of the chapter is divided into three categories based on the conversion strategies discussed in Chapter 5, “Determining Which Conversion Strategies You Want to Use” on page 21. They include:

- Planning your system configuration conversion strategy
- Planning your application, program product, and data conversion strategy
- Planning your user conversion strategy.

Within each of these categories are key conversion tasks you should consider during the planning phase. To give you an idea of how these conversion tasks fit in with system configuration, application, and user conversion strategies, see Table 8.

Table 8. Key Conversion Tasks

When Planning Your:	**Consider:**
System Configuration Conversion Strategy	• Hardware and software installation • Hardware I/O • System definition files • User directory • Spool files • Connectivity • Performance • System configuration testing • Backing out your system.
Application, Program Product, and Data Conversion Strategy	• Saved segment layout • Guest operating systems • Backout plans.
User Conversion Strategy	• User education - public relations • User backout plan.

For each conversion task discussed in this chapter we:

- List information you should gather or topics you should educate yourself on before planning for the task. Many of these sections are geared towards conversions from VM/ESA Version 1 Release 2.0, but many of the same concepts can be applied to the other VM/ESA conversions.
- Discuss what to include in your plan for each task.

When your plan is complete it will include:

- A statement of purpose for your conversion.
- A schedule of what conversion tasks you need to perform, when they will be done, and who will do them.

- Documentation of your conversion strategies as discussed in Chapter 5, “Determining Which Conversion Strategies You Want to Use” on page 21.
- Documentation for the appropriate conversion tasks listed in Table 8 on page 49.

Develop a Statement of Purpose and Gather Requirements for Your Conversion

A statement of purpose helps you set realistic goals for your conversion while meeting key requirements for your installation. It also helps other people in your installation understand why you are moving to the new system.

Before You Document Your Statement of Purpose

Before you document your statement of purpose, you should make a list of the conversion goals and requirements for your installation. While you are collecting your goals and requirements, prioritize the list so you know what is most important to your installation. This can help you choose a conversion strategy that meets all, or at least your most important, goals and requirements. As you set your goals, remember that you should first convert to the new system. Once your conversion is complete, you can then exploit new functions.

The main source for goals and requirements is your users. You should survey your users, including management, system programmers, application programmers, operators, and end users, to determine their needs and wants.

Requirements and goals of your conversion may include:

- Technical requirements - What are the technical goals and requirements of the new system? Some examples of these may be:
  - Remove storage constraints for applications
  - 99% system availability for end users
  - Conversion process is as transparent as possible for your end users
  - All existing applications must run in a VM/ESA 2.4.0 CMS
- Schedule - When do you expect the conversion to be complete? You may also have a date when the new system must be in production. We discuss this topic in more detail in “Develop a Conversion Schedule” on page 51.
- Human resources - How many people do you need to perform the conversion? It is best to have one person coordinate the conversion. The number of other people involved varies from installation to installation and is based on the complexity of the system, the experience of the personnel, the number of applications, and many other factors. Remember to have backups for key people.
- Cost - What is the acceptable limit to the conversion expense?
- System capacity - What is the maximum work load given current resources? Are additional resources needed?

What to Document in Your Plan

- A statement of purpose for your conversion, including a prioritized list of conversion goals and requirements.
- When you will gather and document the goals and requirements.

Develop a Conversion Schedule

As you read through this chapter keep in mind your overall schedule and how each conversion task can fit into it. Write down dates when you think tasks can be done and identify tasks that can be overlapped in your schedule. Also, document who will perform each conversion task.

Before You Document Your Conversion Schedule

Before you document your schedule, make sure you understand the requirements from “Develop a Statement of Purpose and Gather Requirements for Your Conversion” on page 50. Then, decide on the overall start and end dates for your conversion. Once these dates are set, you can fill in the specific conversion tasks and dates as you read through the rest of this chapter.

Items that Can Affect Task Time

All of the conversion tasks discussed in this chapter affect your conversion schedule. The key factors include:

Factors	Effects
System usage	If your system is running near 100%, 24 hours a day, all year, it is difficult to schedule your conversion at a time that will not upset your production system.
Your installation's calendar	If you have many days when certain applications must be running or times when the system must be available, it is more difficult to plan your conversion.
System configuration conversion strategy you plan to use	Some conversion strategies discussed in Chapter 5, “Determining Which Conversion Strategies You Want to Use” on page 21 take longer to implement and convert to the new system. Cut and go may be the quickest of the system conversion strategies but, in most cases, should be used for a guest environment or at an installation with few applications. If you choose to use an LPAR or physically partitioned processor, your conversion may not take as long if you already have them set up in your old environment.
Number of system configuration conversion strategies you plan to use	If you plan to use a combination of strategies, this may add to your total conversion time. For example, it may take longer to implement both the LPAR and second level conversion strategies.
Number of applications, number of program products, and amount of data to convert.	If you have lots of applications that are dependant on VM internals, message or response text, saved segment layout, and CMS storage layout, it will take longer to convert these applications.

Factors	Effects
Number of users at your installation	The more people you have to convert to the new system and the more applications and program products they use, the longer it takes to convert to the new system.

What to Document in Your Plan

- Start date for your conversion
- End date for your conversion
- Key days or dates when you cannot afford to have the system down or when certain applications must be available.

Sample Conversion Schedule

As an example, Lynne is a system programmer at a university. She plans to convert from VM/ESA 1.2.0 to VM/ESA 2.4.0 during the first part of the year. Her production schedule from December to February looks like the one in Figure 1 on page 53.

Note: This is just an example, and therefore the dates are not current.

Lynne's environment includes:

- A 9121 model 480 logically partitioned into two: the VM/ESA 1.2.0 system is running in one partition and the other partition is used for testing purposes.
- Four main programs:
  - OfficeVision
  - Applications that generate:
    - Pay checks every two weeks
    - Student schedules
    - Student grade sheets

  There are also a number of applications used by the university's professors that Lynne will have to consider during the conversion.
- Three major groups of users:
  - Professors
  - Administrative staff
  - Students.

A more detailed plan and schedule is provided in “Sample Plan and Conversion Schedule” on page 65.

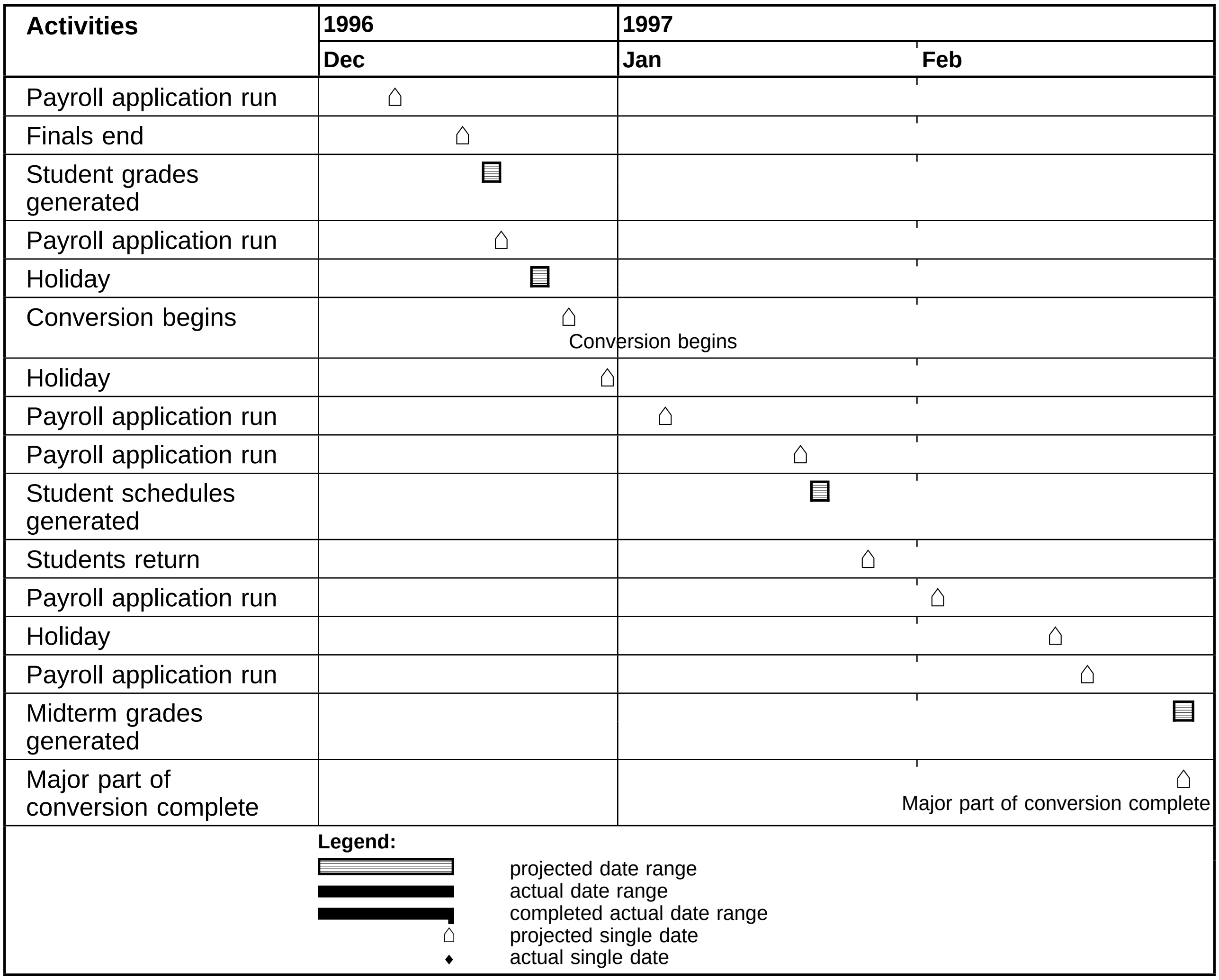


Figure 1. Lynne's Production Schedule

Activities	Projected		Actual	
	Start	End	Start	End
Payroll application run	12/09/96	12/09/96		
Finals end	12/16/96	12/16/96		
Student grades generated	12/18/96	12/19/96		
Payroll application run	12/20/96	12/20/96		
Holiday	12/23/96	12/24/96		
Conversion begins	12/27/96	12/27/96		
Holiday	12/31/96	12/31/96		
Payroll application run	01/06/97	01/06/97		
Payroll application run	01/20/97	01/20/97		
Student schedules generated	01/21/97	01/22/97		
Students return	01/27/97	01/27/97		
Payroll application run	02/03/97	02/03/97		
Holiday	02/14/97	02/14/97		

Activities	Projected		Actual	
	Start	**End**	**Start**	**End**
Payroll application run	02/17/97	02/17/97		
Midterm grades generated	02/25/97	02/26/97		
Major part of conversion complete	02/26/97	02/26/97		

Plan Your System Configuration Conversion Strategy

Before documenting your system configuration conversion strategy, read “System Configuration Conversion Strategies” on page 26 to help you determine which strategy to use to convert your operating system to the new system. You should also complete, as much as possible, the hardware and software inventory tables, so you know the resources you currently own and the resources you need during your system conversion. See Chapter 4, “Preparing Hardware and Software Inventory” on page 13.

Other conversion tasks you should consider at this time are:

- Hardware and software installation
- Hardware I/O
- System definition files
- User directory
- Spool files
- Connectivity
- Performance
- System configuration testing
- Backing out your system

What to Document in Your Plan for System Configuration Conversion Strategy

In your conversion plan you should provide details on the system configuration conversion strategy you plan to implement. This can include a combination of system configuration strategies documented in “System Configuration Conversion Strategies” on page 26 or may include a strategy not documented in this book. You should also document the time frame for implementing your strategy. See the following lists for details to include in your plan.

For the 'one processor with the new system running as a guest on the old system approach' or 'two processors, or one processor physically or logically partitioned approach', tasks to include in your plan include:

- Check that there is enough processor and storage capacity.
- Install and test the base VM/ESA 2.4.0 system, either in a processor, a partition, or as a guest.
- Reconfigure your VM/ESA 2.4.0 system if needed.
- Test the new system.
- Back up the old system's spool files, and restore them on the new system, if needed.
- Move users and applications to the new system. More details on this are discussed in “Plan Your Application, Program Product, and Data Conversion Strategy” on page 60 and “Plan Your User Conversion Strategy” on page 63.

- Archive your old system.
- Reconfigure with the new system first level, if you are using the guest strategy.

For the 'cut and go strategy' tasks to include in your schedule include:

- Check that there is enough processor and storage capacity.
- Archive your old system.
- Bring up the new system and test it.
- Save the old system's spool files and restore them on the new system, if needed.
- Move guests, users, and applications to the new system. More details on this are discussed in “Plan Your Application, Program Product, and Data Conversion Strategy” on page 60 and “Plan Your User Conversion Strategy” on page 63.

Hardware and Software Installation

Before you plan your software installation you should become familiar with the install and service process by reading:

- The chapter in this book that discusses installation changes and conversion considerations.
- The chapter in this book that discusses service changes and conversion considerations.
- *VM/ESA: Installation Guide*.
- *VM/ESA: VMSES/E Introduction and Reference*.
- *VM/ESA: Service Guide*.
- The Program Directory.

You should complete your installation with the CP system configuration files included with the new product. Once you have installed the new system, you can then modify these files. A customized user directory is created during installation, but of course you can also modify it after the installation is complete.

What to Document in Your Plan

- All hardware you plan to use during and after your conversion.
- All releases of VM you need to install during your conversion.
- New processor arrival and installation dates.
- New I/O devices arrival and installation dates.
- Arrival and installation dates for the new system.
- Who will install the hardware and software.

Hardware I/O

If you are converting from VM/ESA 1.1.5 370 Feature, hardware I/O architecture is very different in VM/ESA 2.4.0. Your previous hardware definition mechanism (for example, IOCP) may or may not work in VM/ESA 2.4.0. Before you plan your hardware, make sure you educate yourself on the new I/O architecture. You can start by reading:

- “Configuring I/O [1.1.5]” on page 217 for information on changes in I/O for VM/ESA 2.4.0. This section includes examples of converting your IOCP.
- *ES/9000 and ES/3090 Input/Output Configuration Program User's Guide* for information on hardware I/O.

If you are converting from VM/ESA 1.2.0 or later, the IOCP is basically the same as your old release, but you may still want to refer to the IOCP book.

What to Document in Your Plan

Your conversion plan should describe how your IOCP will be used during your conversion.

Conversion Strategy	Number of IOCPs Needed	Explanation
Second level or cut and go	1	• Converting from VM/ESA 1.1.5 370 Feature, you need a new IOCP for your new system. • Converting from VM/ESA 1.2.0 or later, you should be able to use your old IOCP.
LPAR	2	• An IOCP when using the LPAR during your conversion. • An IOCP for your new system once you cut over.
Two processor	3	• An IOCP for each side of the physically partitioned processor used during your conversion. • An IOCP for when you remove the partitioning.

Other tasks to include in your plan are:

- Who will code the new IOCPs, if needed.
- If new IOCPs are needed:
  - When you will convert them during the conversion.
  - When you will cut over entirely to the new IOCPs.

System Definition Files

If you are converting from VM/ESA 1.1.5 370 Feature, you need to consider converting to the new system generation files. There are four system generation files you need to convert: DMKSYS, DMKRIO, DMKBOX, and DMKSNT. A new system configuration file named SYSTEM CONFIG can be used in place of DMKSYS and DMKRIO. A file named LOGO CONFIG can be used in place of DMKBOX. CP commands replace DMKSNT (see “Converting DMKSNT” on page 217).

During installation, you can choose to exploit the new CP system configuration files immediately, or you can choose to convert your DMKSYS, DMKRIO, and DMKBOX files to HCPSYS, HCPRIO, and HCPBOX files. IBM recommends using the new CP system configuration files. Converting to HCPSYS, HCPRIO, and HCPBOX is not as easy as converting to the new CP system configuration file, SYSTEM CONFIG.

Note that you still need HCPRIO and HCPSYS files with the following entries:

System File	Minimum Entry
HCPSYS	SYSEND
HCPRIO	RIOGEN CONS=DYNAMIC

HCPSYS and HCPRIO files with these entries are shipped on the VM/ESA System DDR tapes. They are also included in the CP loadlist and should not be removed.

Note: HCPRIO entries are also needed for dedicated devices of V=R guests that use V=R recovery.

Before you document how you will convert your system generation files, you should read the following:

- “Converting DMKSYS and DMKRIO [S370]” on page 76
- “Converting to System Configuration Files from HCPRIO and HCPSYS [ESA]” on page 79
- *VM/ESA: Planning and Administration*

What to Document in Your Plan

- Who will convert the DMKSYS and DMKRIO files.
- When you will convert your existing CP definition files (DMKSYS and DMKRIO) to the CP system configuration file (SYSTEM CONFIG). A sample SYSTEM CONFIG file is provided with the new system.

User Directory

Before you document how you will convert your user directory you should read the following:

- “How to Convert Your User Directory [ALL]” on page 107
- “Considerations for Converting Your User Directory [ALL]” on page 211
- The user directory control statements compatibility table for your conversion in Part 3, “Compatibility Tables,” which identifies new control statements and control statement changes
- *VM/ESA: Planning and Administration* for details on the directory control statements

You should not plan to modify the user directory until you have finished installing your new system.

What to Document in Your Plan

- Whether you will use one directory for both systems or maintain two directories.
- If you maintain two directories,
  - What entries will be in both the old and new directories.
  - How you ensure changes to one directory are accurately reflected in the other directory.
- When you will create and update the directory for your new system.
- Who will be responsible for the directory changes.
- When you will merge the new and old directories.

Spool Files

Before you document your spool file movement during your conversion, read:

- "Spool File Conversion Considerations and Procedures [S370]" on page 88 or "Converting Spool Files and Saved Segments [ESA]" on page 91
- The CP commands compatibility table for your conversion to see if there are any changes to the SPTAPE command. See Part 3, "Compatibility Tables" on page 313.
- *VM/ESA: Planning and Administration* for information on allocating spooling space.

Note that VM/ESA Version 1 Release 1.5 370 Feature system data files cannot be moved to and cannot be used in VM/ESA Version 2 Release 4.0.

What to Document in Your Plan

- Amount of space allocated for SPOOL in the new system.
- When you plan to save spool files on tape.
- Whether spool files will be available to users during testing.
- How you will transfer or share spool files between the old system and the new system. See "Procedure for Converting Spool Files Using SPTAPE [S370]" on page 90 for information on moving spool files.
- If you plan to use a CSE complex as part of your conversion, what user IDs will be able to share spool files.

Connectivity

If you convert your users to the new system in a staged fashion, your production system may be split between your current system and your new system. One major concern is how you will share data between these systems.

Before you plan how you will use connectivity during your conversion, see:

- "Sharing Data between Your Old System and Your New System [ALL]" on page 96 and *VM/ESA: Planning and Administration* for information on sharing data using reserve/release
- "Sharing Data among CMS Users on Multiple Systems [ALL]" on page 106 and the *VM/ESA: Planning and Administration* for information on sharing data in a CSE complex.
- *VM/ESA: Connectivity Planning, Administration, and Operation* for information on VM connectivity options (for example, setting up a TSAF collection or setting up CS collection)
- "Establishing Connectivity between First-Level and Second-Level Systems [ALL]" on page 117

What to Document in Your Plan

- How you will communicate between systems (for example, using RSCS or TSAF).
- Where your data will reside.
- Time frame for when your systems will be connected and sharing data.

- The user IDs that need access to data on different systems and how you will connect the user IDs and data.
- Who will be responsible for connectivity during the conversion.
- The system ID specified in the system configuration file and the net ID specified in the system netid file.

Performance

Before you plan the performance for the new system, you should:

- Read “Managing System Performance [1.1.5]” on page 270 for information on managing your performance in the new system. This section includes information on the new system's scheduler and monitor, changes in tuning commands, and changes in paging and swapping.
- See “Advantages and Disadvantages of System Configuration Conversion Strategies” on page 28 for information on performance for each system configuration conversion strategy.
- Review *VM/ESA: Performance* for details on performance.
- Review the most current *VM/ESA Performance Report*.

You should also collect monitor and accounting data before you start your conversion and when you first cut over to the new system.

What to Document in Your Plan

- Your current performance level, the performance level you expect during your conversion, and the performance you expect once you cut over to the new system.
- When you will collect monitor and accounting data on the old system and on the new system.
- Who will collect the performance data.

System Configuration Testing

The testing plan you document depends on the type of system configuration conversion strategy you choose.

What to Document in Your Plan

- How you will test the new system the first time you install it. This can include:
  - Experimenting with the CP system configuration files
  - Trying frequently used commands and utilities
  - Trying to run critical applications or execs.
- How you will test the new system after setting up any partitions or second level systems.

Backing Out Your System

When planning your conversion, you must plan how you will backout if you run into a problem, such as an abend from which you cannot recover. In this case, you need to be able to quickly re-IPL your old system.

Before documenting this part of your plan, understand how the system configuration conversion strategy you choose affects backing out. For example, if you use the two processor (or one processor physically partitioned into two) or the LPAR approach, you always have an old system running. If the new system goes down, your old system will still be running.

For information on how to backout, see "How to Back Out [ALL]" on page 133.

What to Document in Your Plan

- The criteria for when to backout. And if you do backout, the criteria for when to move back to the new system.
- Determine if you want to make different copies or levels of CP available.
- Plan for maintaining a user directory from your old system.
- Backing up spool files.
- How your connectivity plans support backing out.
- Back out procedures for your operators.

Plan Your Application, Program Product, and Data Conversion Strategy

Before you plan your application, program product, and data conversion strategy, make sure you read:

- "System Application, Program Product, and System Data Conversion Strategies" on page 34.
- *Licensed Products Migration Matrix for VM/ESA or WHAT IS SUPPORTED WHERE for VM*. This document is available on the IBM VM operating system home page (http://www.ibm.com/s390/vm).

Other conversion tasks you should consider at this time are:

- Saved segment layout
- Guest operating systems
- Application and program product backout plans

What to Document in Your Plan

- When you should order any new licensed programs or vendor products.
- The applications and program products needed in the new system. See "Software Inventory" on page 16; you may have already collected this information.
- Which application conversion strategy or combination of strategies you plan to use and the time frame for when you will implement it. Checkpoints for your application strategy can include:
  - When a VM/ESA 2.4.0 CMS will be available and when applications and program products will be installed.
  - When a backlevel CMS will be installed (if necessary) and when applications and program products will be installed on it.
  - When an old level of VM will be available as a guest (if necessary) and when applications and program products will be installed.
  - When the conversion will be complete with everything running on the new system.

  - When you plan to remove backlevel CMS.
  - When you plan to remove the old level of VM that is a guest (if desired).
- How your application strategy fits in with your user strategy. For example, if you have users who depend on PROFS®, when you move these users to the new system you have to make sure they have access to OfficeVision.
- Plans for testing applications and program products on:
  - The new system. Note the ones that run in an XC virtual machine, XA virtual machine, and an XA virtual machine with 370ACCOM feature.
  - VM/ESA 2.4.0 in a backlevel CMS. Note the ones that run in that environment.

  Note any critical applications that require an old system. You may have already recorded some of this information in “Software Inventory” on page 16.
- Any necessary APARs you need to apply to the old CMS. For information on APARs that can be helpful during your conversion, see Appendix C, “APARs Needed During Conversion” on page 641.
- Who will be responsible for application, program product, and data conversions.

Saved Segment Layout

When converting from VM/ESA Version 1 Release 1.5 370 Feature, although the basic concept of saved segments is the same, the saved segment support in the new system differs from the saved segment support in System/370 VM operating system. DMKSNT (and SNT OVERRIDE) cannot be used in the new system, you must completely remap your installation's shared storage. You cannot simply convert your DMKSNT or SNT OVERRIDE files but you may need some of the segment information out of these files to build your saved segments in your new system.

Before you plan your saved segment layout you should:

- Converting from VM/ESA Version 1 Release 1.5 370 Feature:
  - Understand the new system's segment management.
  - Determine how big the average user virtual machine should be in the new system and consider the type of virtual machine where your users will run (for example: XA, XC)? Will the applications your users need be in a segment that is accessible to this virtual machine? Will other critical user applications in other segments be overlaid?
- All conversions:
  - Note any applications that can reside in the same segment.
  - Consider programs that interact. Make sure the segments where these programs reside do not overlay each other.
  - Understand how VMFSGMAP works in VMSES/E and how it can help you plan for and map saved segment layouts in VM/ESA. See “Planning Your System's Saved Segment Layout [S370]” on page 81 or “Converting Spool Files and Saved Segments [ESA]” on page 91. and *VM/ESA: VMSES/E Introduction and Reference* for more information on VMFSGMAP.

There are several sources where you can find information on segment management:

"Planning Your System's Saved Segment Layout [S370]" on page 81 or
"Converting Spool Files and Saved Segments [ESA]" on page 91
Helps you plan your segment layout.

"Saved Segments and Named Saved Systems Differences [1.1.5]" on page 254
Describes the differences in segment management between VM/ESA Version 1 Release 1.5 370 Feature and VM/ESA Version 2 Release 4.0 and helps you convert DMKSNT statements to DEFSEG commands.

VM/ESA: Planning and Administration
Helps you plan your saved segment layout based on virtual machine size and type of virtual machine (that is, 370, XA, or XC). This book also describes how to build, purge, and display information on saved segments as well as how to install applications into saved segments.

What to Document in Your Plan

- Applications and program products you plan to store in saved segments and where they will reside:

  1. In an XC virtual machine above the 16MB line.
  2. In an XA virtual machine above the 16MB line.
  3. In an XC virtual machine below the 16MB line.
  4. In an XA virtual machine below the 16MB line.
  5. In a 370 virtual machine.

  You may have collected some of this information in "Software Inventory" on page 16.

- All pre-requisites and co-requisites for each program.
- Users that are dependent on these programs.
- How you will group the programs into saved segments.
- When each segment will be installed on the new system.

Guest Operating Systems

Before you move your guest operating systems, see:

- *VM/ESA: Running Guest Operating Systems* for information on running VSE, MVS, and VM guests on the new system.
- Guest support matrix in the *VM/ESA: General Information* book for details on what releases of guests run on VM/ESA.

What to Document in Your Plan

- Determine what version and release of a guest you need for the new system.
- Decide when you should order any new operating systems.
- Decide when you will install these guests and who will install them.

Backing Out Your Application Programs

Before you develop a backout plan, make sure you understand the system configuration conversion strategy you chose and how that can support your backout.

For example, if you use the 'LPAR' strategy with your old system running on one half and the new system running on the other half, you can keep certain critical applications running on both systems. If you run into problems with your new system, you will still have access to your old system.

You may also consider dual-path code so your application can run in your old release, if necessary.

What to Document in Your Plan

- The criteria for when to backout because of applications. And if you do back out, the criteria for when to move back to the new system.
- Decide who is responsible for backing out.
- Document the applications critical to your installation. Make sure you can backout to an old release if you need to. ('Backout' is sometimes referred to as 'backoff'.)
- If your applications access data, decide how you will ensure data consistency.

Plan Your User Conversion Strategy

Before planning your user conversion strategy, read: “Strategies for Converting Your Users” on page 41 to determine what user conversion strategy you should use. Also, review the following compatibility tables in this book:

- CP compatibility tables
- CMS compatibility tables
- REXX compatibility tables

These tables include information on changes to commands, DIAGNOSE codes, macros, and REXX instructions that your end users and application programmers may notice.

Other conversion tasks you should consider at this time are:

- User education - public relations
- User backout.

What to Document in Your Plan

- Which user conversion strategy or combination of strategies you plan to use and the time frame for when you will implement it.
- Groups of users moving to the new system and when each will move.
- When you will let users test the new system.
- How your user strategy fits in with your application strategy. For example, if you have several clerks dependant on a store inventory application and you move the inventory application and data to a new system, you have to make sure you move the clerks over to the new system at the same time.
- If you will require users to convert their own applications and execs and how you will support them.

User Education - Public Relations

Before converting, communicate the conversion plan to the users of your current system. Inform your users of the:

- Justifications for the conversion
- Benefits of the new system
- Dates for the main tasks in your conversion plan
- System resources that will be unavailable at certain times
- Role your users play in the conversion process (that is, testing their individual applications and execs).

During the conversion, communicate the status of the conversion to the users of your new system.

Because much of the impact of the conversion is to applications and execs, you should involve application programmers in the conversion. They will have to check their applications and execs, recode them if changes are required, and test them. Involve end users, too, because the external interfaces of some system functions may change.

Some forms of communication are:

- Classroom education
- Status and update meetings
- Help phone service
- A bulletin board facility or conferencing system that could be implemented in several ways:
  - As read-only or help files that users can browse.
  - As a disconnected virtual machine to which users can submit questions and comments that are available for all to read and respond to. The questions and comments would reside in files, and users would link with read access to the minidisks containing these files. As users submit their questions or comments (updates), the disconnected virtual machine would append the updates to the appropriate file based on file name or file type.

What to Document in Your Plan

- Education provided for end users, application programmers, system programmers, and operators.
- How to notify your users of the system status.
- Who will provide the education.

User Backout

Before you plan for backing out your users, read “Backing Out Individual Users Because of Problems with Application Programs [ALL]” on page 137.

What to Document in Your Plan

- How you will maintain your directory so your users can easily backout to the old system.
- How you will maintain important applications so your users can get to old levels if the new version is not available.
- How you will transfer user spool files if you have to backout to your old system.

Sample Plan and Conversion Schedule

This section describes Lynne's conversion plan, first discussed in “Sample Conversion Schedule” on page 52, in more detail. Lynne will plan her conversion as follows:

1. System configuration conversion strategy
2. User conversion strategy
3. Application conversion strategy.

A majority of the conversion takes place December through February at her university. It is based on the calendar shown in Figure 1 on page 53.

Note: This is just a sample plan and schedule.

APARs to Apply to the Old System

Lynne needs to make sure that several APARs have been applied to her system before converting to VM/ESA 2.4.0. Lynne found the numbers of the APARs she needs in Appendix C, “APARs Needed During Conversion” on page 641.

System Configuration Information

The university's 9121 model 480 is already set up with an LPAR. A production VM/ESA Version 1 Release 2.0 system is running in one partition; the other partition is for testing purposes.

Lynne plans to use the 'one processor logically partitioned into two' approach for her system conversion strategy. The main reasons for choosing this strategy are:

- The LPAR is already set up, and Lynne has lots of experience with LPARs.
- The time needed to convert using the LPAR strategy will not take as long because there is an LPAR already set up.
- Lynne's installation is made up of mainly CMS users. She does not have any guests, so the LPAR should not affect performance to any great extent.
- The payroll application (discussed later) must be available at certain times during the month. The LPAR approach will let Lynne install and test the new system without affecting the reliability of the old system. Therefore, the reliability of this critical application can be maintained.

For a specific schedule of events, see “Detailed Conversion Schedule” on page 69.

Hardware and Software Installation

A new processor is not needed for this conversion because VM/ESA 2.4.0 can run on a 9121. Lynne only plans to install VM/ESA 2.4.0 in one partition and keep her old system running in the other partition.

Four additional 3380 DASD packs will be installed.

Hardware I/O

Only one IOCP is needed during Lynne's conversion because she plans to keep the LPAR configuration intact even after the conversion is complete. Lynne will convert her own IOCP and will add new DASD devices as necessary.

System Definition Files

The new CP configuration function will be exploited immediately. The sample SYSTEM CONFIG file provided on the VM/ESA System DDR tapes will be used during installation of the new system.

User Directory

Two user directories will be used throughout the conversion - one for the old system and one for the new system. A customized directory for the new system is created during installation of the new system.

For a period of time the directory entries for end users will have to be updated in both directories. Updates will be made to the directories as follows:

Note: This is just an example, and therefore the dates are not current.

Who	Dates	Description of Changes to the Directories
Professors	2/9/97 to 4/1/97	Maintain the directory entries for professors in both the VM/ESA 1.2.0 and VM/ESA 2.4.0 directories during this time. Professors will be required to convert their own applications. This gives them access to both systems while they are converting their applications until 4/1/97. After this, the old directory will be maintained only for backout purposes.
Administrative staff	2/9/97 to 4/1/97	Maintain the directory entries for administrative staff in both the VM/ESA 1.2.0 and VM/ESA 2.4.0 directories during this time. This is for backout purposes. After 4/1/97 only the VM/ESA 2.4.0 directory needs to be updated for the administrative staff.
Students	1/24/97 to 5/7/97	Only students have access to the VM/ESA 1.2.0 system throughout the semester. Updates need to be made for the students only in the VM/ESA 1.2.0 directory.

Spool Files

Lynne's planning indicates that her installation needs a whole DASD pack for VM/ESA 2.4.0 spool space.

For a short period of time the professors will be on the new system and the administrative staff will be on VM/ESA Version 1 Release 2.0. During that time, all OfficeVision notes and other reader files sent to the professors' VM/ESA Version 1 Release 2.0 system will automatically be sent to their new system user IDs.

Spool files will be dumped and restored as users move to the new system:

1. Professors will move first. Their spool files will be dumped and restored on the new system at that time.
2. Administrative staff will move next and their spool files will be moved when they go to the new system.
3. The students will remain in the VM/ESA Version 1 Release 2.0 system so their spool files do not need to be moved.

Connectivity

CSE will be used to set up shared spool files between the old and new systems during the conversion. This means that Lynne needs to upgrade RSCS to set up the connections.

The CSE complex will also allow minidisks to be shared between the VM/ESA Version 1 Release 2.0 and VM/ESA 2.4.0 systems. The data used for:

- Professor's research applications
- Payroll and student grade reporting

will reside on these shared minidisks. Then, if a backout is necessary, the applications running on VM/ESA Version 1 Release 2.0 can be used and have access to the most current data.

System Testing the New System

The first test of the new system includes experimenting with the CP system configuration files, trying frequently-used commands and utilities, and running their applications to determine the conversion effort needed.

Backout

Lynne plans to take certain measures during the conversion to be prepared in case a backout is needed. She will:

- Maintain a backup copy of the system residence volume even after cutover
- Maintain two user directories, one for the old system and one for the new
- For every major cutover to the new system, back up spool files using SPTAPE
- Use shared minidisks in a CSE complex
- Let professors and administrative staff keep both old and new system user IDs for a certain period of time.

User Conversion Information

There are two major user groups that Lynne has to consider during her conversion: professors and administrative staff. Students also use this system but will stay on VM/ESA Version 1 Release 2.0 throughout the semester. Students will use the new system in the fall.

Education sessions will be held for professors and administrative staff before they move to the new system. A phone hot line will be set up during the conversion in case either user group has questions or problems with the new system.

Professors will be moved to the new system first. They will be asked to convert their own applications and will have time to test their applications on the new system. Professors will be allowed to keep their VM/ESA Version 1 Release 2.0 user IDs for the duration of the conversion for the following reasons:

- So they have time to convert their own programs, and
- In case a backout is necessary.

Their data will be stored on shared DASD through the CSE complex so if they have to backout they still have access to the most current data from the VM/ESA Version 1 Release 2.0 system.

The administrative staff will also have user IDs on both systems from the time they move to the new system until the end of the semester. Although they will primarily use their new user IDs after a certain time, maintaining the old user IDs will allow for a backout, if necessary. In addition, the data used by the payroll and student grading applications will be stored on shared DASD through the CSE complex so if a backout is necessary, the VM/ESA Version 1 Release 2.0 system will have access to the most current data.

Application, Program Product and Data Conversion Information

Lynne plans to use a combination of strategies for converting her applications, program products, and data: 'convert on the new system using the new CMS' and 'convert on the new system using backlevel CMS'. The following table lists the applications Lynne has to consider during the conversion and the conversion strategy she plans to use for each program.

Application	Users	Frequency Used	Strategy for Converting
OfficeVision	Administrative staff, professors	Daily	New system with new CMS
Payroll application	Administrative staff	Run every other week	New system with new CMS
Student grade reporting	Administrative staff	Run twice per semester, once at midterm and once at the end of the semester	New system with new CMS
Student scheduling	Administrative staff	Run before the start of the semester	New system with new CMS
Research applications	Professors	Run throughout the semester	New system with new CMS or backlevel CMS

Lynne wants to get all applications used by the administrative staff running in the same type of virtual machine on the new CMS. OfficeVision is especially critical for the administrative staff, and OfficeVision requires that the VM components, CP and CMS, run at the same release level. Therefore, Lynne cannot use a backlevel CMS environment to run OfficeVision.

The data accessed by the payroll and student grading application will reside on shared DASD and will be available to both the VM/ESA Version 1 Release 2.0 and the new systems through the CSE complex. This way both systems can have access to the most up-to-date data.

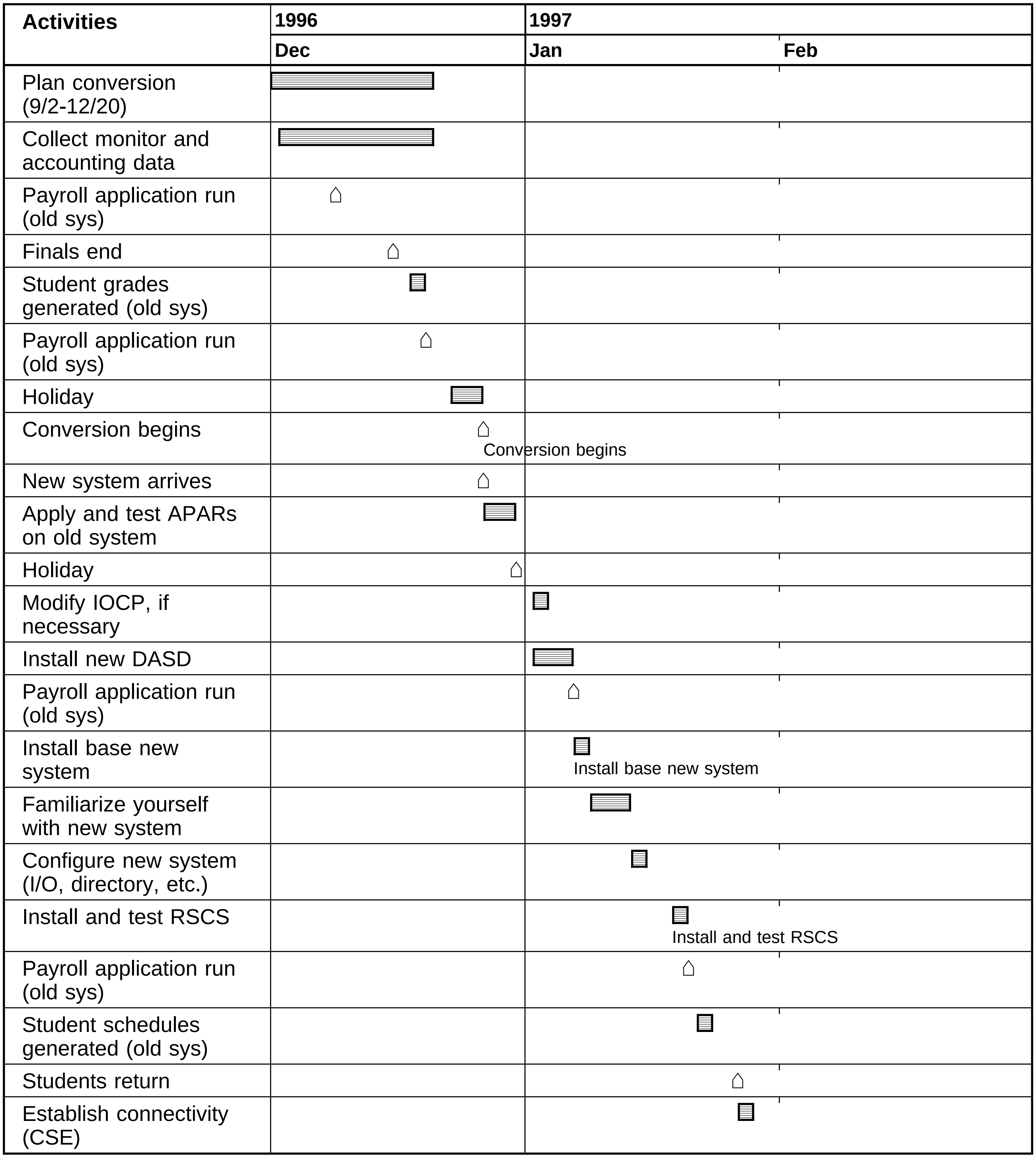
The payroll, student grade reporting, and student scheduling applications will be converted as necessary by application programmers at the university.

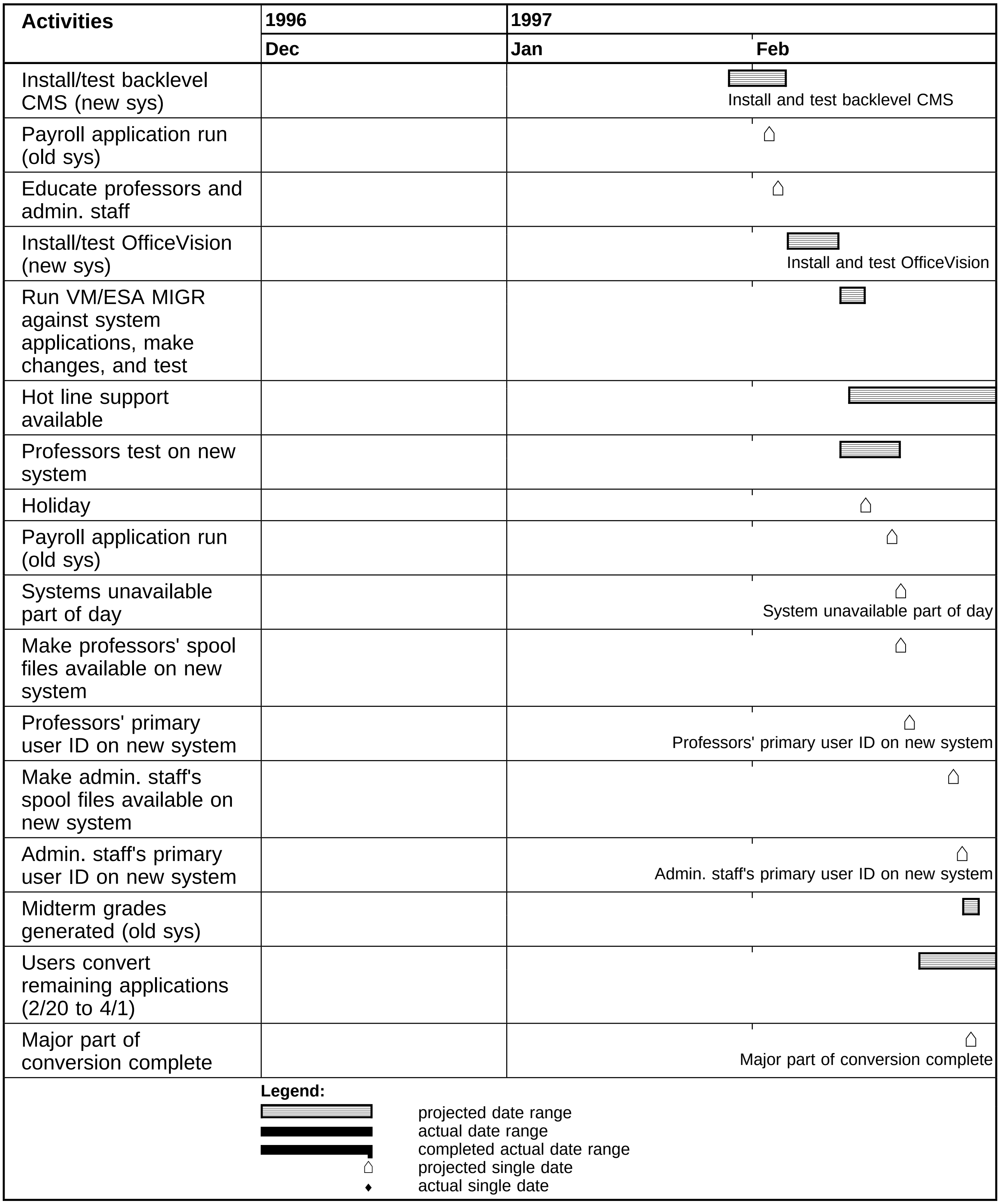
The professors will be allowed to test their own applications on a VM/ESA 2.4.0 system. The professors will be the first user group to move and will be asked to convert their own applications. A backlevel CMS will be available for a period of time after the conversion is finished in case the professors cannot convert their own applications right away.

Detailed Conversion Schedule

Following is a schedule of events for Lynne's conversion.

Note: This is just an example, and therefore the dates are not current.





Activities	1996 Dec	1997 Jan	 Feb
Install/test backlevel CMS (new sys)		▤	▤ Install and test backlevel CMS
Payroll application run (old sys)			⌂
Educate professors and admin. staff			⌂
Install/test OfficeVision (new sys)			▤ Install and test OfficeVision
Run VM/ESA MIGR against system applications, make changes, and test			▤
Hot line support available			▤
Professors test on new system			▤
Holiday			⌂
Payroll application run (old sys)			⌂
Systems unavailable part of day			⌂ System unavailable part of day
Make professors' spool files available on new system			⌂
Professors' primary user ID on new system			⌂ Professors' primary user ID on new system
Make admin. staff's spool files available on new system			⌂
Admin. staff's primary user ID on new system			⌂ Admin. staff's primary user ID on new system
Midterm grades generated (old sys)			▤
Users convert remaining applications (2/20 to 4/1)			▤
Major part of conversion complete			⌂ Major part of conversion complete

Legend:

▤	projected date range
▬	actual date range
▬	completed actual date range
⌂	projected single date
♦	actual single date

Activities	Projected		Actual	
	Start	End	Start	End
Plan conversion (9/2-12/20)	09/02/96	12/20/96		
Collect monitor and accounting data	12/02/96	12/20/96		

Activities	Projected		Actual	
	Start	End	Start	End
Payroll application run (old sys)	12/09/96	12/09/96		
Finals end	12/16/96	12/16/96		
Student grades generated (old sys)	12/18/96	12/19/96		
Payroll application run (old sys)	12/20/96	12/20/96		
Holiday	12/23/96	12/26/96		
Conversion begins	12/27/96	12/27/96		
New system arrives	12/27/96	12/27/96		
Apply and test APARs on old system	12/27/96	12/30/96		
Holiday	12/31/96	12/31/96		
Modify IOCP, if necessary	01/02/97	01/03/97		
Install new DASD	01/02/97	01/06/97		
Payroll application run (old sys)	01/07/97	01/07/97		
Install base new system	01/07/97	01/08/97		
Familiarize yourself with new system	01/09/97	01/13/97		
Configure new system (I/O, directory, etc.)	01/14/97	01/15/97		
Install and test RSCS	01/19/97	01/20/97		
Payroll application run (old sys)	01/21/97	01/21/97		
Student schedules generated (old sys)	01/22/97	01/23/97		
Students return	01/27/97	01/27/97		
Establish connectivity (CSE)	01/27/97	01/28/97		
Install/test backlevel CMS (new sys)	01/29/97	02/04/97		
Payroll application run (old sys)	02/03/97	02/03/97		
Educate professors and admin. staff	02/04/97	02/04/97		
Install/test OfficeVision (new sys)	02/05/97	02/10/97		
Run VM/ESA MIGR against system applications, make changes, and test	02/11/97	02/13/97		
Hot line support available	02/12/97	02/28/97		
Professors test on new system	02/11/97	02/17/97		
Holiday	02/14/97	02/14/97		
Payroll application run (old sys)	02/17/97	02/17/97		
Systems unavailable part of day	02/18/97	02/18/97		
Make professors' spool files available on new system	02/18/97	02/18/97		
Professors' primary user ID on new system	02/19/97	02/19/97		
Make admin. staff's spool files available on new system	02/24/97	02/24/97		
Admin. staff's primary user ID on new system	02/25/97	02/25/97		
Midterm grades generated (old sys)	02/25/97	02/26/97		
Users convert remaining applications (2/20 to 4/1)	02/20/97	02/28/97		
Major part of conversion complete	02/26/97	02/26/97		

Part 2. Conversion Tasks and Considerations

Chapter 7. System Conversion Tasks

This chapter contains topics such as how to convert your IOCP, converting spool files and saved segments, how to back out, installing a backlevel CMS, as well as other topics.

This chapter may contain information for any or all of the conversions documented in this book. For definitions of the conversion identifiers used in this chapter, see “How to Identify the Topics That Apply to Your Conversion” on page 3.

This chapter contains the following major sections:

How to Convert Your IOCP [ALL]

S370

For information on differences in I/O between System/370 and ESA/370 or ESA/390, see “Configuring I/O [1.1.5]” on page 217. That section also discusses changes in the IOCP.

ESA

For an ESA conversion, your current IOCP should work for you on the new system. However, if you need more information, consult the IOCP book that applies to your processor. That book contains information about setting up an IOCP for LPAR.

Converting DMKSYS and DMKRIO [S370]

You cannot use DMKSYS and DMKRIO files in the new system. IBM strongly recommends that you convert your DMKSYS and DMKRIO files to a CP system configuration file. The CP system configuration file, named SYSTEM CONFIG, is new.

The other method of defining your CP configuration is with HCPSYS and HCPRIO files. It is possible to convert your DMKSYS and DMKRIO files to HCPSYS and HCPRIO files but this makes your conversion more difficult. Because system administration is easier with the new CP system configuration support, you will probably want to convert to SYSTEM CONFIG eventually anyway - so it is best to convert to SYSTEM CONFIG now.

Before VM/ESA can run, it must have a system residence volume, one CP-owned DASD, and one operator console. On your current system, these items are defined by macros coded in the DMKSYS ASSEMBLE and the DMKRIO ASSEMBLE files. The SYSRES macro in DMKSYS defines the system residence volume while the SYSOWN macro defines the CP-owned DASDs. The operator console is defined in the DMKRIO ASSEMBLE file by the RIOGEN macro.

The SYSTEM CONFIG file replaces the DMKSYS and DMKRIO files, for the most part. You still need HCPRIO and HCPSYS files with the following entries:

System File	Minimum Entry
HCPSYS	SYSEND
HCPRIO	RIOGEN CONS=DYNAMIC

HCPSYS and HCPRIO files with these entries are shipped on the VM/ESA System DDR tapes. They are also included in the CP loadlist and should not be removed.

You can add statements to the SYSTEM CONFIG file to define your CP configuration instead of coding macros in HCPSYS or HCPRIO. The SYSTEM CONFIG file is located on the PARM disk. The PARM disk is defined during installation. During IPL, CP reads the SYSTEM CONFIG file and dynamically configures the system according to the statements in the file. The statements needed are:

SYSTEM_RESIDENCE	(replaces SYSRES)
CP_OWNED	(replaces SYSOWN)
OPERATOR_CONSOLES	(replaces RIOGEN)

The sample SYSTEM CONFIG file shipped with VM includes these statements, but you will need to change SYSTEM CONFIG to describe your system. In addition, you need to add SYSTEM CONFIG statements that list your user volumes (those that hold user minidisks). You will need to refer to:

- The documentation for DMKSYS and DMKRIO, which is in *VM/ESA CP Planning and Administration for 370*.
- The documentation for the system configuration file statements, which is in *VM/ESA: Planning and Administration*.
- Table 37 on page 348 and Table 38 on page 349, which show the relationships between the old macros and the new statements.

The following steps summarize how to convert your DMKSYS and DMKRIO macros to SYSTEM CONFIG statements:

1. Convert the SYSRES macro.

   For example, the SYSRES macro in DMKSYS ASSEMBLE might be coded as follows:

   ```
         SYSRES  SYSVOL=VMSRES,                                     X
                 SYSRES=123,                                        X
                 SYSTYPE=3380,                                      X
                 SYSCLR=YES,                                        X
                 SYSNUC=(1,5),                                      X
                 SYSWRM=(17,2,VMSRES),                              X
                 SYSERR=(19,2,VMSRES),                              X
                 SYSCKP=(346,1,VMSRES)
   ```

   An equivalent SYSTEM_RESIDENCE statement is:

   ```
   system_residence,
      checkpoint volid VMSRES from 346 for 1,
      warmstart  volid VMSRES from 17  for 2
   ```

   The SYSTEM_RESIDENCE statement defines the checkpoint and warm start areas, but it does not define any of the other characteristics described in the SYSRES macro. These definitions are not needed in the new system.

2. Convert the SYSOWN macro.

   In the following example, the SYSOWN macro in DMKSYS ASSEMBLE defines two CP-owned volumes, VMSRES and VMPK01:

   ```
         SYSOWN  VMSRES, VMPK01
   ```

   To define those volumes in the SYSTEM CONFIG file, use two CP_OWNED statements, as shown below. The SLOT operand tells CP the number of the slot in the CP-owned volume list. The optional OWN operand tells CP that only this system can create or destroy spool files on this volume. OWN is the default.

   ```
   cp_owned slot 001 vmsres own
   cp_owned slot 002 vmpk01 own
   ```

   If you have other CP-owned volumes, add them also. The following statements define three more CP-owned volumes, VMCP23, VMCP24, and VMCP25:

   ```
   cp_owned slot 003 vmcp23 own
   cp_owned slot 004 vmcp24 dump
   cp_owned slot 005 vmcp25 shared
   ```

   The slot positions are important because they are used by CP in determining the placement of spool files. Once the slot numbers are assigned, they should not be reordered.

   The DUMP operand indicates that the spool space on VMCP24 is reserved exclusively for dumps. SHARED means that volume VMCP25 is owned by another system in the cross system spooling complex. Systems in the cross system spooling complex can read spool files from VMCP25 but only the system that owns VMCP25 can write to it.

3. Convert the RIOGEN macro.

   The following RIOGEN macro defines a primary system console at device address 01F with an alternate console at address 050:

   ```
          RIOGEN CONS=01F,ALTCONS=050
   ```

   To define these consoles in the SYSTEM CONFIG file, use the OPERATOR_CONSOLES statement:

   ```
   operator_consoles 01F 050
   ```

   When the system is initializing, CP goes sequentially through the list of device addresses, using the first operational console as the system operator console.

4. Add control statements for user volumes.

   The USER_VOLUME_LIST statement defines a list of user DASD volumes. (There is no equivalent to this statement in DMKSYS or in DMKRIO.) In the following example, VMUS01, MPK002, and TKLS03 are defined as user volumes:

   ```
   user_volume_list VMUS01 MPK002 TKLS03
   ```

   You can specify as many USER_VOLUME_LIST statements as you need. If you have a set of volumes that have similar volume identifiers, consider using the USER_VOLUME_INCLUDE statement. This statement defines user volumes by a generic volume identifier. For example, if you have three volumes with labels, USE001, USE002, and USE003, you can define them all as user volumes using the following statement:

   ```
   user_volume_include use*
   ```

   The USER_VOLUME_INCLUDE statement can be specified as many times as needed. It can be used along with USER_VOLUME_LIST statements.

5. Convert real device definitions if necessary.

   When initializing, CP determines what real devices are attached to your system. In some cases, CP can identify all the devices, but in other cases it cannot. Some older devices cannot reply to CP's request for information, while other devices cannot supply all the information needed (for example, the spooling class and form number of a printer or cross-system sharing information for DASDs).

   For the devices that CP senses, it is not necessary to convert the real device definitions in the DMKRIO file. The definitions are not needed. For devices that CP cannot sense or for devices that send incomplete information, however, some conversion is necessary. Conversion involves writing RDEVICE statements in the SYSTEM CONFIG file that express the information contained in the various DMKRIO macros.

   To determine which devices need RDEVICE statements, refer to the Table 39 on page 349. See *VM/ESA: Planning and Administration* for a list of devices supported by VM/ESA 2.4.0.

   Some reconfiguration will also be necessary if you have remote clusters configured on your system on the CLUSTER and TERMINAL macros. Because the new system does not support binary synchronous communication lines, you must set them up through the VM/Pass-Through Facility (PVM). To do this, code the RDEVICE control statement for the binary synchronous line address and then attach it to PVM. Define the terminals to PVM in PVM's configuration file.

Finally, if you want to use the V=R guest recovery feature on a CP bounce or re-IPL and you have ANY devices defined in HCPRIO ASSEMBLE, then you MUST define all devices that will be dedicated to the V=R guest in HCPRIO ASSEMBLE. In this case, all dynamically defined devices dedicated to the V=R guest will not be preserved. However, if no devices are defined in HCPRIO ASSEMBLE, then the V=R guest recovery feature will preserve all the dynamically defined devices dedicated to the V=R guest over a CP bounce or re-IPL.

Converting to System Configuration Files from HCPRIO and HCPSYS [ESA]

If the VM/ESA system you are converting from already uses system configuration files, you have already made many of the decisions described in this section. However, if your system does not currently use system configuration files, you can use this information to help you convert to them now.

During installation, you can choose to exploit the new CP system configuration files immediately, or you can choose to use the old HCPRIO, HCPSYS, and HCPBOX files instead. IBM recommends using CP system configuration files.

If you choose to convert to the new system configuration files, you need to create a SYSTEM CONFIG file that represents your system (as currently defined in HCPSYS and HCPRIO). To create the SYSTEM CONFIG file statements, run the sample programs HCPTRIO and HCPTSYS. 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. (Note that HCPTSYS and HCPTRIO do not process DMKRIO and DMKSYS files.)

Note: 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 *VM/ESA: 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.

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

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

```
hcptsys 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, 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 *VM/ESA: Planning and Administration*. See Appendix E, “Sample Utilities for CP Configurability” on page 647 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 (only true for VM/ESA 1.2.0 or later) 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 *VM/ESA: Planning and Administration* for a complete description of HCPDCON.

Difference with Multiple Systems Using Shared Data

When using the new 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 [ESA]

If you plan to use the new SYSTEM CONFIG file, you still need an HCPSYS with a SYSEND macro.

If you choose not to convert to the new system configuration files, you may need to make changes to the HCPSYS macros. The HCPSYS compatibility tables in this book show differences between the old releases and the new release. See *VM/ESA: Planning and Administration* for complete descriptions of HCPSYS macros.

Converting HCPRIO Macros [ESA]

If you plan to use the new SYSTEM CONFIG file, you still need an HCPRIO with an RIOGEN CONS=DYNAMIC macro.

If you choose not to convert to the new system configuration files, you may need to make changes to the HCPRIO macros. The RDEVICE macro of HCPRIO has changed. See the HCPRIO compatibility tables in this book for details.

See *VM/ESA: Planning and Administration* for complete descriptions of HCPRIO macros. Note that some devices are no longer supported. See the *VM/ESA: General Information* book for information on device support.

Migration Tool [ALL]

The REXX/EXEC Migration Tool for VM/ESA (VM/ESA MIGR) can help you convert your DMKRIO/DMKSYS files to HCPRIO/HCPSYS, or upgrade your existing HCPRIO/HCPSYS files to the latest level. Note, however, that IBM recommends converting to using CP system configuration files instead.

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 files due to incompatibilities or changes in macros. VM/ESA MIGR can also present you with information about these changes through Help panels.

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

Planning Your System's Saved Segment Layout [S370]

Although the concept is the same, saved segment support in the new system differs from saved segment support in System/370. Some of the key differences are:

- You can define and build saved segments dynamically using commands.
- A segment space and a DCSS in VM/ESA is 1MB or a multiple of 1MB. Physical saved segments must be defined as member saved segments of segment spaces or as DCSSs. Also, consider packing your old release's DCSSs into member saved segments of segment spaces. See “Saved Segments and Named Saved Systems Differences [1.1.5]” on page 254 for information about the differences in saved segment support.
- The CMS nucleus resides from 15MB through 20MB, leaving you with storage up to 15MB to define saved segments for virtual machines in your VM/ESA system.

- VMFSGMAP EXEC in VMSES/E can be used to plan for and map out saved segment layouts in VM/ESA. Once saved segments are completely defined with VMFSGMAP, you can use VMFBLD in VMSES/E to build saved segments.

Because of the differences between saved segments and because of other factors—for example, the size of your applications may be larger—you must completely remap your installation's shared storage. Refer to “Saved Segments and Named Saved Systems Differences [1.1.5]” on page 254 for more thorough information about how saved segments are different, conversion inhibitors, and back out considerations.

For a complete description of saved segments in VM/ESA see the *VM/ESA: Planning and Administration* book, which contains the following information:

- Planning for saved segments based on your virtual machine size
- Planning for saved segments based on your type of virtual machine
- How to create saved segments and display information about them
- How to install applications in saved segments
- How to use the VMFSGMAP EXEC to plan the layout of and define your saved segments
- How to use VMFBLD to build segments that you defined using VMFSGMAP.

Steps to Follow to Plan Your Saved Segment Layout [S370]

In general, to plan for and map out your new system's saved segment layout, you need:

- To gather information about the saved segments you need in your system
- Then use VMFSGMAP, a segment mapping and planning tool, to help you map your system's saved segments.

To plan for and set up your saved segment layout in VM/ESA, you can follow these general steps:

1. Identify all of the products or applications in your VM/ESA system that will use saved segments.

   Refer to Chapter 4, “Preparing Hardware and Software Inventory” on page 13 where you were instructed to identify all of the products and applications that you will use on your new VM/ESA system and to identify those that will use saved segments.

2. Collect all of the saved segment definitions (that is, the default DEFSEG commands) for each of the products or applications that will use saved segments in your VM/ESA 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 the example below. “Table for Saved Segment Information” on page 638 contains an empty table like the example that you can use to gather this information.

Table 9. Example of Saved Segment Information in a System/370 Conversion

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 convert to VM/ESA, you may gather this information in two ways:

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

Although it may require more time planning for your conversion, you should consider mapping your entire saved segment layout at once. Otherwise, as you gradually install additional products, you may have to re-map and re-build 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 VM/ESA system.

a. Whether the saved segment is a CMS logical segment or a CP physical segment
b. Whether the saved segment can reside above the 16MB line
c. What execs or commands build and save the segment.

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

- Depending on how your VM/ESA 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 function. 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 either. 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 VMSES/E's VMFBLD exec. You input this information to VMSES/E using the VMFSGMAP interface.

3. For all of those products or applications that do not require a new version or upgrade to run on VM/ESA, you have to convert DMKSNT ASSEMBLE macros to DEFSEG commands or, if applicable, DEFSYS commands.

   See “Converting DMKSNT ASSEMBLE Macros or SNT OVERRIDE Tags to VM/ESA Commands [S370]” on page 84 for detailed steps on how to convert these statements.

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

   VMFSGMAP is a VMSES/E exec that provides a saved segment planning and mapping interface. 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 *VM/ESA: Planning and Administration* book.

   When manipulating your system's saved segment layout:

   - Remember that CMS requires 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 cannot be run above the 16MB line. Also, make sure that all segments that you need available for user virtual machines are mapped below the 15MB line.
   - Be careful not to overlay saved segments that require one another.

5. 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 input 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 *VM/ESA: Planning and Administration* book.

Converting DMKSNT ASSEMBLE Macros or SNT OVERRIDE Tags to VM/ESA Commands [S370]

If you will be moving products directly from your old production system to VM/ESA, you can either:

- Convert the saved segment definitions for those products to VM/ESA commands

- Or, if the product's installation documentation addresses installation on VM/ESA, you can use the default saved segment definitions provided. If this information is available to you, use it rather than following the steps described below for converting your existing saved segment definitions.

For any product or application that you plan to move directly to VM/ESA, you need to convert the saved segment definitions in DMKSNT ASSEMBLE or SNT OVERRIDE to DEFSEG or DEFSYS commands.

This conversion or remapping is a one-time effort.

The steps below specifically illustrate converting an entry in DMKSNT ASSEMBLE to a DEFSEG command. If the system you are converting from uses SNT OVERRIDE, converting these tags is similar. To see how entries in DMKSNT ASSEMBLE map to those in SNT OVERRIDE, see the following table:

DMKSNT ASSEMBLE macros	**SNT OVERRIDE tags**
NAMESYS	DEFSYS DEFSEG
NAMENCP	DEFNCP
NAME3800	DEF3800
NAMELANG	DEFLANG

Suppose that you have the following entry in DMKSNT ASSEMBLE:

```
NAMESYS SYSNAME=SFMASS00

SYSPGCT=144,
SYSPGNM=(2128-2271),
SYSHRSG=(133,134,135,136,137,138,139,140,141),
SYSVOL=volid,
SYSSTRT=(40,07),
SYSSIZE=576K,
VSYSRES=IGNORE,
VSYSADR=IGNORE
```

Figure 2. DMKSNT Sample Entry

To translate the information in this entry to definitions for the DEFSEG command, follow the steps below. It shows you how to obtain the equivalent DEFSEG command for the above DMKSNT entry.

The equivalent command for the preceding entry is:

```
DEFSEG  SFMASS00 800-88F SR
```

1. Extract the name on the SYSNAME parameter to use as *dcssname* on the DEFSEG command. This is the name given to the segment. In the DMKSNT entry, this is entered as:

   ```
   SYSNAME=SFMASS00
   ```

In the new system, this same segment name is entered as:

```
DEFSEG  SFMASS00
```

2. Convert the decimal value of SYSPGCT to hexadecimal. It is recommended that you use a hexadecimal calculator or conversion table to make the hexadecimal conversions.

   The result shows how many pages the program needs (in hexadecimal).

   In the example, the result would be X'90', meaning that the program requires X'90' pages of storage. If all pages were in the same segment, the page range would be X'0' to X'8F' within the chosen segment. Be sure to count page 0 as the first page. In the DMKSNT entry, this is entered as:

   ```
   SYSPGCT=144
   ```

   In the new system, X'90' pages are required. The page range may be X'00' to X'8F' within the chosen segment.

3. Divide the decimal value of SYSPGCT by 16.

   The result tells you how many 64KB segments the program needs.

   In the example, the result would be 9, meaning that the program requires nine 64KB segments. In the DMKSNT entry:

   ```
   SYSPGCT=144
   ```

   To convert this for the new system:

   ```
   144 divided by 16 equals 9
   ```

   The program needs nine 64KB segments. This information is used later (in step 5) in the DEFSEG command definition.

4. Convert both decimal values of the SYSPGNM parameter to hexadecimal values. This shows where the segment is located in storage.

   Use the hexadecimal values to define the starting and ending pages and relative location of the segment.

   In the example shown, 2128 is X'850'. The likely place to locate this program might be to start it on page X'0' in segment 8. (If you use segment 8 for CMSVSAM, this new segment will overlay CMSVSAM.) Round *down* to the nearest segment boundary because saved segments in the new system must start and end on a megabyte boundary. The ending page is X'8DF' (X'850' plus X'8F' pages). Count page 0 as the first page. In the DMKSNT entry, this is entered as:

   ```
   SYSPGNM=(2128-2271)
   ```

   Converting the values for the new system:

   ```
   2128 = X'850'     2271 = X'8DF'
   ```

   In the new system, this is entered as:

   ```
   DEFSEG  SFMASS00 800
   ```

   (You will use the ending page value later in the DEFSEG command definition.)

5. Look for the SYSHRSG parameter. If present, all the 64KB segments listed are to be shared by users. The first letter of the *mode* operand is S. Count the number of segments listed singly or within a segment range. Compare the count to the result of SYSPGCT divided by 16 (as calculated in step 3).

If the count equals the result of SYSPGCT, all storage is shared. When the count is less than the result of SYSPGCT, you must figure out which pages and how many pages of storage must be put in an exclusive segment or segments. The first letter of the *mode* operand is an E.

Another way to determine the first letter of the *mode* operand is to multiply the beginning segment number by 16 (the number of pages in a 64KB segment) and then multiply the ending segment number by 16 and add 15 (pages 0-15 of the last segment). Compare the two results to the decimal values for SYSPGNM:

- When your results are the same as the values in SYSPGNM, all segments are shared.
- When your results do not match the values in SYSPGNM, the segments outside your results are exclusive.

In the example, there are 9 segments listed as shared; 9 is the result of dividing SYSPGCT by 16 (as calculated in step 3 on page 86). So all segments defined for SFMASS00 will be shared. In the DMKSNT entry, this is entered as:

```
SYSHRSG=(133,134,135,136,137,138,139,140,141)
```

To convert this for the new system:

```
9 segments
133 x 16 = 2128
141 x 16 = 2256 + 15 = 2271
```

In the new system, the equivalent statement is:

```
DEFSEG  SFMASS00 800-88F Sm
```

The *m* in the above example is the second letter of the mode operand (either R, W, or N). This is discussed below.

If the example had `SYSHRSG=(133,134,135,136,137,138)`, 6 segments would be shared; the remaining 3 segments would be exclusive and would have to be placed in the next 1MB segment. Each segment of SYSHRSG represents `X'0'` to `X'F'` pages, so the shared portion would be the first `X'60'` pages (pages `X'0'` to `X'5F'`) of the lower segment, and `X'30'` pages would be the first pages in the next segment (pages `X'0'` to `X'2F'`). The page range might be `X'800'` to `X'85F'` for shared code and `X'900'` to `X'92F'` for the exclusive code. In the DMKSNT entry, this is entered as:

```
SYSHRSG=(133,134,135,136,137,138)
```

To convert this for the new system:

```
6 segments
133 x 16 = 2128
138 x 16 = 2208 + 15 = 2223
```

In the new sytem, the equivalent statement is:

```
DEFSEG  SFMASS00 800-85F Sm 900-92F Em
```

6. Look for the PROTECT parameter. When PROTECT=ON or when the parameter is missing (the default is ON), the second letter of the *mode* operand is R (for read-only). When PROTECT=OFF, the second letter of the *mode* operand is W (for read/write access). All exclusive segments should have W as the second letter.

In Figure 2 on page 85 the default PROTECT=ON is used. For the new system, this means that all pages are shared read-only. In the new system, the equivalent statement is:

```
DEFSEG  SFMASS00 800-88F SR
```

In Figure 2 on page 85 with both shared and exclusive segments, you would assign R to the shared segment because there was no PROTECT parameter, but W to the exclusive segment. In Figure 2 on page 85 the default PROTECT=ON is used. For the new system, shared pages are read-only and exclusive pages are read/write. In the new system, the equivalent statement is:

```
DEFSEG  SFMASS00 800-85F SR 900-92F EW
```

7. Look for the RCVID operand. If present, add the RSTD operand to the DEFSEG command. When you want to restrict access to the program, you might want to add the RSTD operand. You would also want to add the RSTD operand when a person loads the segment using DIAGNOSE code X'64' with the LOADNSHR function code. Access to a restricted segment requires that the user have directory authorization through the NAMESAVE directory statement. If you want to restrict access to the program in the example, you would add RSTD to the DEFSEG command. Suppose the following were entered in the DMKSNT entry:

   ```
   RCVID=userid
   ```

   In the new system, the equivalent statement would be:

   ```
   DEFSEG  SFMASS00 800-88F SR RSTD
   ```

8. The following parameters of the DMKSNT entry are not applicable to a DEFSEG command in the new system: SYSVOL, SYSSTRT, SYSSIZE, VSYSRES, VSYSADR, SYSCYL, SYSBLOK, USERID, and SAVESEG.

Spool File Conversion Considerations and Procedures [S370]

This section discusses how to avoid the loss of data, spool file limit considerations, and procedures for moving spool files.

Note: The procedures use the SPTAPE command. Do not use the SPXTAPE command that is available on your new system. SPTAPE and SPXTAPE are not compatible. SPXTAPE cannot load or scan files dumped by SPTAPE.

Also, refer to the CP Command compatibility table for your old release to review possible compatibility problems between the old and new versions of SPXTAPE or SPTAPE.

Avoiding the Loss of Spool Files during Conversion [S370]

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

- Back up your spool files using SPTAPE DUMP before reallocating SPOOL space
- Reallocate the SPOOL space

- Shutdown your old system; **do NOT use SHUTDOWN REIPL**
- IPL the new system with a cold start
- Restore the backed up spool files using SPTAPE LOAD.

Note: In your old system, spool space was allocated as part of your TEMP space. In the new system, it is allocated as part of your SPOOL space.

You Cannot Convert Your System Data Files from Your Old System [S370]

You cannot convert system data files from your old system to your new system. The new SPTAPE command ignores these files. System data files include:

- Files created by your old system's ACNT, CPTRAP, or MONITOR commands
- Virtual machine dumps (output from the VMDUMP command), CP abend dumps, and CP stand-alone dumps from either your old system or your new system
- Message repository files
- System trace files.

Spool File Limit Considerations [S370]

In the new system, there is no parameter for limiting the number of spool files in the system. The total number of files allowed in the system is determined by the size of the warm-start area. In your old system, you could control the total number of spool files in the system using the SYSSPL parameter on the SYSRES macro and the size of the checkpoint area.

Also, the new system allows up to 9999 spool files per user. Your old system allowed up to 9900 spool files per user.

What To Do About the New Spool File Limit If You Have To Back Out [S370]

If you have to back out to your old system, a user with more than 9900 spool files will not have spool files 9901-9999 restored. If you use SPTAPE to restore the spool files, however, you still have spool files 9901-9999 on your tape. You can restore these to the user once they reduce their total number of spool files.

To avoid this temporary loss of spool files during a back out, you can use the MAXSPOOL parameter on the SPOOLFILE directory control statement to limit a user to a maximum of 9900 spool files.

You can, of course, set up a MAXSPOOL parameter with a value of 9900 for every user in the directory. But it is probably more practical to just do this for those users that typically use a large number of spool files, such as service machines.

Procedures for Moving Spool Files [S370]

Because of the incompatibilities in spool file formats between System/370 VM operating systems and the new system, you must perform a cold start when converting to or backing out from the new system.

Use RSCS or the SPTAPE command to transport the spool files back and forth. The following table shows you when to use RSCS and when to use SPTAPE.

If you:	Then use:
Bring up a System/370 guest on VM/ESA 2.4.0 and have RSCS service machines running on both the System/370 system and VM/ESA 2.4.0.	RSCS to send the System/370 spool files to the new system
Either • Do not bring up a System/370 guest, or • Do not have RSCS service machines running on both the System/370 system and VM/ESA 2.4.0	The SPTAPE command to convert spool files from the System/370 system to VM/ESA 2.4.0

Procedure for Converting Spool Files Using SPTAPE [S370]

You can convert spool files from your old system to your new system using the CP SPTAPE command:

1. Backup important spool files before conversion. Use the SPTAPE DUMP command to put these important spool files on tape.
2. Shutdown the old system.
3. IPL the new system with a COLD start.

   Note: It is important to SHUTDOWN and IPL in two separate steps. In other words, do not use SHUTDOWN REIPL.
4. Move the tape with spool files over to your new system.
5. Restore your backed up spool files onto your new system. Use the SPTAPE LOAD command to move the spool files you dumped using SPTAPE dump to the new system.

In addition, if you move a spool file that originated on a System/370 VM operating system to a VM/ESA 2.4.0 system, file attributes such as date and spool ID number change. Therefore, if you move the same file back to the System/370 VM operating system, it may not be identical with the original file.

Estimating How Long SPTAPE Will Take to Convert Spool Files [S370]

The amount of time that SPTAPE processing takes depends directly on the number of spool file records in your system. For example, it will take you roughly twice as long to SPTAPE dump 50000 records as it will to dump 25000 records.

To get a rough estimate of how long it will take to dump the spool files in your system, do the following:

1. Measure how long it takes to dump a small number, such as 1000 records.
2. Determine how many spool file records you have in the system. Exclude monitor and dump records, which are not transferable to the new system and which have a different length (4KB) than other spool file records (80 bytes). You should process these records before converting to the new system.
3. Multiply the number of spool file records in your system by the amount of time it takes per record.

For example, suppose you find that it takes about 80 seconds for you to dump 1000 records, or .08 seconds per record. Now assume that you have 110000 spool file records in your system. By multiplying 110000 by .08, you find that it will take 8800 seconds, or almost 2-1/2 hours, to dump all the spool files in your system. This number is not exact, but it gives you an idea of how much time to allot.

You can decrease the amount of actual time it takes if you attach several tape drives and dump spool files from several virtual machines at once. Or you may consider dumping spool files over several nights.

Converting Spool Files and Saved Segments [ESA]

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

Notes:

1. The IPL process handles conversion of spool files during IPL, but as a precaution you may want to consider doing your own spool file conversion, especially for important spool files.
2. If you are converting from VM/ESA 1.2.2 or later, use the SPXTAPE command to dump files from the old system and load them on the new system. If you are converting from a release prior to VM/ESA 1.2.2, you must use the SPTAPE command for both parts of this process. SPTAPE and SPXTAPE are not compatible. SPXTAPE cannot load or scan files dumped by SPTAPE.

   Also, refer to the CP Command compatibility table for your old release to review possible compatibility problems between the old and new versions of SPTAPE.

In VM/ESA 1.2.0 and later releases, enhancements to VMSES/E provide the capability to:

- Plan and map saved segment layouts without affecting your production system
- Build saved segments for products using a common interface.

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 *VM/ESA: 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 convert to the new system, you can use different methods for converting your old system's spool files and saved segments to your new system.

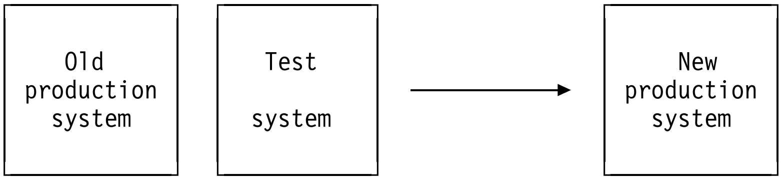
Method	When you can use this method
Using the same warm start and checkpoint areas	• When you are ready to cut over completely from your old system to your new system. • To convert all spool files and saved segments.
Using SPXTAPE or SPTAPE	• When you are ready to cut over completely from your old system to your new system, or • When you want to use your new system as a test system before cutting over completely. • When you want to convert all your spool files and saved segments, or • When you want to convert selected spool files or saved segments.
Building saved segments individually	• Anytime.

Using the Same Warm Start and Checkpoint Areas to Convert All Spool Files at Once [ESA]

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 your old production system on VM/ESA 1.2.0 or later, and you also have a VM/ESA 2.4.0 test system that you have been running.



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 either the:

   - CP_OWNED statement in SYSTEM CONFIG, or
   - SYSCPVOL statement in HCPSYS ASSEMBLE

   The order that you list DASD that CP owns is critical. 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. (On releases prior to VM/ESA 1.2.2, you must use the SPTAPE command in this and successive steps instead of SPXTAPE. Do not mix SPXTAPE and SPTAPE commands.)

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 from your new test system. You will later (in step 6 on page 93) load these back on your new system that becomes your production system. You have to do this because in the next step you IPL your new system whose warm start and checkpoint areas are the same 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 those spool files and saved segments from your new test system that were dumped in step 4.

7. Enter `Q NSS` to see what saved segments and saved systems your new system has. 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 re-IPL your new system, or enter `xautolog autolog1` to start up all of your system's service machines.

   Attention: Make sure your old production system does not get autologged. Because it is defined with the same checkpoint and warm start areas, it uses your VM/ESA 2.4.0 areas and 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 [ESA]” on page 94 for a brief description of VMFSGMAP, and for further details on managing saved segments see the *VM/ESA: Planning and Administration* book.

Using the SPXTAPE Command to Convert All or Some of Your Spool Files [ESA]

Another method for converting your spool files is to use SPXTAPE. (On releases prior to VM/ESA 1.2.2, you must use the SPTAPE command instead of SPXTAPE. Do not mix SPXTAPE and SPTAPE commands.) You can use this method if you want to convert 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 convert just those spool files that contain the 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 convert to your new system.

2. On your old system, use SPXTAPE to dump to tape all of your spool files or just those that you want to convert 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 will load on your system in the next step.

4. On your new system, use SPXTAPE to load the spool files you dumped in step 2.
5. Enter `Q NSS` to see what saved segments and saved systems your new system has. 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 [ESA]” for a brief description of VMFSGMAP, and for further details on managing saved segments see the *VM/ESA: Planning and Administration* book.

Building Saved Segments Individually [ESA]

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 the example below.

Table 10. Example of Saved Segment Information in an ESA Conversion

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 convert to your new system, you may gather this information in two ways:

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

   Although it may require more time planning for your conversion, you should consider mapping your entire saved segment layout at once. Otherwise, as you gradually install additional products, you may have to re-map and re-build 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.

a. Whether the saved segment is a CMS logical segment or a CP physical segment
b. Whether the saved segment can reside above the 16MB line
c. 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 either. 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 VMSES/E's VMFBLD exec. You input this information to VMSES/E using the VMFSGMAP interface.

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

   VMFSGMAP is a VMSES/E exec that provides a saved segment planning and mapping interface. 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 *VM/ESA: Planning and Administration* book.

   When manipulating your system's saved segment layout:

- Remember that CMS requires 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 cannot be run above the 16MB line. Also, make sure that all segments that you need available for users of 370 virtual machines are mapped below the 15MB 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 input 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 *VM/ESA: Planning and Administration* book.

Avoiding the Loss of Spool Files and System Data Files During Conversion [ESA]

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 conversion 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:

- Back up your spool files and system data files using SPXTAPE DUMP (SPTAPE DUMP on releases prior to VM/ESA 1.2.2) before reallocating SPOOL space
- Reallocate the SPOOL space
- Purge your system data files from the old system
- Shutdown the old system; **do NOT use SHUTDOWN REIPL**
- IPL the new system with a cold start
- Restore the backed up spool files and system data files using SPXTAPE LOAD (SPTAPE LOAD on releases prior to VM/ESA 1.2.2).

Sharing Data between Your Old System and Your New System [ALL]

If you convert 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.

VM/ESA 2.4.0 software 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.

For conversions from VM/ESA 1.2.0 and later, virtual reserve/release, real reserve/release, and concurrent virtual and real reserve/release work the same in the new system as they did in your old system.

Differences in Reserve/Release [S370]

For conversions from System/370, virtual reserve/release works the same as in your old system.

Real reserve/release works:

- The same for dedicated DASD
- Differently for nondedicated DASD

In the new system, the following are now required for real reserve/release on a nondedicated DASD:

- SHARED YES on the RDEVICE statement in the system configuration file; for example:

  ```
  RDEVICE 0327 TYPE DASD SHARED YES
  ```

- A full-pack minidisk that has the V suffix included for the access mode on its MDISK statement; for example:

  ```
  MDISK 197 3350 DEVNO 200 WORKPK MWV ORANGE
  ```

Reserve/Release Considerations for VSE [ALL]

VM/ESA 2.4.0 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 VM/ESA 2.4.0 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: VM/ESA 2.4.0 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 [ALL]

The sections that follow 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 *VM/ESA: Planning and Administration* book.

What If We Cannot Share Data through Reserve/Release? [ALL]

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 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 VM/ESA 2.4.0 software; you must set up the controls yourself.
- Use the cross system extensions (CSE) support in VM/ESA 2.4.0. See "Sharing Data among CMS Users on Multiple Systems [ALL]" on page 106 for details.

Sharing Data among Multilevel Virtual Machines [S370]

In a System/370 conversion, 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 [S370]" on page 99 explains how to do this. MVS 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. CMS is an example of a virtual machine that does *not* support reserve/release CCWs. "Without Using Virtual Reserve/Release [S370]" on page 100 shows you how to share data between virtual machines that do not support virtual reserve/release.

Using Virtual Reserve/Release [S370]: 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 in VM/ESA 2.4.0 as it does in your old system. Figure 3 depicts MVS virtual machines sharing DASD through virtual reserve/release in a System/370 conversion.

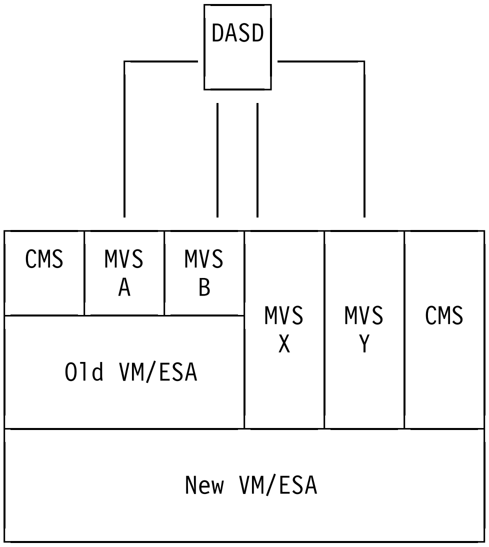


Figure 3. Sharing DASD Using Virtual Reserve/Release in a System/370 Conversion

To set up virtual reserve/release, do the following on the old system guest:

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

Do the following on the new system:

1. Define the DASD where the minidisk resides by updating the SYSTEM CONFIG file on the PARM disk. The PARM disk is a CMS-formatted minidisk that CP can read. It is defined during installation and contains the system definition information needed at IPL, including the CP nucleus module and the SYSTEM CONFIG file.

   To define the DASD, specify its volume identifier on the CP_OWNED statement or on the USER_VOLUME_LIST statement in SYSTEM CONFIG. If the DASD you are using is not automatically sensed by CP, you also need to add an RDEVICE statement to SYSTEM CONFIG. When coding the RDEVICE statement, omit the SHARED operand so that CP does not share the device among multiple real systems. See Table 39 on page 349 to determine whether an RDEVICE statement is needed. More about the PARM disk and SYSTEM CONFIG file is in the *VM/ESA: Planning and Administration* book.

2. Code the MDISK statement in your old system's directory entry in the new 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 HPO 197 197 MW
   ```

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

Without Using Virtual Reserve/Release [S370]: 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 old system guest; the others can only have read access. This is shown in Figure 4.

Note the restriction in “Minidisk Caching Consideration [S370]” on page 105.

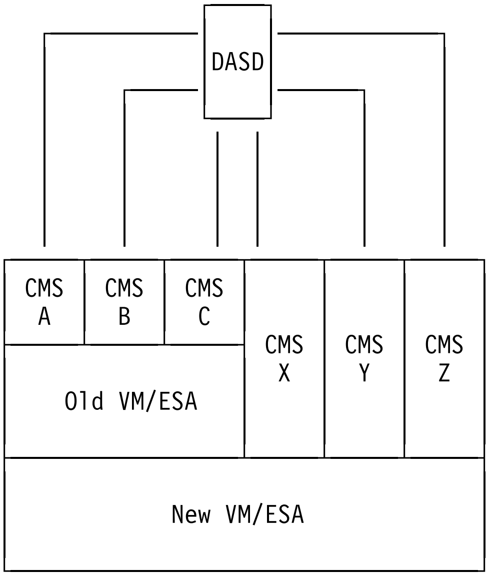


Figure 4. Sharing DASD without Using Virtual Reserve/Release in a System/370 Conversion

To set up DASD sharing without using virtual reserve/release, do the following on the old system guest:

1. Define the DASD as a minidisk for one of the virtual machines, perhaps A, in the old 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 old system directory. Specify RR as the access mode.

Do the following on the new system:

1. Define the DASD where the minidisk resides by updating the SYSTEM CONFIG file on the PARM disk. The PARM disk is a CMS-formatted minidisk that CP can read. It is defined during installation and contains the system definition information needed at IPL, including the CP nucleus module and the SYSTEM CONFIG file.

   To define the DASD, specify its volume identifier on the CP_owned statement or on the USER_VOLUME_LIST statement in SYSTEM CONFIG. If the DASD you are using is not automatically sensed by CP, you also need to add an RDEVICE statement to SYSTEM CONFIG. When coding the RDEVICE statement, omit the SHARED operand so that CP does not share the device among multiple real systems. See Table 39 on page 349 to determine whether an RDEVICE statement is needed. More about the PARM disk and SYSTEM CONFIG file is in the *VM/ESA: Planning and Administration*.

2. Code the MDISK statement in your old system's directory entry in the new 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 HPO 197 197 MW
   ```

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

Sharing Data among Multilevel Virtual Machines [ESA]

In a VM/ESA 1.2.0 or later conversion, 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 [ESA]” explains how to do this. MVS 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. CMS is an example of a virtual machine that does *not* support reserve/release CCWs. “Without Using Virtual Reserve/Release [ESA]” on page 103 shows you how to share data between virtual machines that do not support virtual reserve/release.

Using Virtual Reserve/Release [ESA]: 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 in VM/ESA 2.4.0 as it does in your old system. Figure 5 on page 102 depicts MVS virtual machines sharing DASD through virtual reserve/release in a conversion from VM/ESA 1.2.0 or later.

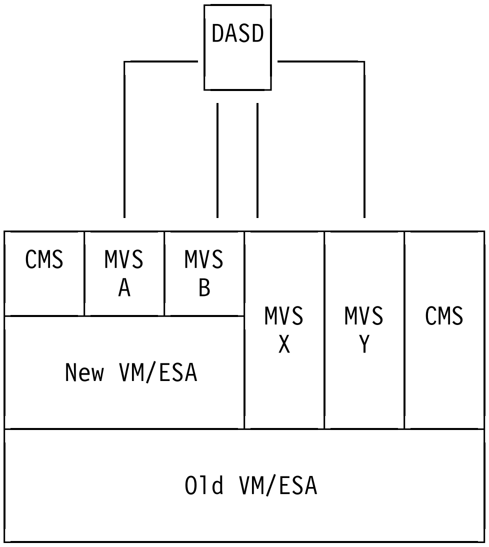


Figure 5. Sharing DASD Using Virtual Reserve/Release in an ESA Conversion

To set up virtual reserve/release, it has to look like concurrent virtual real reserve/release to the guest, which in the scenario depicted in Figure 5 is your new system. Do the following on the new system guest:

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 your first level system, the old 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 [ESA]: 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 only have read access. This is shown in Figure 6.

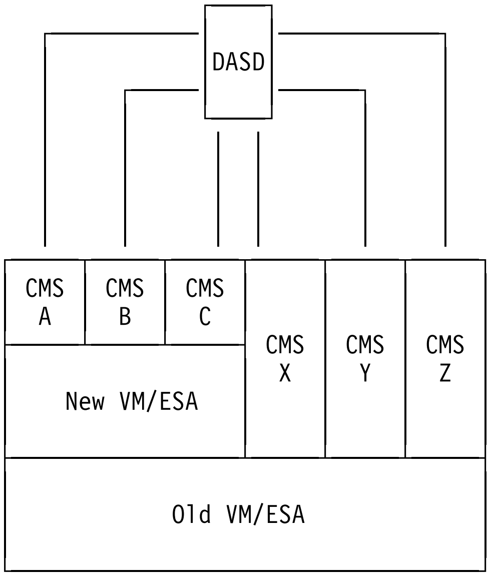


Figure 6. Sharing DASD without Using Virtual Reserve/Release in an ESA Conversion

To set up DASD sharing without using virtual reserve/release, do the following on the new system guest:

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 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. 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.

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

How Do I Share Data among Virtual Machines and Other Systems? [ALL]

To share data among multiple virtual machines running on VM/ESA 2.4.0 and other 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 7, the MVS virtual machines running on the new system can share DASD with the MVS virtual machine running on the old system in this manner. The CMS virtual machines 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.

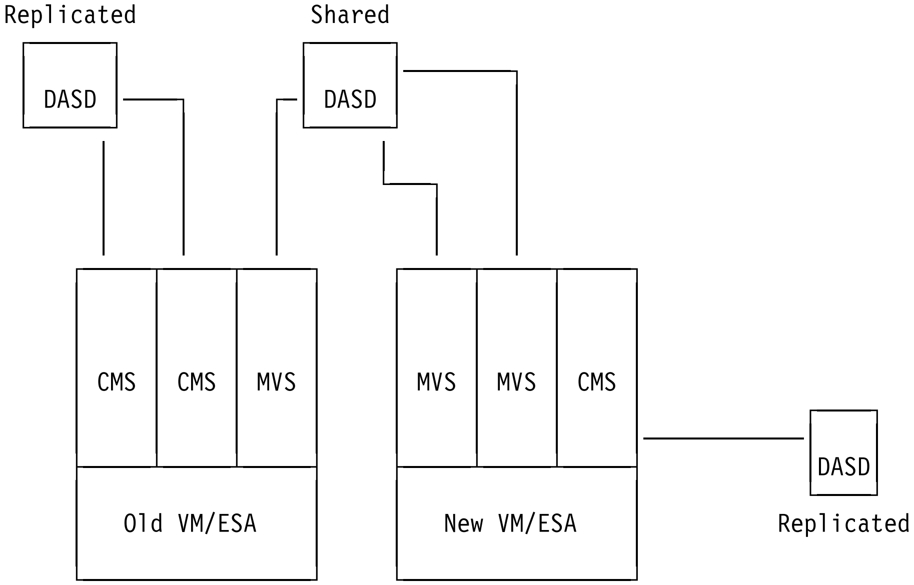


Figure 7. Sharing DASD between Virtual Machines on Multiple Systems

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 on VM/ESA 2.4.0 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.

V=R Recovery [S370]: In a System/370 conversion, If the system terminates because of a hard abend or a machine check, CP attempts to restart with the V=R virtual machine “intact.” V=R recovery in the new system differs from that in your old system. In the new system, CP restores the V=R user's current I/O configuration rather than, as in your old system, the I/O configuration defined in the V=R user's user directory entry. This means that the V=R user recovers the reserves and links to all the dedicated devices and full-pack minidisks to which it is linked at the time of the crash.

Restriction with Concurrent Reserve/Release Support and Real Reserve/Release to a Dedicated Device [S370]

You cannot use concurrent virtual and real reserve/release support with the old system if you generate old system devices with alternate paths. *The definition of alternate paths in DMKRIO and the use of hardware-supported reserve/release are mutually exclusive.* (Alternate paths are supported differently in the new system; they are supported in the hardware rather than the software.)

Minidisk Caching Consideration [S370]

To provide virtual machines with fast access to data on minidisks, VM/ESA 2.4.0 supports the caching in Expanded Storage of the most recently used data blocks from non-full-pack minidisks of 4KB block size. However, to guarantee data integrity, minidisk caching should be disabled for any minidisk that is shared between two systems when one or both systems have write access to the minidisk.

In VM/ESA 1.1.5 370 Feature, there is no minidisk cache. In VM/ESA 2.4.0, 4KB minidisks use cache by default. To disable caching for an individual minidisk, specify the NOMDC parameter on its MINIOPT user directory control statement. To disable caching for all minidisks, enter the command RETAIN XSTORE MDC OFF. (If you use this command, you should place it in the AUTOLOG1 PROFILE.) Also, when the RDEVICE SHARED operand is set to YES, minidisk caching does not take place.

Also, “Aligning Minidisks on 4KB Boundaries for Minidisk Cache and for SFS [1.1.5]” on page 233 contains information about considerations for minidisk cache if your DASD is an FBA device.

Sharing Data among CMS Users on Multiple Systems [ALL]

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 *VM/ESA: Planning and Administration* for detailed information about CSE.

Preparing for Cross-System Link [ALL]

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 *VM/ESA: Planning and Administration*.

Preparing for Cross-System Spool [ALL]

Activation of cross-system spool requires that the spooling volumes be defined to the control programs by the SYSCPVOL macro in HCPSYS or 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. When a SYSTEM CONFIG file is used, 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 VM/ESA 2.4.0. 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.

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.

Creating the Single-Source Directory [S370]: CSE assumes that all systems in the complex operate with the same source directory, that is, with a single source directory contained on one CMS minidisk that is shared and available throughout the complex. Assuming that the CP user directory in use is already the required single-source directory in operation for the complex, the only conversion consideration is to add the necessary VM/ESA 2.4.0 statements to the existing directory entries. The task of maintaining this directory and propagating changes in the single-source directory to the online directories of all systems in the complex is assumed to be handled by the Directory Maintenance Program (DirMaint).

How to Convert Your User Directory [ALL]

You cannot use the object directory created by your old release on VM/ESA 2.4.0 because it is not upwardly compatible. This section helps you convert your old directory to VM/ESA 2.4.0. It discusses:

- Some of the changes made to the default sample directories
- A procedure for converting your user directory
- What to do if you cannot IPL because of user directory problems

Also refer to “Considerations for Converting Your User Directory [ALL]” on page 211.

Do not convert the user directory until you finish installing your VM/ESA 2.4.0 system.

Note: The information shown in this section is most specific for a conversion from VM/ESA 1.2.0 to VM/ESA 2.4.0, but conversion from other releases should be similar.

Changes in the User Directory [ALL]

There are many changes in the directory you create with your new system compared to the directory on your old system. Some of the key changes are:

- Minidisk locations, sizes, and passwords.
- Options and the virtual machine default and maximum sizes for some user IDs.
- Initial logon passwords are now supplied for all user IDs except for the system reserved area user IDs.
- AUTOONLY, NOPASS, and LBYONLY are reserved passwords. If any users already have these passwords, change them unless the users are to take advantage of the function these passwords provide.
- Within the MAINT user ID in the new directory, links are now made to all the server minidisks and to many other system user ID minidisks. The default minidisk links are no longer MW. Minidisks are now linked with the highest level needed, usually MR.
- MAINT minidisks 595 and 59E have been removed because they are no longer used by GCS.

For information about new and changed directory control statements, refer to the user directory control statements compatibility table for your conversion in Part 3, "Compatibility Tables." For a complete description of directories, see *VM/ESA: Planning and Administration*.

Changed Default Volume Labels [ALL]

The default volume labels have changed from your old release to VM/ESA 2.4.0. Table 11 lists the old labels for VM/ESA Version 1 releases and the corresponding new labels. Table 12 lists the old labels for VM/ESA Version 2 releases and the corresponding new labels.

Table 11. Default volume label changes from VM/ESA Version 1 releases

VM/ESA 1.1.5 370 Feature	VM/ESA 1.2.0	VM/ESA 1.2.1	VM/ESA 1.2.2	Volume description	VM/ESA 2.4.0
VMSRES	ES2RES	E21RES	E22RES	System residence volume	240RES
VMPK01	ES2W01	E21W01	E22W01	Additional CP-owned volume	240W01
VMPK02	ES2W02	E21W02	E22W02	Additional CP-owned volume	240W02
VMPK03	ES2W03	E21W03	E22W03	Additional CP-owned volume	240W03
VMPK04	ES2W04	E21W04	E22W04	Additional CP-owned volume	240W04
n/a	ES2W05	E21W05	E22W05	Additional CP-owned volume	240W05
n/a	ES2W06	E21W06	E22W06	Additional CP-owned volume	240W06
n/a	ES2W07	E21W07	E22W07	Additional CP-owned volume	240W07
n/a	n/a	E21W08	E22W08	Additional CP-owned volume	240W08
PROFPK	n/a	n/a	n/a	Reserved for PROFS program product	deleted
SQLPK	n/a	n/a	n/a	Reserved for SQL program product	deleted

Table 12. Default volume label changes from VM/ESA Version 2 releases

VM/ESA 2.1.0	VM/ESA 2.2.0	VM/ESA 2.3.0	Description	VM/ESA 2.4.0
210RES	220RES	230RES	System residence volume	240RES
210W01	220W01	230W01	Additional CP-owned volume	240W01
210W02	220W02	230W02	Additional CP-owned volume	240W02
210W03	220W03	230W03	Additional CP-owned volume	240W03
210W04	220W04	230W04	Additional CP-owned volume	240W04
210W05	220W05	230W05	Additional CP-owned volume	240W05
210W06	220W06	230W06	Additional CP-owned volume	240W06
210W07	220W07	230W07	Additional CP-owned volume	240W07
210W08	220W08	230W08	Additional CP-owned volume	240W08

Before You Convert Your User Directory [ALL]

If you have a problem IPLing VM/ESA 2.4.0, it may be because of changes you have made to the directory. If you suspect this, try using the NODIRECT option when you bring up VM/ESA. This brings up the system without accessing the user directory.

Before you can do this, 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 13 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 13 are examples. Refer to the *VM/ESA: Planning and Administration* book for more information on loading VM/ESA 2.4.0. Refer to the *VM/ESA Installation and Service Sample Files* document packaged with the VM/ESA product for information on the IBM default location and sizes of minidisks.

Table 13. 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 - **example**	2C2	240RES	430	5
CMS system disk - **example**	190	240RES	302	128
System residence volume - **example**	123	240RES	0	END
Minidisk containing source directory				
CMS system disk				
System residence volume				

Considerations for Converting Your User Directory [ALL]

During your conversion 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 VM/ESA 2.4.0 without worrying about backing out.

Considerations for Using a Directory Maintenance Program [ALL]

If you have already installed a directory maintenance program on your VM/ESA 2.4.0 system, or if you are using only one directory, you may have to issue additional commands while converting the source directory.

Example Using IBM Directory Maintence VM/ESA (DirMaint) [ALL]

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.
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:

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

Steps for Converting Your User Directory [ALL]

Whether you use two directories or one during conversion, you can convert 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 VM/ESA 2.4.0 directory and merge pieces from the old directory into it as the old directory entries are converted. To convert the old directory entries:

   a. Convert each system user ID.

      IBM supplies a sample directory with the VM/ESA product. In this directory, there are several system user IDs defined.

      S370

      Table 14 shows the system user IDs that have changed from System/370 to VM/ESA 2.4.0.

Table 14 (Page 1 of 2). Default System User ID Differences between System/370 and VM/ESA 2.4.0

System/370	VM/ESA 2.4.0	Description
$CPNUC$	deleted	Reserves space for the CP nucleus **Note:** This user ID has been replaced by a CP nucleus module.
$TDISK$	$T-DISK$	Reserves space for temporary disks

Table 14 (Page 2 of 2). Default System User ID Differences between System/370 and VM/ESA 2.4.0

System/370	VM/ESA 2.4.0	Description
CMSUSER	CMS1	Sample CMS user entry
$TEMP$	deleted	Replaced by $SPOOL$ and $PAGE$
$SAVSYS$	deleted	Saved System area
$SYSERR$	deleted	EREP Error Recording area
$OVRD$	deleted	Override area
n/a	$PAGE$	Reserves paging space
n/a	$SPOOL$	Reserves spooling space
n/a	LGLOPR	PROP logical operator
n/a	GCSXA	GCS recovery virtual machine
n/a	SYSMAINT	
n/a	OP1	Alternate OPERATOR user ID
n/a	DISKACNT	System accounting user ID
n/a	OPERSYMP	Collects symptom records
n/a	VMUTIL	

For each system user ID that you used in your old system, verify that your directory entries are applicable in VM/ESA 2.4.0.

- 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.

S370

See “CMS Nucleus Changes” on page 230 for details on the storage needed to IPL a nonshared copy of CMS.

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 in VM/ESA 2.4.0.

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 converted 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. 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? [ALL]

If you have a problem IPLing VM/ESA 2.4.0, you can try using the NODIRECT option. Because VM/ESA 2.4.0 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 13 on page 109 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 13 on page 109.

   For example, using the sample information in Table 13 on page 109:

   ```
   define mdisk as 2c2 430 5 240res
   define mdisk as 190 302 128 240res
   define mdisk as 123 0 END 240res
   ```

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

The following is an example of the messages you receive when you choose to IPL your system with the NODIRECT option.

1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0

Also see "IPLing with the NODIRECT Option [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]" on page 284.

```
12:10:16 VM/ENTERPRISE SYSTEMS ARCHITECTURE  V2 R4.0  SERVICE LEVEL  0000;
12:10:16 SYSTEM NUCLEUS CREATED ON 1999-01-25 AT 19:48:01, LOADED FROM 240RES
12:10:17
12:10:17 ****************************************************************
12:10:17 * LICENSED MATERIALS - PROPERTY OF IBM*                        *
12:10:18 *                                                              *
12:10:18 * 5654-030 (C) COPYRIGHT IBM CORP. 1983, 1999. ALL RIGHTS      *
12:10:18 * RESERVED. US GOVERNMENT USERS RESTRICTED RIGHTS - USE,       *
12:10:18 * DUPLICATION OR DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE     *
12:10:19 * CONTRACT WITH IBM CORP.                                      *
12:10:19 *                                                              *
12:10:19 * * TRADEMARK OF INTERNATIONAL BUSINESS MACHINES.              *
12:10:20 ****************************************************************
12:10:20
12:10:20 HCPZCO6718I Using parm disk 1 on volume 240RES (device 01A0).
12:10:20 HCPZCO6718I Parm disk resides on cylinders 262 through 281.
12:10:20 Start ((Warm|Force|COLD|CLEAN) (DRain) (DIsable)  (NODIRect)
12:10:20 (NOAUTOlog)) or (SHUTDOWN)
12:12:00 NODIRECT
12:12:00 NOW 12:12:00 EST TUESDAY 1999-01-26
12:12:00 Change TOD clock (Yes|No)
12:12:04 NO
12:12:13 HCPWRS2513I
12:12:13 HCPWRS2513I Spool files available       24
12:12:22 HCPUDR1507E The directory is not valid.
12:12:22 HCPWRS9309E CP was unable to allocate a soft abend dump file.
12:12:22 HCPWRS2512I Spooling initialization is complete.
12:12:34 DASD 01A0 dump unit CP IPL pages 1729
12:12:34 There is no logmsg data
12:12:34 FILES:   NO RDR, 0001 PRT,   NO PUN
12:12:34 LOGON AT 12:12:34 EST TUESDAY 01/26/99
12:12:34 GRAF  001F LOGON  AS  OPERATOR USERS = 1
12:12:34 HCPIOP952I 0016M system storage
12:12:34 FILES: 0000006 RDR, 0000002 PRT,      NO PUN
12:12:34 HCPAAU2700I System gateway ESAV2R40 identified.
12:21:53 * DEFINE S-DISK
12:22:05 DEFINE MDISK 190 302 128 240RES
12:22:05 DASD 0190 DEFINED
12:22:26 * DEFINE SYSRES
12:23:18 DEFINE MDISK 123 0 END 240RES
12:23:18 DASD 0123 DEFINED
12:23:35 * DEFINE USER DIRECT MINIDISK
12:28:11 DEFINE MDISK 2C2 430 5 240RES
12:28:11 DASD 02C2 DEFINED
12:28:22 * IPL THE 190 TO GET CMS
12:29:24 IPL 190
DMSIND2015W Unable to access the Y-disk. Filemode Y (19E) not accessed
12:29:26 HCPUDR1507E The directory is not valid.
12:29:26 HCPUDR1507E The directory is not valid.
DMSINS2050W Unable to provide CMS support for IUCV and APPC/VM; return co
DMSWSP314W Automatic re-IPL by CP; no information available
VM/ESA V2.4.0 1999-01-26 13:04

DMSACP113S A(191) not attached or invalid device address
Ready; T=0.04/0.06 12:29:49
12:29:56 * GET THE SOURCE DIRECTORY
12:30:00 TERM MODE VM
acc 2c2 b
```

```
Ready; T=0.01/0.01 12:30:04
* fixup the source directory by using xedit
xedit user direct b
Ready; T=0.01/0.01 12:31:47
* place the directory on line
directxa user direct b
VM/ESA USER DIRECTORY CREATION PROGRAM - VERSION 2 RELEASE 4.0
EOJ DIRECTORY UPDATED AND ON LINE
Ready; T=0.06/0.08 12:32:01
```

Considerations for Defining a Second-Level System on VM/ESA 2.4.0 [S370]

This section describes some options that you must consider when running a second-level VM system on top of VM/ESA 2.4.0.

You can run the guest VM operating system as one of three types of virtual machine:

- Virtual = real machine (V=R)
- Virtual = fixed machine (V=F) - PR/SM required and machine must run in basic mode
- Virtual = virtual machine (V=V).

Before you decide which type of virtual machine to run your guest in, you must understand how real storage is arranged in a VM/ESA 2.4.0 system. See *VM/ESA: Running Guest Operating Systems* for information on running guests on VM/ESA 2.4.0. This book also describes how to define VM as a second-level system on VM/ESA 2.4.0.

You must also realize the performance implications and benefits for configuring your system with guests running as one of these types of virtual machines. For instance, running your second-level system as a preferred guest (V=R or V=F) means that you have to dedicate a certain amount of real storage for use by only that guest machine. Because real storage is dedicated to the guest, the guest may see a performance enhancement. However, other virtual machines and guests in that same system may see a decrease in their performance.

There are advantages to dedicating devices to a preferred guest. For example:

- A V=R machine does not contend with other virtual machines for dedicated devices.
- When running the real processor in Basic mode (not LPAR), the V=R machine can use I/O interpretation which allows many of the I/O requests to be processed without assistance from CP (use SET IOASSIST command).

Other advantages and considerations for running preferred guests are described in *VM/ESA: Running Guest Operating Systems*.

Configuring your First-Level Virtual Machine [S370]

The first-level virtual machine will be the real machine for your second-level system. You must make the proper definitions in your first-level virtual machine in order to run a second-level system successfully.

In the first-level system directory, you may use the following entries for the user ID that will be running the second-level system:

- If you decide to run your second-level system as a preferred guest, code one of the following in the user directory:

  ```
  OPTION VIRT=REAL - for V=R guests
  OPTION VIRT=FIXED - for V=F guests
  ```

  Note that you can have up to six preferred guests, either:

  – One V=R guest and up to five V=F guests, or
  – Up to six V=F guests.

  The number of preferred guests also depends on how much real storage you have available to dedicate to guests.

- To define a dedicated processor for your guest, use the CPU directory control statement (you must have a real multiprocessing environment). For example, to dedicate CPU number 00:

  ```
  CPU 00
  ```

- To define a dedicated vector for your guest, use the CPU directory control statement. For example, to dedicate vector 00 to the guest:

  ```
  CPU 00 VECTOR
  ```

  Note that you can also define a processor to the guest without dedicating it to that guest. For example:

  ```
  CPU 00 VECTOR NODEDICATE
  ```

- Define appropriate SPECIAL statements for addresses of graphics devices used for logon purposes. For example:

  ```
  SPECIAL 22 3270
  SPECIAL 23 3270
  SPECIAL 24 3270
  SPECIAL 25 3270
  SPECIAL 26 3270
  SPECIAL 27 3270
  SPECIAL 28 3270
  ```

- Use the LINK and MDISK directory control statements in the first-level system to define minidisks that can be used by the second-level system. The following steps outline how to do this and give an example of making the first-level 191 minidisk available to a second-level system.

  1. Code the MDISK statement in the first-level system:

     ```
     MDISK 0191 3380 1187 010 240RES MR READ WRITE MULTI
     ```

  2. Define the addresses for the disks in the DMKRIO ASSEMBLE file of the second-level system.

  3. To get access to these disks from the second-level system, enter the following commands from a user ID in the second-level system:

```
    ATTACH 191 * vdev     /* vdev is the address known to that user id */
    ACCESS vdev fmode      /* fmode is a CMS file mode */
```

- VM/ESA supports alternate processor (AP) and multiprocessor (MP) environments for virtual machines. For running a second-level system, a 370 virtual machine must run as a uniprocessor (UP) or AP.

Sample Directory Entry [S370]

The following directory entries in the first-level directory describe some of the operating characteristics of the second-level System/370 VM operating system. This example specifies enough characteristics so that when logged on (or autologged), the second-level system will come up automatically. It includes some entries for defining a second-level system but your installation and desired characteristics may vary from this sample.

```
USER VM370 VM370PW  8M 64M ABCDEFG
 ACCOUNT VM370 SECLEVEL
*
 OPTION VIRT=REAL TODENABLE QUICKDSP CPUID 123370 MAINTCCW
*
* set the machine mode to 370 and automatically IPL the system
 MACH 370
 IPL E00 PARM=AUTOLOG
*
* define the virtual CPU on which the second-level system will run
 CPU 00 VECTOR BASE NODEDICATE
*
* give these dasd to the second-level system
 DEDICATE 0E00 0E00 VOLID 370RES
 DEDICATE 0E01 0E01 VOLID 370K01
 DEDICATE 0E02 0E02 VOLID 370K02
 DEDICATE 0E03 0E03 VOLID 370K04
*
* define simulated virtual devices
 SPECIAL 6D0 3088
 SPECIAL 6D1 3088
 SPECIAL 6D2 CTCA TSAF
 SPECIAL 6D3 3088
 SPECIAL 6D4 CTCA VTAM
 SPECIAL 021 3270
 SPECIAL 022 3270
 SPECIAL 023 3270
 SPECIAL 024 3270
 SPECIAL 025 3270
 SPECIAL 026 3270
 SPECIAL 027 3270
 SPECIAL 028 3270
 SPECIAL 029 3270
*
* the virtual console addr must run as a 3270 type,
* and the userid specified here as MAINT is the secondary console
* for routing messages if this userid is running disconnected
 CONSOLE 001F 3270 MAINT
*
* define virtual unit record devices
 SPOOL C0 2540 READER *
 SPOOL D0 2540 PUNCH A
 SPOOL E0 1403 A
*
* these dasd will be available to the second-level system
 LINK MAINTESA   100 100 RR
 LINK MAINTESA   19E 19E RR
 LINK TOOLS      528 528 RR
 LINK CMSUSER    19F 19F RR
 MDISK 0191 3380 1187 010 240RES MR READ WRITE MULTI
```

Establishing Connectivity between First-Level and Second-Level Systems [ALL]

This section describes the considerations for setting up connectivity between:

- A first-level VM/ESA 2.4.0 system and a second-level System/370 system, or
- A first-level VM/ESA 1.2.0 (or later) system and a second-level VM/ESA 2.4.0 system

This section shows connectivity using products that implement APPC/VM and products that run on GCS and are SNA applications.

For example, APPC/VM is used by the Shared File System, Structured Query Language (SQL/DS), DATABASE 2™ (DB2 Server for VM), and OfficeVision/VM™. If you use any of these, you may want to enable communication between your old and new systems to facilitate conversion.

TSAF [ALL]

VM/ESA systems can use the **Transparent Services Access Facility (TSAF)** for peer-to-peer APPC/VM communications. TSAF provides users with transparent access to server resources across a collection of up-to-eight systems. The systems in the collection can be any of the following:

- All VM/ESA releases
- VM/SP or VM/SP HPO Release 6
- VM/SP or VM/SP HPO Release 5

In VM/ESA 2.4.0, TSAF runs in an XA virtual machine with 370ACCOM feature enabled. The default in the TSAF virtual machine's user directory entry is MACHINE XA and the profile exec must include CP SET 370ACCOM ON.

Figure 8 on page 118 shows the relationship between first-level and second-level TSAF machines in a System/370 conversion. Figure 9 on page 118 shows the relationship between first-level and second-level TSAF machines in an ESA conversion.

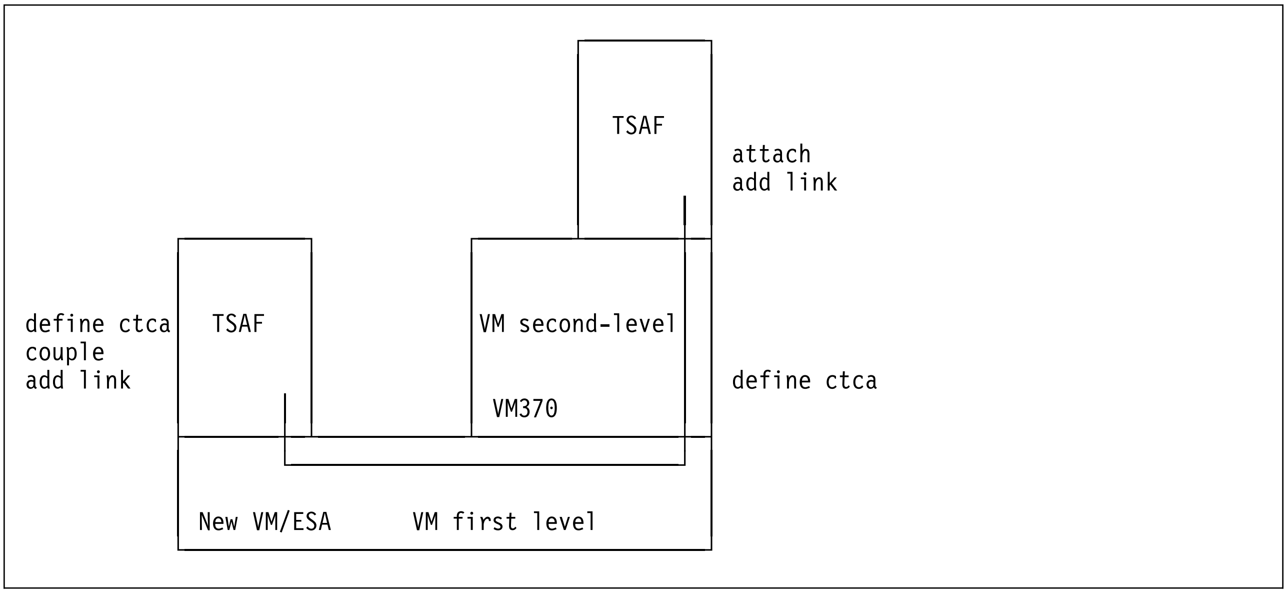


Figure 8. Connectivity between First- and Second-Level TSAF Machines in a System/370 Conversion

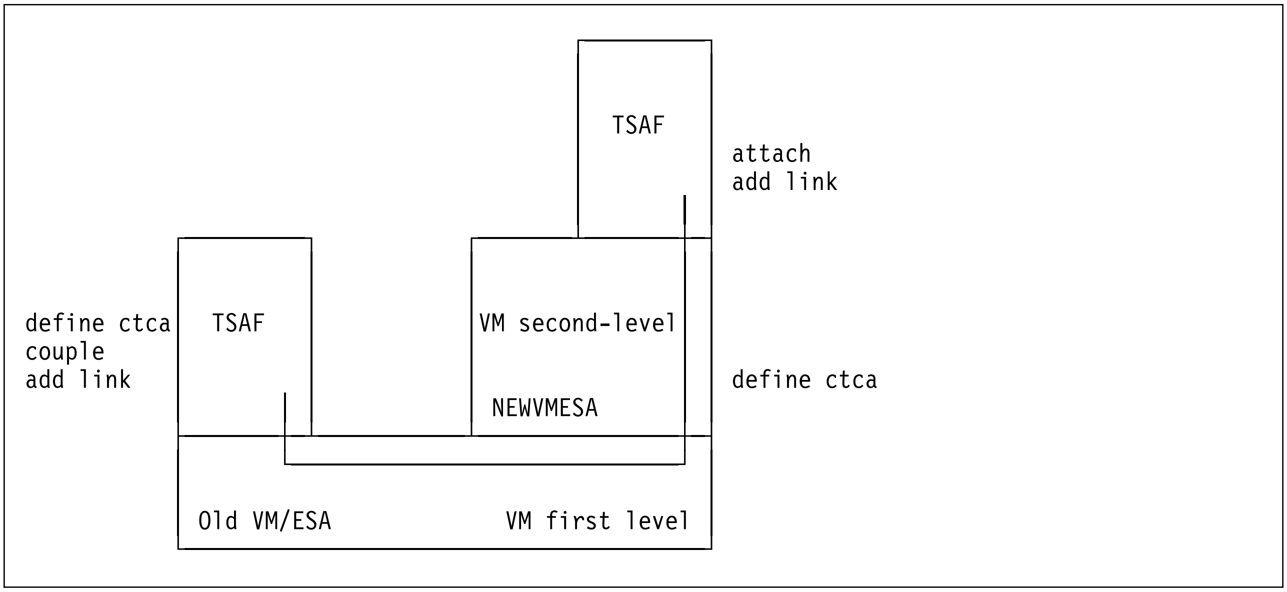


Figure 9. Connectivity between First- and Second-Level TSAF Machines in an ESA Conversion

Steps to follow for First-to-Second-Level System Connectivity [ALL]

On the First-level System:

1. Define a link between the first-level TSAF 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:

   - **[S370]** VM370
   - **[ESA]** NEWVMESA

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

a. From **both** the first-level TSAF virtual machine and the VM370 or 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 the TSAF virtual machine and the VM370 or NEWVMESA user ID, 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:

   - **[S370]** DMKRIO
   - **[ESA]** SYSTEM CONFIG file or HCPRIO

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

2. From either the TSAF machine or the VM370 or NEWVMESA user ID, couple the addressed links to enable communications over the virtual channel. For example, to couple the links from the first-level TSAF virtual machine, you can enter one of the following commands:

   - **[S370]**:

     ```
     couple 312 vm370 312
     ```

   - **[ESA]**:

     ```
     couple 312 newvmesa 312
     ```

3. From the TSAF virtual machine console, add the link to the TSAF collection. For example:

   ```
   add link 312
   ```

4. Bring up the second-level system, including the second-level TSAF machine.

On the Second-level System:

1. From the second-level TSAF virtual machine, attach the channel address defined by VM370 or NEWVMESA. For example, enter:

   ```
   attach 312 tsafvm 312
   ```

2. From the second-level TSAF virtual machine console, add the link to the collection. For example, enter:

   ```
   add link 312
   ```

Samples [ALL]

Because you must perform the preceding steps each time you start the TSAF virtual machines, you may want to automate the procedure.

In the following examples, 312 is the address of the link, and VM370 or NEWVMESA is the user ID that runs the second-level system.

Sample Exec for Setting Up TSAF Connectivity in the User ID That Will Run the Second-level System: From the VM370 or NEWVMESA user ID, issue the DEFLINK exec to activate the link on the first-level side.

```
/*** DEFLINK EXEC ***/
address command
'CP DEFINE CTCA 312'
```

Set Up the First-level TSAF Profile: To automatically activate the link from the first-level TSAF machine's side, specify the following commands in the TSAF first-level PROFILE EXEC:

- **[S370]**:

  ```
  /* TSAF first-level profile */
  'CP DEFINE CTCA 312'
  'CP COUPLE 312 VM370 312'
  ```

- **[ESA]**:

  ```
  /* TSAF first-level profile */
  'CP DEFINE CTCA 312'
  'CP COUPLE 312 NEWVMESA 312'
  ```

Set Up the Second-level TSAF Profile: The second-level TSAF machine must be configured the same as the first-level machine. You can put the following in the second-level TSAF PROFILE EXEC:

```
/* TSAF second-level profile */
'CP ATTACH 312 * 312'
```

VTAM [ALL]

The following steps describe how to set up communications using **Advanced Communications Function for Virtual Telecommunications Access Method (ACF/VTAM or VTAM)** between your first-level and second-level systems. Figure 10 shows the relationship between first-level and second-level VTAM machines in a System/370 conversion. Figure 11 on page 121 shows the relationship between first-level and second-level VTAM machines in an ESA conversion.

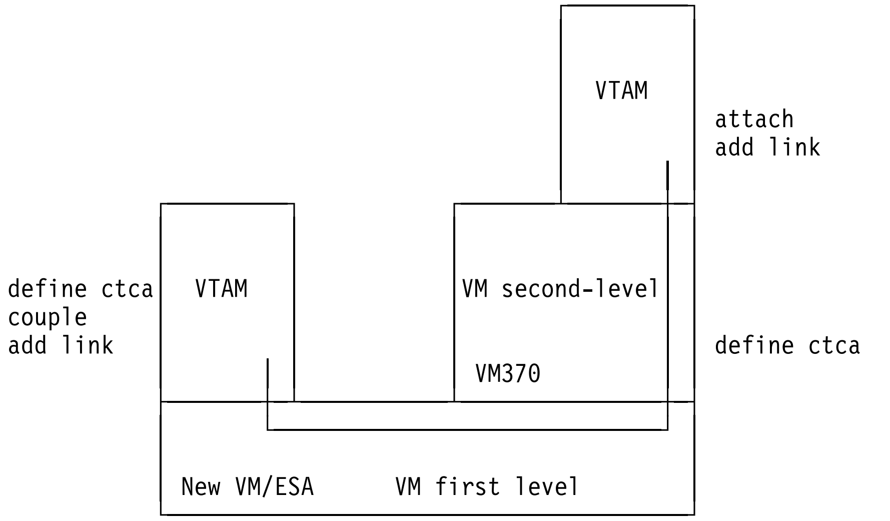


Figure 10. Connectivity between First- and Second-Level VTAMs in a System/370 Conversion

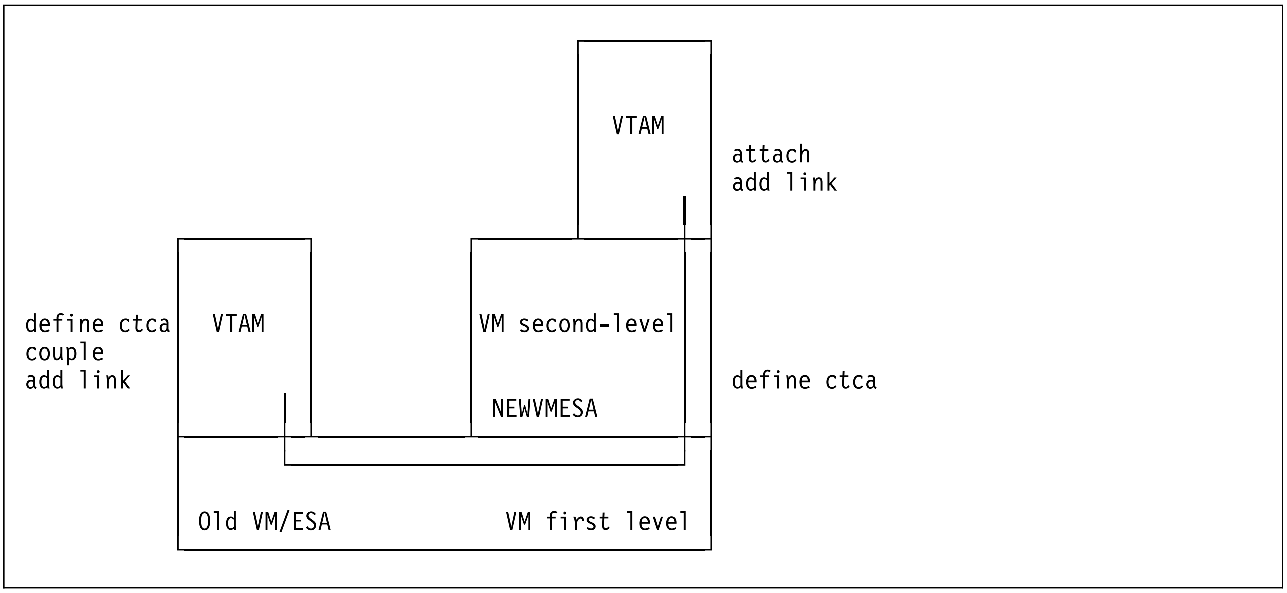


Figure 11. Connectivity between First- and Second-Level VTAMs in an ESA Conversion

Steps to Follow for First-to-Second-Level System Connectivity [ALL]

Establish VTAM on Your First-level System:

1. Install VTAM on your first-level system, which is:
   - **[S370]** Your new VM/ESA 2.4.0 system
   - **[ESA]** Your old VM/ESA system
2. Duplicate desired or necessary resources from the definitions of your old VTAM in your new VTAM.
3. Attach the necessary devices to the new VTAM.
4. Test the attached devices on the new system. For example, define some users on your new system that are attached to a certain control unit. See if the users can log on properly.

On the first-level system:

1. Define a link between the first-level VTAM 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:

   - **[S370]** VM370
   - **[ESA]** NEWVMESA

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

   a. From **both** the first-level VTAM virtual machine and the VM370 or 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 of the VM370 or NEWVMESA user ID and the VTAM virtual machine, specify the following statement:

```
SPECIAL CTCA 312
```

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

- **[S370]** DMKRIO
- **[ESA]** SYSTEM CONFIG file or HCPRIO

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

2. From either the VTAM machine or the VM370 or NEWVMESA user ID, couple the addressed links to enable communications over the virtual channel. For example, from the first-level VTAM virtual machine, you can enter:

   - **[S370]:**

     ```
     couple 312 vm370 312
     ```

   - **[ESA]:**

     ```
     couple 312 newvmesa 312
     ```

3. Bring up the second-level system, including the second-level VTAM machine.

On the second-level system:

1. From the second-level VTAM virtual machine, attach the channel address defined by VM370 or NEWVMESA. For example, enter:

   ```
   attach 312 vtam 312
   ```

Setting Up VTAM to Participate in the SNA Network: The preceding steps set up the channel of communications between the two systems. The VTAM administrator must then make definitions in the new VTAM for participation in the SNA network. This requires unique definitions of the following:

- Subareas
- Terminals
- Application major nodes
- All minor nodes

Defining GROUP, LINE, and PU Statements: In defining the channel-to-channel connection between the first-level and second-level VTAM machines, you must define GROUP, LINE, and PU statements on both systems; for example:

```
CTC312V  VBUILD TYPE=CA
CTC312G  GROUP  LNCTL=CTCA,REPLYTO=10.0,MIH=YES
CTC312L  LINE   ADDRESS=312,MAXBFRU=2
CTC#312  PU     DELAY=.100,PUTYPE=4,TGN=1
```

Other considerations: A complete description of setting up a channel-attached (TYPE=CA) connection and considerations for CDRM and CDRSC definitions for communications over separate SUBAREAs are described in the *VTAM Resource Definition Reference* and *VTAM Network Implementation Guide* books.

When You Are Ready to Remove Your Old VTAM [ALL]

When you are ready to remove your old system from the VTAM network, and you are satisfied with the function of VTAM on your new system:

1. Attach all devices and links to your new system
2. Duplicate all VTAMLST definitions
3. Remove the old VTAM.

AVS [ALL]

APPC/VM VTAM Support (AVS) provides an APPC/VM connection to VTAM, which allows APPC/VM programs to communicate over an SNA network to other APPC programs. AVS runs in a GCS group and requires VTAM to communicate with an SNA LU 6.2 network. Figure 12 shows the relationship between first-level and second-level AVS machines in a System/370 conversion. Figure 13 shows the relationship between first-level and second-level AVS machines in an ESA conversion.

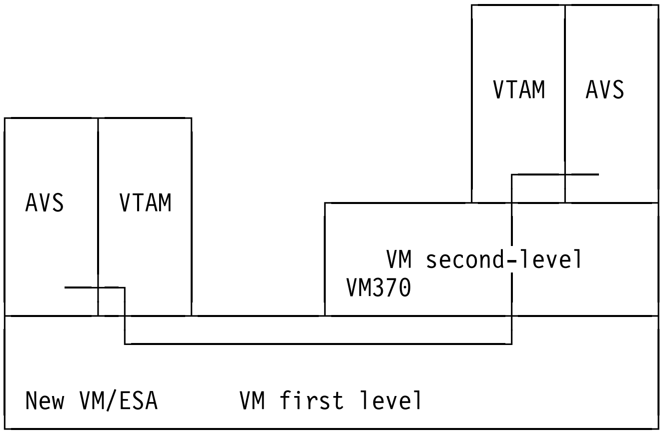


Figure 12. Connectivity between First- and Second-Level AVS Machines in a System/370 Conversion

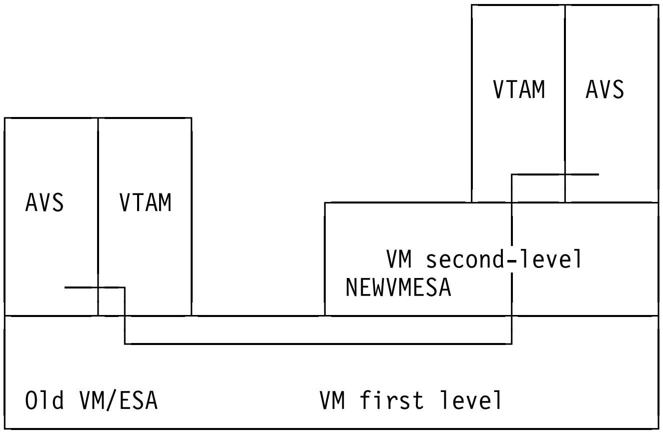


Figure 13. Connectivity between First- and Second-Level AVS Machines in an ESA Conversion

Steps to follow for First-to-Second-Level System Connectivity [ALL]

Defining AVS to VTAM: Before you can use AVS to communicate with programs in the SNA network, you must define the AVS virtual machine to VTAM. To define the AVS virtual machine to VTAM, you must make the following definitions on VTAM, either changing or adding them:

1. Set up a logon mode table.
2. Define an application program major node by entering one VTAM VBUILD statement for the major node and an APPL definition statement for each application program in the major node.

Setting Up a Logon Mode Table: A logon mode table (also called a logmode table) contains entries specifying different session parameters, which are sets of rules that describe how a session is conducted. A sample logon mode table, AGWTAB ASSEMBLE, is shipped with VM/ESA. Setting up a logon mode table is described in the *VTAM Resource Definition Reference.*

Defining an Application Program Major Node: Defining an application program major node means that you need to enter a VTAM VBUILD statement for the AVS virtual machine and an APPL statement for each gateway that is managed by this AVS virtual machine. See the *VTAM Resource Definition Reference* for a complete description of the VTAM VBUILD and APPL statements and their parameters.

The following is an example of an application program major node definition for the AVSVM virtual machine and the global gateway GLBLGAT1. Only the relevant parameters are shown.

```
AVSVM    VBUILD TYPE=APPL
GLBLGAT1 APPL   APPC=YES,                                             X
                AUTHEXIT=YES,                                         X
                AUTOSES=10,                                           X
                DSESLIM=100,                                          X
                DMINWNL=50,                                           X
                DMINWNR=50,                                           X
                MODETAB=AGWTAB,                                       X
                PARSESS=YES,                                          X
                SYNCLVL=SYNCPT,                                       X
                SECACPT=ALREADYV
```

Setting Up the AVS Virtual Machine Profile [ALL]

Before entering any commands that control the AVS virtual machine, the following must be done:

- Load the AVS load library.
- Define the command name for the AVS load module.
- Start the AVS virtual machine.

The profile that comes with the AVSVM virtual machine is set up to perform these functions. This profile is executed when the GCS saved segment is initialized in the AVS virtual machine.

Setting Up the AGWPROF GCS Exec [ALL]

As the last step in AVS initialization, AVS invokes an AVS exec called AGWPROF GCS. You should include commands in this exec that you want issued after AVS has completed initialization. These would usually be AGW ACTIVATE GATEWAY and AGW CNOS commands. The following is an example of the AGWPROF GCS exec.

```
/* AGWPROF GCS activates one gateway for global resource */
/* communications and sets values for the session.       */
'AGW ACTIVATE GATEWAY GLBLGAT1 GLOBAL'
'AGW CNOS GLBLGAT1 REMOTLU1 TABLDAT 10 5 5'
```

This example activates the gateway GLBLGAT1. The AGW CNOS command defines a maximum of ten sessions that can be established between the gateway GLBLGAT1 and the remote LU REMOTLU1.

For connecting the first-level and second-level VM systems, you must define and activate both gateways to communicate with each other. Make sure that the gateways have unique names.

For more information on AVS operation and autologging the GCS, VTAM, and AVS when the system is started, see the *VM/ESA: Connectivity Planning, Administration, and Operation* book.

RSCS [ALL]

You can use the **Remote Spooling Communications Subsystem (RSCS)** to communicate:

- **[S370]** From your first-level VM/ESA 2.4.0 system to your second-level System/370 system
- **[ESA]** From your first-level old ESA system to your second-level VM/ESA 2.4.0 system

This allows you to transfer files such as flat files, data files, and dumps. To set up RSCS, you make a virtual channel and add the proper definitions in the configuration and control files.

The RSCS networking licensed product provides a store-and-forward file transfer capability to VM/ESA systems. RSCS can also be used to transfer files to and from MVS, OS/400®, and VSE systems.

RSCS uses binary synchronous communication (BSC) or channel-to-channel adapter (CTCA) connections to transfer data on non-SNA links. 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, multi-tasking services, and general I/O services.

Figure 14 on page 126 shows the relationship between first-level and second-level RSCS machines in a System/370 conversion. Figure 15 on page 126 shows the relationship between first-level and second-level RSCS machines in an ESA conversion.

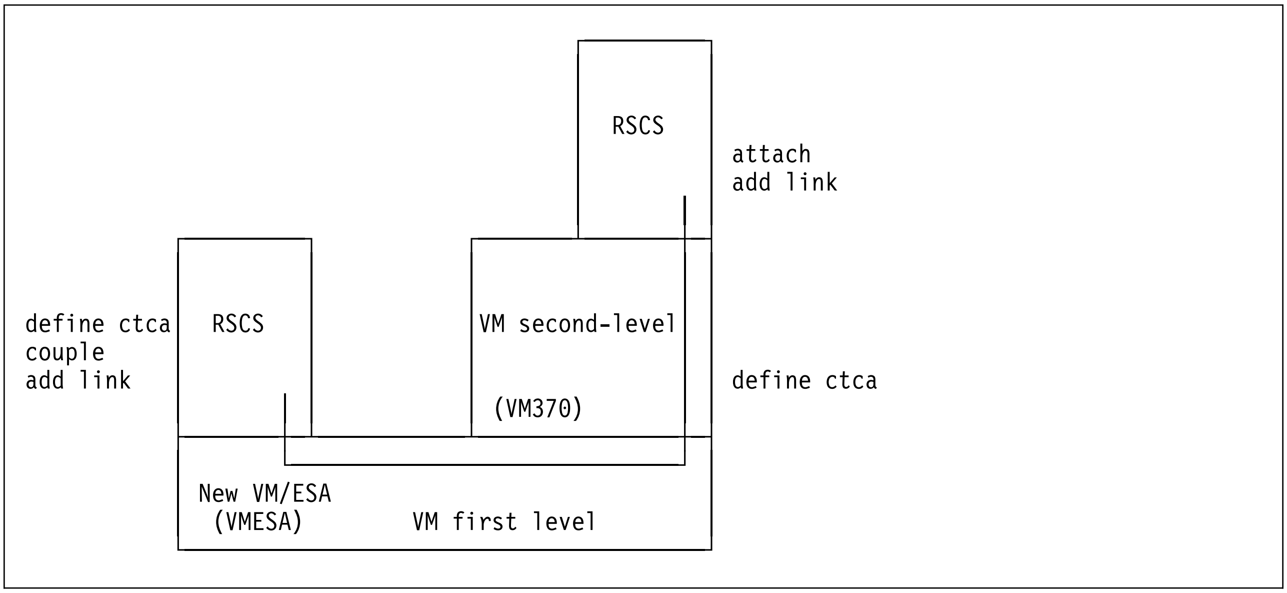


Figure 14. Connectivity between First- and Second-Level RSCS Machines in a System/370 Conversion

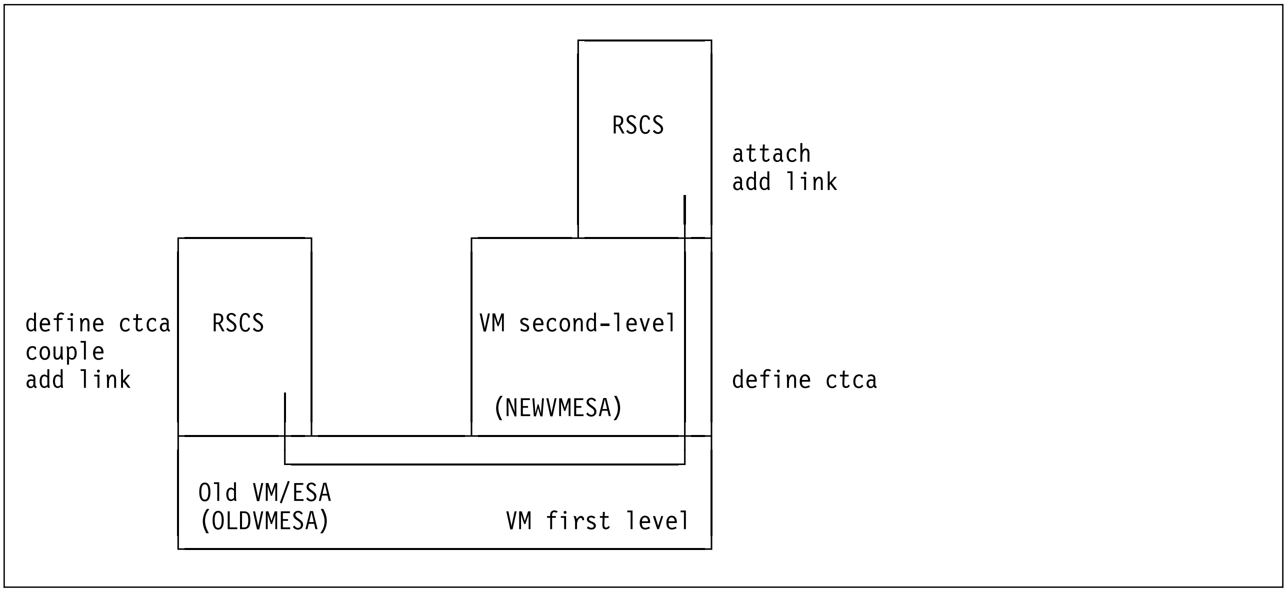


Figure 15. Connectivity between First- and Second-Level RSCS Machines in an ESA Conversion

Steps to Follow for First-to-Second-Level System Connectivity [ALL]

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:

   - **[S370]** VM370
   - **[ESA]** 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 VM370 or 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 VM370 or 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:

   - **[S370]** DMKRIO
   - **[ESA]** SYSTEM CONFIG file or HCPRIO

   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 VM370 or 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:

   - **[S370]:**

     ```
     couple 312 vm370 312
     ```

   - **[ESA]:**

     ```
     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 VM370 or NEWVMESA. For example, enter:

   ```
   attach 312 rscs 312
   ```

Samples [ALL]

Because you must perform the preceding steps each time you start your RSCS virtual machines, you might want to automate the procedure.

In the following samples, 312 is the address of the link, and the user ID that runs the second-level system is:

- **[S370]** VM370
- **[ESA]** NEWVMESA

Sample Exec for Setting Up RSCS Connectivity in the User ID That Will Run the Second-level System: From the VM370 or NEWVMESA user ID, issue the DEFRSCS exec to activate the link active on the first-level side.

```
/*** DEFRSCS EXEC ***/
address command
'CP DEFINE CTCA 312'
```

Set Up the First-level RSCS's PROFILE GCS: To automate the link from the first-level RSCS machine's side, add the following commands in RSCS's PROFILE GCS file:

- **[S370]:**

  ```
  /* First-level RSCS's PROFILE GCS */
  'CP DEFINE CTCA 312'
  'CP COUPLE 312 VM370 312'
  ```

- **[ESA]:**

  ```
  /* First-level RSCS's PROFILE GCS */
  'CP DEFINE CTCA 312'
  'CP COUPLE 312 NEWVMESA 312'
  ```

Set Up the First-level RSCS CONFIG Files: Modify the RSCS CONFIG files as follows:

- **[S370]:**

  ```
  'LINK VM370 NJE 312 ...   /* Add this to the systemid CONFIG file.*/
                            /* Use parameters for virtual channel.  */
  'PARM VM370 TA=1 ST=2 ... /* Add this to the systemid CONFIG file.*/
                            /* Use parameters for virtual channel.  */

  'ROUTE VM370 ...          /* Add ROUTE statement to  CONFIG file. */
                            /* Tailorable parms are not shown here. */

  'RSCS START VM370'        /* This can be added to the PROFILE GCS */
                            /* or can be issued from RSCS for the   */
                            /* current session only.                */
  ```

- **[ESA]:**

  ```
  'LINK NEWVMESA NJE 312 ...   /* Add this to the systemid CONFIG file.*/
                               /* Use parameters for virtual channel.  */
  'PARM NEWVMESA TA=1 ST=2 ... /* Add this to the systemid CONFIG file.*/
                               /* Use parameters for virtual channel.  */

  'ROUTE NEWVMESA ...          /* Add ROUTE statement to  CONFIG file. */
                               /* Tailorable parms are not shown here. */

  'RSCS START NEWVMESA'        /* This can be added to the PROFILE GCS */
                               /* or can be issued from RSCS for the   */
                               /* current session only.                */
  ```

Set Up the Second-level RSCS's PROFILE GCS: The second-level RSCS machine must be configured in the same manner as the first-level machine. Modify the RSCS PROFILE GCS profile as follows:

```
'CP ATTACH 312 * 312'
```

Set Up for Second-level RSCS CONFIG Files: Add the first-level system identification to the second-level system. Modify the RSCS CONFIG files as follows:

- **[S370]:**

  ```
  'LINK VMESA NJE 312 ...     /* Add this to the systemid CONFIG file.*/
                              /* Use parameters for virtual channel.  */
  'PARM VMESA TA=1 ST=2 ...   /* Add this to the systemid CONFIG file */
                              /* Use parameters for virtual channel   */

  'ROUTE VMESA ...            /* Add ROUTE statement to  CONFIG file  */
                              /* tailorable parms are not shown here  */
  ```

```
'RSCS START VMESA'          /* This can be added to the PROFILE GCS */
                            /* or can be issued from RSCS for the   */
                            /* current session only.                */
```

- **[ESA]:**

```
'LINK OLDVMESA NJE 312 ...    /* Add this to the systemid CONFIG file.*/
                              /* Use parameters for virtual channel.  */
'PARM OLDVMESA TA=1 ST=2 ...  /* Add this to the systemid CONFIG file */
                              /* Use parameters for virtual channel   */

'ROUTE OLDVMESA ...           /* Add ROUTE statement to  CONFIG file  */
                              /* tailorable parms are not shown here  */

'RSCS START OLDVMESA'         /* This can be added to the PROFILE GCS */
                              /* or can be issued from RSCS for the   */
                              /* current session only.                */
```

Set Up the SYSTEM NETID File: Edit the SYSTEM NETID file on the system 190 disk to ensure the processor ID and system identification corresponds to the identification that you have specified for communications for both the first and second-level systems.

PVM [ALL]

The **VM/Pass-Through Facility (PVM)** licensed program provides virtual terminal capabilities for VM/ESA. PVM allows VM/ESA users to log on to a non-VM system, such as MVS or VSE, through a dedicated PVM link that emulates a 3174 or 3274 control unit. PVM supports BSC, CTC (370), ESCON CTC, and 3088 for VM-to-VM communications. When VTAM is installed on your VM system, PVM can use APPC/VM to communicate over an SNA LU 6.2 network or can allow VM/ESA users to log on to SNA applications through the SNA LU 2 gateway.

PVM can be useful for allowing the system programmer, system administrator, or general user to move between the old system and the new system using the DIAL command; for example:

```
DIAL PVM
```

Figure 16 on page 130 shows the relationship between first-level and second-level PVM machines in a System/370 conversion. Figure 17 on page 130 shows the relationship between first-level and second-level PVM machines in an ESA conversion.

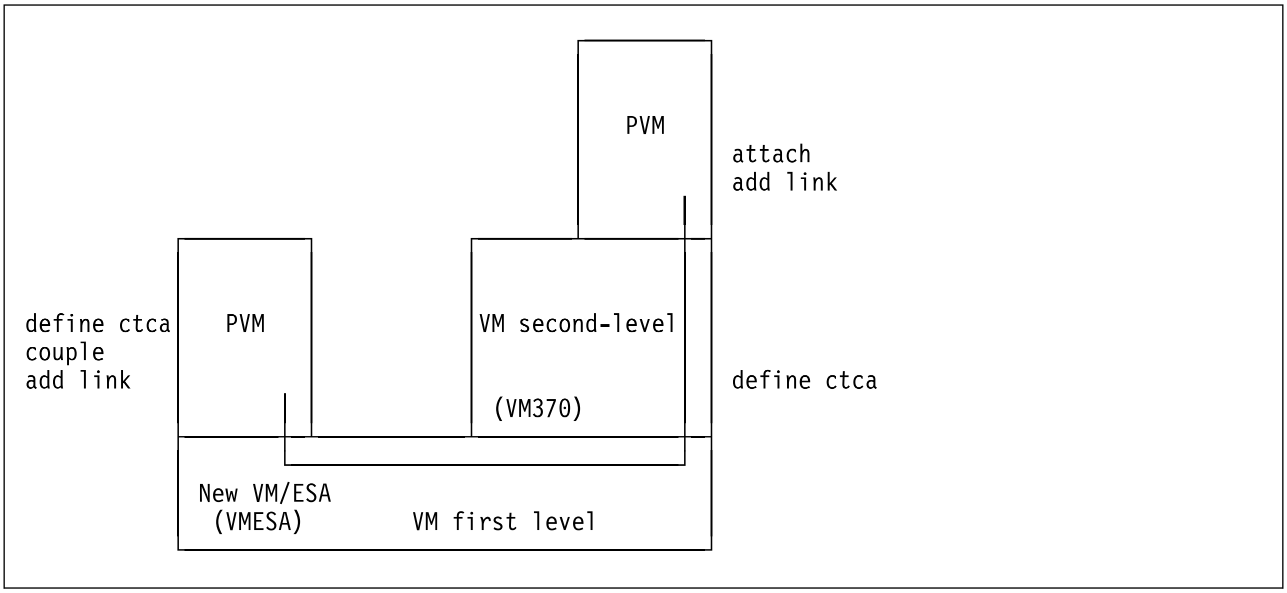


Figure 16. Connectivity between First- and Second-Level PVM Machines in a System/370 Conversion

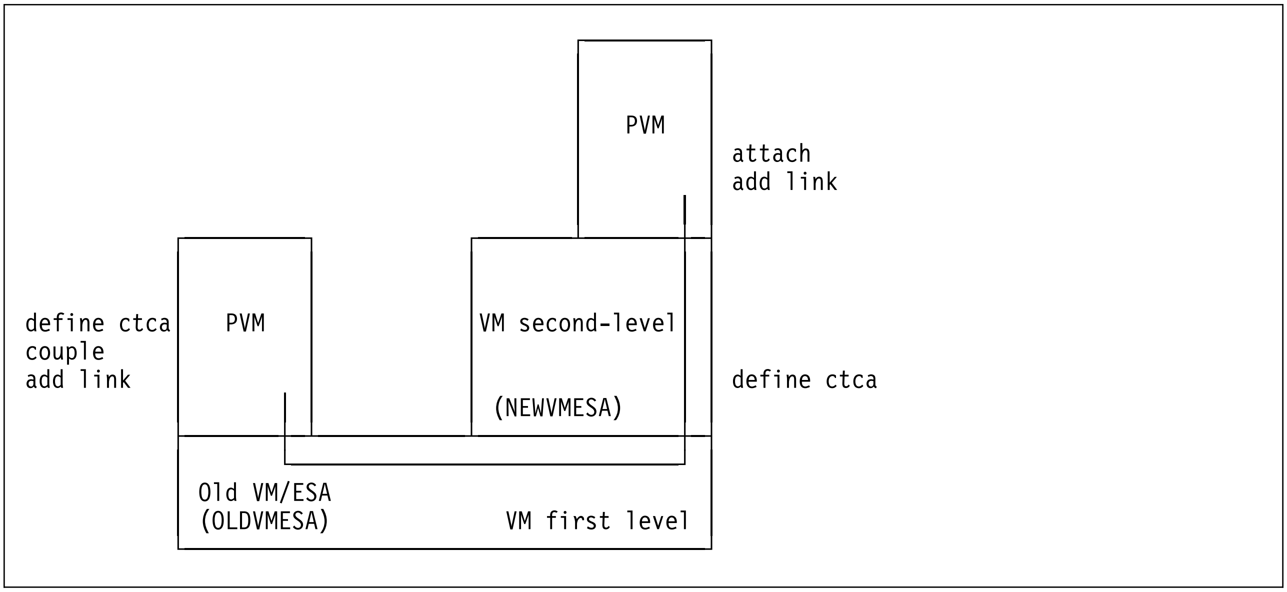


Figure 17. Connectivity between First- and Second-Level PVM Machines in an ESA Conversion

Steps to Follow for First-to-Second-Level System Connectivity [ALL]

On the First-level System:

1. Define a link between the first-level PVM 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:

   - **[S370]** VM370
   - **[ESA]** NEWVMESA

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

a. From **both** the first-level PVM virtual machine and the VM370 or 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 of the VM370 or NEWVMESA user ID and the PVM 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:

   - **[S370]** DMKRIO
   - **[ESA]** SYSTEM CONFIG file or HCPRIO

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

2. From either the PVM machine or the VM370 or NEWVMESA user ID, couple the addressed links to enable communications over the virtual channel. For example, from the first-level PVM virtual machine, you can enter:

   - **[S370]:**

     ```
     couple 312 vm370 312
     ```

   - **[ESA]:**

     ```
     couple 312 newvmesa 312
     ```

3. Bring up the second-level system, including the second-level PVM machine.

On the Second-level System:

1. From the second-level PVM virtual machine, attach the channel address defined by VM370 or NEWVMESA. For example, enter:

   ```
   attach 312 pvm 312
   ```

Samples [ALL]

Because you must perform the preceding steps each time you start the PVM virtual machines, you may want to automate the procedure.

In the following examples, 312 is the address of the link, and the user ID that runs the second-level system is:

- **[S370]** VM370
- **[ESA]** NEWVMESA

Sample Exec for Setting Up PVM Connectivity in the User ID That Will Run the Second-level System: From the VM370 or NEWVMESA user ID, issue the DEFPVM exec to activate the link on the first-level side:

```
/*** DEFPVM EXEC ***/
address command
'CP DEFINE CTCA 312'
```

Set Up the First-level PVM's PROFILE EXEC: To automatically activate the link from the first-level PVM virtual machine's side, you can put the following in PVM's PROFILE EXEC file:

- **[S370]:**

  ```
  /* First-level PVM's PROFILE EXEC */
  'CP DEFINE CTCA 312'
  'CP COUPLE 312 VM370 312'
  ```

- **[ESA]:**

  ```
  /* First-level PVM's PROFILE EXEC */
  'CP DEFINE CTCA 312'
  'CP COUPLE 312 NEWVMESA 312'
  ```

Set Up the First-level PVM's CONFIG File: Modify the PVM CONFIG file as follows:

- **[S370]:**

  ```
  'LINK 312 VM370 CTCA'     /* Add this to the PVM CONFIG file.     */
                            /* Use parameters for virtual channel.  */
  'ROUTE VM370 ...          /* Add ROUTE to PVM CONFIG file.        */
                            /* Tailorable parms are not shown here. */
  ```

- **[ESA]:**

  ```
  'LINK 312 NEWVMESA CTCA'     /* Add this to the PVM CONFIG file.     */
                               /* Use parameters for virtual channel.  */
  'ROUTE NEWVMESA ...          /* Add ROUTE to PVM CONFIG file.        */
                               /* Tailorable parms are not shown here. */
  ```

Set Up the First-level PVM's PROFILE PVM: Modify the PROFILE PVM file as follows:

```
'START LINE 312'             /* This can be added to the PROFILE PVM */
                             /* or can be issued from PVM for the    */
                             /* current session only.                */
```

Set Up the Second-level PVM's PROFILE EXEC: The second-level PVM machine must be configured in the same as the first-level machine. Add the following to the second-level PVM's PROFILE EXEC:

```
'CP ATTACH 312 * 312'
```

Set Up the Second-level PVM's CONFIG File: Modify the CONFIG file on the second-level PVM virtual machine as follows:

- **[S370]:**

  ```
  'LINK 312 VMESA CTCA'      /* Add this to the PVM CONFIG file.     */
                             /* use parameters for virtual channel   */
  'ROUTE VMESA ...           /* Add ROUTE to PVM CONFIG file.        */
                             /* Tailorable parms are not shown here. */
  ```

- **[ESA]:**

  ```
  'LINK 312 OLDVMESA CTCA'     /* Add this to the PVM CONFIG file.     */
                               /* use parameters for virtual channel   */
  'ROUTE OLDVMESA ...          /* Add ROUTE to PVM CONFIG file.        */
                               /* Tailorable parms are not shown here. */
  ```

Set Up the Second-level PVM's PROFILE PVM File: Modify the second-level PVM's PROFILE PVM file as follows:

```
'START LINE 312'          /* This can be added to the PROFILE PVM */
                          /* or can be issued from PVM for the    */
                          /* current session only.                */
```

How to Back Out [ALL]

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

Note: To back out spool files, use the SPXTAPE command on VM/ESA 1.2.2 or later. On releases prior to VM/ESA 1.2.2, use the SPTAPE command. SPTAPE and SPXTAPE are not compatible. SPTAPE cannot load or scan files dumped by SPXTAPE.

Also, refer to the CP Command compatibility table for your old release to review possible compatibility problems between the old and new versions of SPTAPE.

Backing Out Because of System Problems [S370]

When converting from System/370, if you run into a problem with VM/ESA 2.4.0 that forces you to back out, and you have not yet exploited functions specific to VM/ESA 2.4.0 or converted CMS users, the backout will not affect your end users. If, on the other hand, you run into a problem and have to back out after you have converted CMS users, you will greatly affect these users. Once users convert their applications to run on the VM/ESA 2.4.0 CMS, they may not be able to run these applications on the old CMS. Therefore, you should not convert users to VM/ESA 2.4.0 until you are satisfied with its stability.

If you expect certain users to have large numbers of spool files, you should set the spool file limit for them by using the SPOOLFILE control statement in the user directory so that a limit of 9900 is not exceeded.

```
SPOOLFILE MAXSPOOL 9900
```

VM/ESA 2.4.0 supports a limit of 9999 spool files per user; VM/ESA 1.1.5 370 Feature has a limit of 9900 spool files per system.

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

Because VM/ESA 2.4.0 stores dumps in system data files, you cannot transport them from your VM/ESA 2.4.0 system to your old system. If you have a problem and want to back out to your old system, you need the stand-alone dump facility to examine the problem.

If you must back out from VM/ESA 2.4.0 and IPL an old system, do the following:

Note: When you reformat cylinder 0 or pages 0-3 in the following steps, you lose any information in cylinder 0 or pages 0-3, such as pointers to the nucleus and the user directory. Note also that VM/ESA 2.4.0 reserves pages 0-3 of FBA DASD for system use.

1. Dump the spool files with the SPXTAPE DUMP command (VM/ESA 1.2.2 or later) or the SPTAPE DUMP command.
2. If you have not maintained your system CP-owned DASD, reformat cylinder 0 or pages 0-3 and reallocate enough CP-owned DASD to bring up an old system.

   If you are using 370-formatted FBA DASD during conversion, you do not have to reformat and reallocate for a backout to a System/370 system.

   The allocation maps of a System/370 system are different from those of VM/ESA 2.4.0. Refer to “Temporary Use of 370 DASD Allocations for VM/ESA 2.4.0 CP-Owned Volumes [1.1.5]” on page 232 for a comparison of allocation options.

   Because you cannot reformat and reallocate paging and spooling packs to which you are currently attached, you must:

   - Shut down the system
   - Re-IPL it with the CP-owned DASD (that you plan to reformat) offline
   - Attach the DASD to a virtual machine
   - Reformat cylinder 0 or pages 0-3 and reallocate.

   To save time, you may want to write an exec that reformats cylinder 0 or pages 0-3 and reallocates the CP-owned DASDs. Another method is to keep copies of the two different cylinder 0 or pages 0-3 formats in free cylinders or pages on the device. Then you can use stand-alone DDR to copy the appropriate cylinder or pages to cylinder 0 or pages 0-3, depending on which way you are switching.
3. If you have not maintained a separate user directory on the old system residence pack, redirect the user directory using the DIRECT command so it can be used on the old system.
4. Shut down the VM/ESA 2.4.0 system.
5. Re-IML the processor to System/370 mode.
6. IPL the old system.
7. If necessary, reformat cylinder 0 or pages 0-3 and reallocate the remainder of your old system CP-owned DASD. Either re-IPL or physically attach the DASD to the system.
8. Load the spool files with the SPXTAPE LOAD or SPTAPE LOAD command (depending on whether you used SPXTAPE or SPTAPE to dump the files).
9. Enable the terminals.

Reformatting Cylinder 0 or Pages 0-3 and Reallocating CP-Owned DASD Considerations [S370]

If you use the same CP-owned DASD for your VM/ESA 2.4.0 system and your old system, and you want to either convert to VM/ESA 2.4.0 from the old system or back out from VM/ESA 2.4.0 to the old system, you must:

- Reformat cylinder 0 or pages 0-3.

  Note: When you reformat cylinder 0 or pages 0-3, you lose any information in cylinder 0 or pages 0-3, such as pointers to the nucleus and the user directory. Note also that VM/ESA 2.4.0 reserves pages 0-3 of FBA DASD for system use.

- Reallocate the CP-owned DASD.

You can do this by using the Device Support Facilities (ICKDSF) program, Release 11 or later for CKD or ECKD™ DASD and Release 14 or later for FBA DASD. Use the MODE(ESA or XA) parameter.

If you are used to using a format/allocate program, you can use CPFMTXA rather than ICKDSF. If you are moving from your old system to your VM/ESA 2.4.0 system, load the CPFMTXA module from your VM/ESA 2.4.0 Initial Installation System onto your old system. VM/ESA 2.4.0's CPFMTXA uses ICKDSF "under the covers" to do the actual formatting and allocating.

A DASD formatted by a System/370 system's format/allocate program (DMKFMT) works under VM/ESA 2.4.0 only if you do not make the volume CP-owned to VM/ESA 2.4.0.

Note: You do not need to reformat or reallocate if the DASD is:

- CP-owned by either the old system or VM/ESA 2.4.0, but not both
- Not CP-owned by either system
- A 370-formatted DASD

What To Do If You Backed Out Without Reformatting and Reallocating: If you backed out from your VM/ESA 2.4.0 system and IPLed your old system without reformatting cylinder 0 or pages 0-3 or reallocating the DASD, the old system does not issue an error message informing you that the DASD was incorrectly formatted or allocated. Instead, the system comes up with the DASD offline. If you try to attach the DASD to your system using the ATTACH command, the DASD is attached, *but it is not CP-owned*. You need to reformat cylinder 0 or pages 0-3 and reallocate the DASD. Then you can successfully attach the DASD to the system and it will be CP-owned.

User Directory Concerns [S370]

Before backing out, enter the DIRECT command to bring the old system's user directory online. If you do not do this, you have to use the old system's stand-alone directory function in order to IPL the old system.

Backing Out Because of System Problems [ESA]

How you back out depends on whether you are using:

1. Separate residence volumes for the old system and the new system
2. The same residence volume for both the old and new systems

Using Separate Residence Volumes [ESA]

If you run into a problem with VM/ESA 2.4.0 that forces you to back out, and you have not yet exploited functions specific to VM/ESA 2.4.0 or converted CMS users, the backout will not affect your end users. If, on the other hand, you run into a problem and have to back out after you have converted CMS users, you will greatly affect these users. Once users convert their applications to run on the VM/ESA 2.4.0 CMS, they may not be able to run these applications on the old CMS. Therefore, you should not convert users to VM/ESA 2.4.0 until you are satisfied with its stability.

Throughout the conversion effort, 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 VM/ESA 2.4.0 and IPL an old system, do the following:

1. Dump the spool files with the SPXTAPE DUMP command (VM/ESA 1.2.2 or later) or the SPTAPE DUMP command.
2. Shut down the VM/ESA 2.4.0 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 or SPTAPE LOAD command (depending on which command you used to dump the files).
6. Enable the terminals.

Using the Same Residence Volume [ESA]

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 (on VM/ESA 1.2.2 or later) or the SPTAPE 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 or the SPTAPE LOAD command (depending on which command you used to dump the files) to restore any saved spool files.

Spool File Concerns [S370]

When converting from a System/370 system, you must use the SPTAPE DUMP command to dump the spool files so you can restore them to your old system. Also see “Spool File Limit Considerations [S370]” on page 89.

Second-Level Systems Running on VM/ESA 2.4.0 [S370]

If you are running your old system on top of VM/ESA 2.4.0 and you have to back out, you might want to bring up your old system first-level. Before you decide to do this, remember that not all processors support native System/370 processor mode. For processor information, see the VM/ESA Processor Support Matrix in the *VM/ESA: General Information* book or on the IBM VM operating system home page (http://www.ibm.com/s390/vm).

Once you decide to bring up your System/370 system first-level:

- Make sure your processor is running in 370 mode.
- Select an IOCP that you used with your System/370 system. Remember that this IOCP may have to define an LPAR if you have a 9121, 9021, or 3090™.

- IPL the volume that holds your System/370 system.

Note that once you bring up your System/370 system first-level, you only have access to the spool files from your old system. You can move spool files from your VM/ESA 2.4.0 system to the old system, but the files will not be identical to the original spool file. See “Spool File Conversion Considerations and Procedures [S370]” on page 88 for more information on spool files.

Backing Out Individual Users Because of Problems with Application Programs [ALL]

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

Preventing Users from Converting Their Applications [S370]

An easy way to temporarily prevent users from converting their applications running on XA or XC virtual machines is to use the user class restructure capability to make the SET MACHINE command a privileged command. Then general users cannot set their virtual machines to run as XA or XC virtual machines. When you are satisfied that the system is stable, you can return SET MACHINE to a privilege class G command.

For example, to set the user class of SET MACHINE to privilege class A, B, C, and D:

1. Put the following entry in your class override file:

   ```
   SET MACHINE  NEWCLASS=ABCD  OLDCLASS=G
   ```

2. Verify the control statements in the class override file have the correct syntax:

   ```
   override class override a (validate
   ```

3. Apply the new privilege class for SET MACHINE:

   ```
   override class override a
   ```

To return SET MACHINE to privilege class G:

1. Remove the SET MACHINE entry in your override file.
2. Verify the control statements in the class override file have the correct syntax:

   ```
   override class override a (validate
   ```

3. Remove the privilege class for SET MACHINE:

   ```
   override class override a
   ```

See *VM/ESA: Planning and Administration* for more information about user class restructure and *VM/ESA: CP Command and Utility Reference* for information about the SET MACHINE command.

User Directory Concerns [ALL]

If a user or group of users has problems running applications on VM/ESA 2.4.0 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 VM/ESA 2.4.0 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.

Moving from Your Old System to VM/ESA 2.4.0 [S370]

When you are ready to go back to the VM/ESA 2.4.0 system, use the stand-alone Device Support Facilities program (ICKDSF) to reformat cylinder 0 or pages 0-3 of your volume and reallocate DASD to VM/ESA 2.4.0 format. Use ICKDSF Release 11 or later for CKD or ECKD DASD and ICKDSF Release 14 or later for FBA DASD.

Note: When you reformat cylinder 0 or pages 0-3, you lose any information in cylinder 0 or pages 0-3, such as pointers to the nucleus and the user directory. Note also that VM/ESA 2.4.0 reserves pages 0-3 of FBA DASD for system use.

Installing a Backlevel CMS [ALL]

IBM provides limited support for multiple levels of CMS on VM/ESA 2.4.0. You can continue to use your production CMS with the CP component of VM/ESA 2.4.0, and then gradually convert 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 VM/ESA 2.4.0.

The CMS component from VM/ESA 1.1.5 370 Feature is supported on VM/ESA 2.4.0 only in a 370 virtual machine.

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.

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 a small programming enhancement or the APAR service stream. An exception to this nonsupport for new function is the FBA support in VM/ESA 2.4.0, which is supported on a backlevel CMS from VM/ESA 1.1.5 370 Feature.

Note: Throughout this discussion, backlevel CMS refers to an older level of CMS, still in service, running on VM/ESA 2.4.0 CP.

The following example shows how to bring up a backlevel CMS on VM/ESA 2.4.0 CP.

Note: VM/ESA 2.4.0 CMS does not support back levels of CP.

Before You Install a Backlevel CMS [ALL]

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 the MDISK statement that defines the 190 minidisk on your old system in Table 15 on page 140. This can be used as a reference when defining the minidisk to hold the backlevel CMS on VM/ESA 2.4.0.
- Record the number of cylinders needed for your old 190 disk in Table 15 on page 140. 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 VM/ESA 2.4.0 system. While installing a backlevel CMS, you will have to copy the contents of your 190 minidisk from the old system to your VM/ESA 2.4.0 system.
  - If you use the same type of DASD to hold the backlevel CMS on your VM/ESA 2.4.0 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 VM/ESA 2.4.0 in Table 15 on page 140.

- If you are using the VMFPLC2 command to copy files, you have to format the VM/ESA 2.4.0 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/O     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/O   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 *VM/ESA: 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 pages
For 3390-1:	66 * 180 = 11880 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 pages
For 3390-1:	6 * 180 = 1080 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 15.

Table 15. 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 Backlevel CMS [ALL]

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

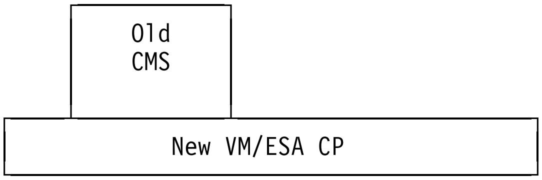


Figure 18. Backlevel CMS Running on a VM/ESA 2.4.0 CP

The steps for installing the old CMS on the VM/ESA 2.4.0 CP are as follows:

1. Log on to your VM/ESA 2.4.0 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 15 on page 140.

   The MDISK directory control statement in your VM/ESA 2.4.0 user directory for the backlevel CMS may look like this:

   ```
   MDISK 590 3380 125 72 240RES 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 15 on page 140.

     – If you are using the DDR method later in the procedure, you must make your backlevel CMS minidisk on VM/ESA 2.4.0 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 VM/ESA 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 *VM/ESA: 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 *VM/ESA: 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. **[ESA]** 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 VM/ESA 2.4.0, skip to step 10 on page 143.
   >
   > 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 VM/ESA 2.4.0 590. The following steps show how to DDR the old 190 contents to the VM/ESA 2.4.0 590 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 VM/ESA 2.4.0 MAINT user ID.

   e. Attach the tape to the MAINT user ID on the VM/ESA 2.4.0 system:

      ```
      attach 494 to maint as 181
      ```

   f. Load the files off the tape onto the VM/ESA 2.4.0 590 minidisk:

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

   Now you have the old VM system 190 CMS files on your VM/ESA 2.4.0 590 minidisk.

8. Define the 190 minidisk that contains VM/ESA 2.4.0 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 18 on page 145.

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

10. While you are still on your VM/ESA 2.4.0 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 15 on page 140.) 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 VM/ESA 2.4.0 590. The following steps show how to move the old-system 190 contents to the VM/ESA 2.4.0 590 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 VM/ESA 2.4.0 MAINT user ID.

    h. Attach the tape to the MAINT user ID on your VM/ESA 2.4.0 system:

```
attach 494 to maint as 181
```

i. Access the VM/ESA 2.4.0 590 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 VM/ESA 2.4.0 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 VM/ESA 2.4.0 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. **[S370]** Set MAINT's virtual machine to 370 mode:

```
set machine 370
```

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

```
ipl 00c cl
```

17. 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.

18. Determine the DEFSYS command to use to define CMSOLD as a named saved system:

    - **[S370]:**

      To figure out what your DEFSYS command should be, you can convert the entry in the DMKSNT for your System/370 CMS. "Converting DMKSNT ASSEMBLE Macros or SNT OVERRIDE Tags to VM/ESA Commands [S370]" on page 84 shows an example of converting a DMKSNT entry into a DEFSYS command.

    - **[ESA]:**

      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 19 on page 146 defines various CMS named saved systems.

19. Issue the DEFSYS command or use the sample exec in Figure 19 on page 146 to define the CMSOLD named saved system; for example:

    ```
    defnss cmsold
    ```

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

21. **[S370]** Set MAINT's virtual machine to 370 mode:

    ```
    set machine 370
    ```

22. IPL the 190 minidisk with CMSOLD on it:

    ```
    ipl 190 parm savesys cmsold
    ```

Defining Various Levels of CMS [ALL]

Use the following sample exec to define different levels of CMS in a named saved system. Note that **three versions are shown** of using the DEFSYS command to define an old CMS: one for a conversion from VM/ESA 1.1.5 370 Feature; one for a migration from VM/ESA 1.2.0, VM/ESA 1.2.1, or VM/ESA 1.2.2; and one for a migration from VM/ESA 2.1.0 or later.

```
/********************************************************************/
/*                                                                  */
/* 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"
```

1.1.5

```
/* DEFSYS command to define an old CMS */
nss.2  ="CP DEFSYS CMSOLD   0-8 EW E-22 EW EF0-10FF SR MINSIZE=256K ",
          "MACHMODE 370"
```

1.2.0, 1.2.1, 1.2.2

```
/* DEFSYS command to define an old CMS */
nss.2  ="CP DEFSYS CMSOLD   0-D EW 20-23 EW E00-10FF SR MINSIZE=256K ",
          "MACHMODE 370,XA,XC"
```

2.1.0 and later

```
/* 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
```

Figure 19 (Part 1 of 2). DEFNSS EXEC

```
/* 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 19 (Part 2 of 2). DEFNSS EXEC

Swapping Between the Backlevel CMS and VM/ESA 2.4.0 CMS [ALL]

If you want to be able to switch between the VM/ESA 2.4.0 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 20. SWAP2NEW EXEC

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

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

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

Figure 21. SWAP2OLD EXEC

The user should then:

- **[S370]** Issue SET MACH 370 and IPL CMSOLD.
- **[ESA]** Issue SET MACH XA and IPL CMSOLD.

Chapter 8. Overview of Compatibility between VM/ESA 2.4.0 and Previous Releases

This chapter provides a high-level overview of how compatible VM/ESA 2.4.0 is with previous releases.

This chapter may contain information for any or all of the conversions documented in this book. For definitions of the conversion identifiers used in this chapter, see "How to Identify the Topics That Apply to Your Conversion" on page 3.

This chapter contains the following sections:

- "Year 2000 Compatibility [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]"
- "Compatibility with VM/ESA 1.1.5 370 Feature [1.1.5]" on page 150
- "Compatibility with VM/ESA 1.2.0 or Later [ESA]" on page 157

Year 2000 Compatibility [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

VM/ESA 2.4.0 is a Year-2000-ready product. The primary objective of VM/ESA's Year 2000 support is to ensure that the VM/ESA operating system continues to operate correctly in the year 2000 and beyond, as well as providing functions to allow applications to do the same.

VM/ESA's Year 2000 support includes:

- Commands that accept and display date formats that include 4-digit years
- Application program interfaces (APIs) that support date formats that include 4-digit years
- Resolution of dates with 2-digit years to correct 4-digit-year forms
- Tools to help you do date format conversions

The capability to define system-wide and user default date formats provides flexibility for installations and users for all VM/ESA commands that support multiple date formats.

VM/ESA supports running guest systems with different system dates and times from the host VM/ESA system. This enables testing of OS/390®, MVS, TPF, VM/ESA, and VSE systems with Year 2000 dates without affecting the rest of your VM/ESA host system.

You can find more information about VM/ESA's Year 2000 approach at VM/ESA's Year 2000 Internet site:

http://www.ibm.com/s390/vm/year2000/

For more information in this book about changes to VM/ESA for Year 2000 support, see:

- New Functions - Planning:
  - "Year 2000 Support [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]" on page 175
  - "Year 2000 Support, Phase 2 [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]" on page 178

- New Functions - Application Programming:
  - “Year 2000 API Support [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]” on page 201
- Administration changes and conversion considerations:
  - “ISO Date Used in Default CMS IPL Heading [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]” on page 254
- Virtual machine operation changes and conversion considerations:
  - “Year 2000 Considerations [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]” on page 287
- Application programming changes and conversion considerations:
  - “Calling CSL Routines That Have Dates as Output [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]” on page 301
- Diagnosis changes and conversion considerations:
  - “FST and ADT Macro Changes to Support 4-digit Years [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]” on page 311

Compatibility with VM/ESA 1.1.5 370 Feature [1.1.5]

Although VM/ESA 2.4.0 is generally compatible with VM/ESA 1.1.5 370 Feature, there are enough differences caused by the different architecture, new functions, and usability improvements that the conversion from VM/ESA 1.1.5 370 Feature to VM/ESA 2.4.0 is more difficult than a release-to-release conversion.

Comparison of Component Structure [1.1.5]

Table 16 compares the components in VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0.

Table 16. Comparison of Components in VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0

VM/ESA 1.1.5 370 Feature	**VM/ESA 2.4.0**
CP370	CP
CMS	CMS
GCS370	GCS
IPCS	Dump Viewing Facility
Procedures Language VM/REXX	REXX/VM
TSAF	TSAF
AVS	AVS
VMSES/E	VMSES/E

VM/ESA 2.4.0 also supports:

- DFSMS/VM®, which simplifies data storage management
- OpenEdition® for VM/ESA, which provides support for POSIX, sockets, and the Distributed Computing Environment (DCE)
- VM/ESA Graphical User Interface (GUI) Facility
- S/390® Open Systems Adapter Support Facility (OSA/SF) for VM/ESA

Compatibility by Component [1.1.5]

CP: Many VM/ESA 2.4.0 CP external interfaces are incompatible with CP370, mostly because of architectural differences.

IPCS and Dump Viewing Facility: VM/ESA 2.4.0 provides the Dump Viewing Facility component instead of IPCS. There are differences in the naming of dumps, in symptom record handling for non-CP dumps, in processing trace table data, and in display-management capability. Also, the Dump Viewing Facility does not support the problem-recording and management functions of IPCS.

CMS: The CMS level for VM/ESA 2.4.0 is CMS Level 15. CMS Level 15 runs in 370-XA (XA virtual machine) and ESA/XC (XC virtual machine) architectures. Only CMS levels prior to CMS Level 12 can run in a 370 virtual machine. Most CMS external interfaces are compatible.

REXX/VM: VM/ESA 2.4.0 REXX external interfaces are upwardly compatible with VM/ESA 1.1.5 370 Feature REXX. In VM/ESA 1.1.5 370 Feature, REXX is formally called 'Procedures Language VM/REXX.' In VM/ESA 2.4.0, it is called 'REXX/VM.' VM/ESA 2.4.0 REXX runs in 370, XA, and XC virtual machines and has been enhanced to exploit 31-bit addressing.

REXX/VM includes the full REXX Level 2 support that is available on other SAA platforms.

TSAF: TSAF lets users connect to and communicate with other APPC/VM applications. TSAF is supported in a 370 virtual machine. In VM/ESA 2.4.0, TSAF runs in an XA virtual machine with the 370ACCOM feature enabled. APPC/VM is supported in 370, XA, ESA, and XC virtual machines.

AVS: AVS lets programs connect to and communicate with APPC programs in an SNA network. AVS is supported in 370, XA, ESA, and XC virtual machines.

VMSES/E: Most VMSES/E commands are upwardly compatible.

CP Configuration Differences [1.1.5]

CP system configuration is changed for VM/ESA 2.4.0. DMKSYS and DMKRIO files cannot be used at all in VM/ESA 2.4.0. They are replaced, for the most part, by the new system configuration files. You must have HCPRIO and HCPSYS files with the following entries:

System File	Minimum Entry
HCPSYS	SYSEND
HCPRIO	RIOGEN CONS=DYNAMIC

HCPSYS and HCPRIO files with these entries are shipped on the VM/ESA System DDR tapes. They are included in the CP loadlist and should not be removed.

Note: HCPRIO entries are also needed for dedicated devices of V=R guests that use V=R recovery.

CP Nucleus Change

The CP nucleus can now also be stored as a module. See “CP Nucleus Enhancements [1.1.5]” on page 162 for a summary of the changes.

Storage Differences [1.1.5]

Real Storage Size: VM/ESA 1.1.5 370 Feature supports a maximum real storage size of 64MB. VM/ESA 2.4.0 supports the architected real storage size of ESA/370 and ESA/390, which is 2GB.

Virtual Storage Addressing: VM/ESA 1.1.5 370 Feature supports 24-bit virtual storage addressing only and supports virtual machine storage sizes up to 16MB. VM/ESA 2.4.0 supports 24-bit or 31-bit virtual storage addressing and supports virtual machine storage sizes up to 2047MB, providing almost 128 times the addressing capability of VM/ESA 1.1.5 370 Feature. VM/ESA 2.4.0 also supports VM data spaces in XC virtual machines on ESA/390 processors. With data spaces, virtual storage is not limited to 2047MB; it is limited only by the processor.

Shared Storage: Installing user-written applications and licensed programs in shared storage differs. You have to rebuild named saved systems and saved segments into 1MB segments (you can store more than one licensed program in each 1MB segment) and remap saved-segment storage layouts. This is a one-time effort. See “Planning Your System's Saved Segment Layout [S370]” on page 81 for information on how to plan your saved segment layout in VM/ESA 2.4.0.

Saved Segments: In VM/ESA 2.4.0, you do not define saved segments in a system name table (DMKSNT or SNT OVERRIDE) at system generation. Rather, you define saved segments using CP commands. See “Planning Your System's Saved Segment Layout [S370]” on page 81 and “Saved Segments and Named Saved Systems Differences [1.1.5]” on page 254 for details.

Virtual Machine Differences [1.1.5]

Virtual Machine Modes: VM/ESA 1.1.5 370 Feature supports only 370 virtual machines. VM/ESA 2.4.0 supports 370, XA, ESA, and XC virtual machines. However, VM/ESA 2.4.0 CMS is supported only in XA and XC virtual machines. Only CMS levels prior to CMS Level 12 (VM/ESA 2.1.0) can run in a 370 virtual machine. Table 17 lists the VM/ESA 2.4.0 virtual machine modes and for each mode shows the operand used on SET MACHINE command or MACHINE user directory control statement, the architecture simulated, the addressing, and the addressable storage.

Table 17 (Page 1 of 2). Virtual Machine Differences

VM Mode	Operand*	Architecture Simulated	Addressing	Addressable Storage
370	370	System/370 **Note:** CMS Level 12 and later are not supported in a 370 virtual machine.	24-bit	16MB
XA	XA	ESA/370 or ESA/390 **Note:** CMS uses only the 370-XA architecture subset.	31-bit	2047MB
ESA	ESA	ESA/370 or ESA/390	31-bit	2047MB

Table 17 (Page 2 of 2). Virtual Machine Differences

VM Mode	Operand*	Architecture Simulated	Addressing	Addressable Storage
XC	XC	ESA/XC **Note:** VM data spaces are supported in an XC virtual machine.	31-bit	2047MB + VM data spaces**

Note:

* The operand is on the SET MACHINE command or on the MACHINE user directory control statement.
** When using VM data spaces you can address more than 2047MB of virtual machine storage.

Preferred Virtual Machines: VM/ESA 1.1.5 370 Feature supports one preferred guest (V=R). VM/ESA 2.4.0 supports up to six preferred guests (V=R and V=F; there can be only one V=R). Note that the support of multiple preferred guests requires the Processor Resource/Systems Manager (PR/SM) hardware feature running in basic mode.

Virtual Processors: VM/ESA 1.1.5 370 Feature allows up to two virtual processors for the preferred virtual machine only (one dedicated and one virtual that shares the other instruction processor). In VM/ESA 2.4.0, virtual machines can have up to 64 virtual processors, allowing you to test the multiprocessing configurations of guest operating systems. Note also that any virtual machine can control all but one of the real processors of the real machine as dedicated processors. Additional virtual processors can run on the master processor.

Input/Output Differences [1.1.5]

Some devices are no longer supported. See the *VM/ESA: General Information* book for information on device support.

I/O Management: VM/ESA 2.4.0 uses the channel subsystem, which moves the management of I/O from the processor and operating system into the channels themselves. Devices are addressed by any available path and an I/O interrupt does not always return on the same path from which it was started. See "Configuring I/O [1.1.5]" on page 217 for more information.

Real Channel Programs: As in VM/ESA 1.1.5 370 Feature, programs running in 370 V=V virtual machines may build and execute real channel programs, and they use DIAGNOSE code `X'98'` or GCS macros to do this. But in VM/ESA 2.4.0, you must reserve the real storage for those channel programs using DIAGNOSE code `X'98'` by specifying the RIO370 operand of the STORAGE statement in the SYSTEM CONFIG file. (This same requirement exists in VM/ESA 2.4.0 for ESA or XC V=V machines running in 24-bit addressing mode, but not for ESA V=V machines running in 31-bit addressing mode.) The RIO370 area must be below the 16MB line.

Extended Count-Key Data (ECKD*) Architecture: ECKD subsystems can run Count-Key Data (CKD) channel programs when necessary. This compatibility facility provides full functional compatibility with CKD for channel programs that do not attempt to access Alternate, Diagnostic (CE), or Device Support (SA) tracks and do not attempt to format Home Address and Record Zero on any track. CKD-compatibility mode does not, however, assure the same performance that would be expected from a CKD control unit. For this reason, all

CKD-compatibility-mode-only channel command words (CCWs) should be converted to ECKD CCWs when accessing ECKD subsystems.

Straightforward CKD Read channel programs can be expected to consume more of the channel and control unit resources in a nonsynchronous environment than an equivalent ECKD channel program would. Overall subsystem performance suffers because of the synchronization established between the channel, controller, and device.

CKD Write channel programs, when executed in a nonsynchronous environment, incur a delay of one device rotation for each write command in the channel program.

When writing ECKD channel programs, you may no longer rely on the presence of gaps between fields of a record and between records for performing special operations.

Performance Differences [1.1.5]

VM/RTM: RTM VM/ESA, a real-time monitor, is supported in VM/ESA 2.4.0. Its output looks similar to VM/RTM's in VM/ESA 1.1.5 370 Feature but has different meaning because of architectural differences between the two systems.

VMPRF Replaces VMMAP: VM Performance Reporting Facility (VMPRF), a monitor analysis program, is supported in VM/ESA 2.4.0. Its reporting and file-handling capabilities are better than VMMAP in VM/ESA 1.1.5 370 Feature, but it does not support graphics.

Note: VM Performance Planning Facility (VMPPF) is no longer supported.

CP Monitor: The VM/ESA 2.4.0 monitor collects data in categories called domains rather than in categories called classes. The monitor writes records to saved segments, not to spool files. See “VM/ESA 2.4.0 Monitor” on page 274 for more details on differences.

Checkpoint Start: Checkpoint start is not required because of the VM/ESA 2.4.0 spooling design.

Nondisruptive Transition: Nondisruptive transition is no longer available, but VM/ESA 2.4.0 has V=R machine recovery, which allows the V=R machine to resume operation after a CP hard abend has occurred.

System File Differences [1.1.5]

Converting System Data Files: VM/ESA 2.4.0 does not support conversion of system data files between VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0. System data files include the following:

- Files created by the VM/ESA 1.1.5 370 Feature ACNT, CPTRAP, and MONITOR commands
- Named saved systems, saved segments, image libraries, user class restructure (UCR) files, message repository files, and system trace files
- Virtual machine dumps, CP abend dumps, and CP stand-alone dumps

Spool Files: The spool file format is different in VM/ESA 2.4.0. As a result, you must use the SPTAPE command (along with a cold start) or RSCS to convert spool files from VM/ESA 1.1.5 370 Feature to VM/ESA 2.4.0. This is discussed in "Spool File Conversion Considerations and Procedures [S370]" on page 88.

External Interface Differences [1.1.5]

Many VM/ESA 2.4.0 external interfaces are incompatible with their counterparts in VM/ESA 1.1.5 370 Feature because of new functions since VM/ESA Version 1 Release 1.5 370 Feature. However, many other external interfaces are upwardly compatible.

For details on external interface differences, see the compatibility table chapters.

EMSG Default Setting: The default setting for EMSG is ON in VM/ESA 2.4.0. It was TEXT in VM/ESA 1.1.5 370 Feature. ON means that you see the message number as well as the text. For example, you can get a message from the FILELIST command that looks like this:

- In VM/ESA 2.4.0: `DMSWFL002E File` *`fn ft fm`* `not found`
- In VM/ESA 1.1.5 370 Feature: `File` *`fn ft fm`* `not found`

You can change the system EMSG setting using your SYSPROF EXEC.

Installation Differences [1.1.5]

Some devices are no longer supported. See the *VM/ESA: General Information* book for information on device support.

Installation of VM/ESA 2.4.0 has many differences. For instance, CP can be loaded from a MODULE stored on a CMS-formatted minidisk. You can use the SYSTEM CONFIG file instead of the system definition (SYS) file and real I/O (RIO) file. There are differences in the user directory, and the system name table (SNT) does not exist in VM/ESA 2.4.0. You generate saved segments while the system is operating, so you do not need to convert saved segments during the installation procedure. See Chapter 10, "Preinstallation and Installation Changes and Conversion Considerations" on page 209 for a detailed discussion.

Application Programming Differences [1.1.5]

Assembler Application Programs and Any Program Using 24-Bit Addressing: They should execute without change. Exceptions could be programs that:

- Depend on internal CP or CMS structure or control blocks
- Depend on functions that CP does not provide
- Depend on incompatible CP or CMS external interfaces
- Depend on explicit System/370 architecture implementation

See Chapter 15, "Application Programming Changes and Conversion Considerations" on page 289 for more information.

OS Simulation Services: OS simulation is compatible with VM/ESA 1.1.5 370 Feature CMS. VM/ESA 2.4.0 replaces the MVS/SP™ macro set with MVS/XA™ macro set.

DOS/VSE Simulation: DOS/VSE simulation is compatible with VM/ESA 1.1.5 370 Feature CMS.

User Modifications: The internal structure of CP in VM/ESA 2.4.0 is significantly different from that in VM/ESA 1.1.5 370 Feature. You must rework VM/ESA 1.1.5 370 Feature user modifications in order to apply them to VM/ESA 2.4.0.

CP user modifications based on earlier message-command-processing code no longer work. VM/ESA 2.4.0 improves support for processing message commands with an installation-wide exit, HCPMSU. HCPMSU is an assembler module that allows customization of the message command functions. The *VM/ESA: Planning and Administration* book describes HCPMSU.

Also, user modifications that write error messages or output to the screen may no longer work.

Processor, Device, and Licensed Program Differences [1.1.5]

Processors: See the *VM/ESA: General Information* book for a table showing which processors support which levels of VM/ESA.

Multiprocessor and Alternate-Processor Environments: In System/370 architecture, there were the concepts of a multiprocessor environment and an alternate processor environment. In these environments, two processors shared real storage under the control of a single Control Program. The difference between the two had to do with I/O capability. In a multiprocessor environment, both processors had I/O capability. In an alternate processor environment, only one processor had I/O capability. In ESA/370 or ESA/390 architecture, all processors in a multiprocessor complex always have I/O capability. Therefore, the alternate processor concept does not exist in ESA/370 or ESA/390.

Processor Architecture Mode: VM/ESA 2.4.0 runs in:

- ESA/390 processor architecture mode on S/390 Parallel Enterprise Servers and ES/9000 processors
- ESA/370 processor architecture mode on ES/3090 and ES/4381™ processors

Running VM/ESA 2.4.0 on a S/390 Parallel Enterprise Server™ or ES/9000 processor in ESA/390 mode allows you to use VM data spaces, which provide almost unlimited data addressability and let users of XC virtual machines share data without passing data through CP services. ESA/370 mode does not support VM data spaces.

For more information about VM data spaces, XC virtual machines, and supported processors, see the *VM/ESA: General Information* book.

Device Support: VM/ESA 2.4.0 does not support the same devices as VM/ESA 1.1.5 370 Feature. See the *VM/ESA: General Information* book for information about device support in VM/ESA 2.4.0. Most virtual device and hardware compatibility support remains unchanged.

Note: See the current hardware announcements for configuration information.

Licensed Program Support: VM/ESA 2.4.0 does not support the same licensed programs as VM/ESA 1.1.5 370 Feature. You can get information about licensed products that are supported in VM/ESA 2.4.0 in the document called *Licensed*

Products Migration Matrix for VM/ESA or WHAT IS SUPPORTED WHERE for VM. This document is available on the IBM VM operating system home page (http://www.ibm.com/s390/vm).

Guest Operating Systems: For information about supported guest operating systems, see the *VM/ESA: General Information* book or the *VM/ESA: Running Guest Operating Systems* book.

Compatibility with VM/ESA 1.2.0 or Later [ESA]

VM/ESA 2.4.0 is sufficiently compatible with VM/ESA 1.2.0 and later releases that your conversion should be as simple as a release-to-release change.

Compatibility by Component [ESA]

There are no changes in the overall structure of VM/ESA components in VM/ESA 2.4.0. The following descriptions compare the components in the old and new systems.

CP: Most VM/ESA 2.4.0 CP external interfaces are compatible.

GCS: Most VM/ESA 2.4.0 GCS external interfaces are upwardly compatible and are supported in XA and XC virtual machines.

Dump Viewing Facility: Most VM/ESA 2.4.0 Dump Viewing Facility external interfaces are compatible.

CMS: The CMS level for VM/ESA 2.4.0 is CMS Level 15. CMS Level 15 runs in 370-XA (XA virtual machine) and ESA/XC (XC virtual machine) architectures. Only CMS levels prior to CMS level 12 can run in a 370 virtual machine. Most CMS external interfaces are compatible.

REXX/VM: REXX external interfaces are upwardly compatible.

1.2.0

REXX/VM includes the full REXX Level 2 support that is available on other SAA platforms. This completes the Level 2 support that was started in VM/ESA 1.2.0. The following new input and output functions replace the external user functions with the same names: CHARIN, CHAROUT, CHARS, LINEIN, LINEOUT, LINES, and STREAM.

VMSES/E: Most VMSES/E commands are upwardly compatible.

TSAF: TSAF is supported in an XA or XC virtual machine with 370ACCOM ON.

AVS: AVS is supported in XA, ESA, and XC virtual machines.

Chapter 9. New Functions and Support

The functions and support described in this chapter are included in VM/ESA 2.4.0 but may not be in your old system.

This chapter contains information for all of the conversions documented in this book. Each section is marked with identifiers to indicate the particular conversions to which it applies. For definitions of the conversion identifiers used in this chapter, see "How to Identify the Topics That Apply to Your Conversion" on page 3.

1.1.5

If you are converting from VM/ESA 1.1.5 370 Feature, some of the functions described in this chapter were added by between-release enhancements and are carried forward to VM/ESA 2.4.0. If you have installed between-release enhancements on your VM/ESA 1.1.5 370 Feature system, you may have seen some of these new functions already. This book does not describe how to exploit these new functions.

You should not exploit new functions until your conversion is complete, so that you can keep your conversion as simple as possible.

See the appropriate publications for complete descriptions of these functions.

Note: New licensed products and devices are also supported in VM/ESA 2.4.0 but, for the most part, are not described in this section. You can get information about licensed products that are supported in VM/ESA 2.4.0 from the *Licensed Products Migration Matrix for VM/ESA*. This document is available on the IBM VM operating system home page (http://www.ibm.com/s390/vm). See the VM/ESA Device Support Matrix for lists of the devices supported in VM/ESA 2.4.0. You can find that document in the *VM/ESA: General Information* book and also on the VM home page.

This chapter contains the following major sections:

- "Planning [ALL]"
- "Service [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]" on page 178
- "Administration [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]" on page 181
- "Application Programming [ALL]" on page 187
- "Real System Operation [1.1.5, 1.2.0, 1.2.1]" on page 203
- "Virtual Machine Operation [1.1.5]" on page 204
- "Diagnosis [1.1.5, 1.2.0, 1.2.1, 1.2.2]" on page 205

Planning [ALL]

Asynchronous Data Mover Facility [1.1.5, 1.2.0, 1.2.1]

Support for the ADMF hardware feature was added.

CD-ROM Supported as an Installation Media [1.1.5, 1.2.0]

VM/ESA supports CD-ROM as a new installation media.

CMS Communications Directory Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2]

Prior to VM/ESA 2.1.0, if you IPLed CMS with PARM NOSPROF, CMS did not load the system-level communication directory. On VM/ESA 2.1.0 and later, CMS loads the system-level communication directory from SCOMDIR NAMES * even when you use PARM NOSPROF.

CMS Dynamic Timezone Support [1.1.5, 1.2.0, 1.2.1]

Fields are updated to reflect the current time zone. A multitasking event, VMTIMECHANGE, lets multitasking applications monitor the time zone changes. Files created immediately before or after a timezone change may not have monotonically increasing timestamps.

CMS Productivity Aids Performance Enhancement [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

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 [1.2.0, 1.2.1, 1.2.2, 2.1.0]” on page 266.) This will improve the performance of these commands and also aid in the servicing of these parts.

CP Configurability and Related Enhancements [1.1.5, 1.2.0, 1.2.1, 1.2.2]

ALTERNATE_OPERATORS SYSTEM CONFIG Statement [1.1.5, 1.2.0]

The new ALTERNATE_OPERATORS statement that you can use in your SYSTEM CONFIG file lets you specify up to eight alternate operator user IDs. If the current primary system operator logs off and the default primary system operator is not logged on, one of the alternate user IDs automatically becomes the primary system operator if the user ID:

- Is logged on
- Has the appropriate privilege class.

CP Device Sensing [1.1.5]

It is no longer necessary to define all real devices to CP in a DMKRIO file. During initialization, CP tries to sense the real devices attached to the system. If CP can sense the device, it is not necessary to define it. Some older devices cannot be sensed, however. For these devices an RDEVICE statement must be added to the SYSTEM CONFIG file. (The DMKRIO file is obsolete.) An RDEVICE statement is also needed if you want to define other device characteristics that CP cannot sense. For example, for an impact printer you might want to define the output spooling classes or form numbers that the printer can process.

CP Dynamic Configuration [1.1.5]

Many CP parameters can be changed without having to rebuild CP. The parameters include:

- Most I/O definitions (previously defined in DMKRIO)
- All system definition parameters (previously defined in DMKSYS)
- All logo definitions (previously defined in DMKBOX)

These parameters can be set by statements in two external configuration files, the SYSTEM CONFIG and LOGO CONFIG files. During installation, these files are on a minidisk known as the *parm disk*. (Two parm disks are created—one is a backup in case the other becomes corrupted.) The files can be changed at any time. To make SYSTEM CONFIG changes effective, re-IPL CP. It is not necessary to rebuild CP. LOGO CONFIG changes are effective immediately—it is not necessary to rebuild or re-IPL CP.

Many of the parameters can be changed dynamically by CP commands. By setting the parameters dynamically, you can avoid an IPL. To make the changes permanent, update the system configuration file so that the new settings are used on subsequent IPLs.

If you want to use V=R machine recovery, you must define all devices that will be dedicated to the V=R guest in HCPRIO ASSEMBLE. Failure to do so causes preferred guest recovery to fail and requires re-initialization of the guest upon completion of the IPL sequence.

The use of HCPSYS, HCPRIO, and HCPBOX files instead of SYSTEM CONFIG and LOGO CONFIG files is still supported.

Note: The parm disk must be CMS-formatted. The two parm disks created during installation are already CMS-formatted. If a parm disk is allocated but not CMS-formatted, the system enters a disabled wait state, code 6758. If you get this wait code, you can use the alternate parm disk at IPL time; to do this, use the Stand-alone Program Loader in console mode and change the parm disk extent on the Stand-alone Program Loader screen to the alternate parm disk extent.

CP Configurability II [1.1.5, 1.2.0, 1.2.1, 1.2.2]

The dynamic capabilities of VM/ESA have been further enhanced by fully implementing the ES/9000 Dynamic I/O Configuration capability. This facility enables the user to dynamically define, delete and modify channel paths, control units, and devices to the ES/9000 hardware.

Additionally, CP Configurabilty II enables users to:

- Delete real device control blocks (RDEVs).
- Select the next active IOCDS file for the next power-on reset.
- Query subchannel measuring for a device or devices.
- Turn subchannel measuring on and off for a device or devices.
- Query the status of all the channel paths on a system.
- Determine if the ES/9000 is in configuration mode (also called "configmode").
- Turn configuration mode on or off.

- Display all of the starting addresses of the hardware system area (HSA) and their lengths.
- Query a real device to determine its status and the channel paths online to the device or devices.
- Turn VM's dynamic I/O support on or off.
- Query VM's dynamic I/O support.
- Vary a CHPID online or offline.
- Vary subchannels online or offline.
- Display information about the physical, logical, and VM characteristics of all channel paths to a device.
- Limit the number of guest I/O's to a device.
- Remove the requirement of having to specify RDEVs in HCPRIO for V=R recovery.
- Determine whether to prompt the operator for startup information after a poweroff, restart, or shutdown_reipl.
- Change, after IPL, all of the bit maps set during initialization and permit or deny changes, after IPL, to the following bit maps: accepted, dynamic I/O, sensed, and throttled.
- Determine how CP should handle new real devices as they are added to a running system.
- Tape devices can now also be assigned during IPL or by using the VARY (real device) ON command.

The CP Configurability II enhancements provide VM customers the option to no longer have to shutdown to load a new IOCDS containing device changes. Customers are provided improved information and more control of real I/O devices using a wide variety of new commands and facilities.

Note: VM/ESA 2.1.0 was the last release of VM/ESA to allow equivalent definitions for devices and system definitions in HCPSYS, HCPRIO, and the system configuration file. Any future device and system definition additions, changes, and enhancements will be made only to the system configuration file.

CP Nucleus Enhancements [1.1.5]

The CP nucleus can now be stored as a module on a CMS-formatted minidisk, and CP can now access CMS formatted minidisks. (The old method of saving a CP nucleus and performing an IPL on it is still available.) These enhancements let you store many different versions of CP as modules on one or more CMS minidisks. To load CP from a module on a CMS-formatted minidisk, however, the IPL process for CP has changed.

When CP is IPLed, a Stand-alone Program Loader (SAPL) is invoked. SAPL resides at cylinder 0 of a CP-formatted disk; that is, SAPL is the IPL record on a CP-formatted disk. SAPL loads a module from a CMS-formatted minidisk and passes control to the module. The name of this module, the offset of the CMS minidisk it is to be read from, and the load point (for relocatable modules) are specified at the time the IPL record is written to disk. You can override SAPL settings by specifying the device number of a console as a LOADPARM at IPL.

When a LOADPARM is specified, SAPL displays a menu that lets you change the current settings. For example, you can change the name of the module to be loaded.

You can write your own IPL record on a CP-formatted disk by using the Stand-alone Program Loader Creation Utility. This utility displays the same menu as SAPL when a LOADPARM is specified. SAPL and the creation utility are documented in *VM/ESA: Planning and Administration*. To write IPL records while running CMS, use the SALIPL command. The SALIPL command is described in *VM/ESA: CP Command and Utility Reference*.

Cross System Extensions (CSE) [1.1.5]

CSE allows users attached to a single system to participate in a multisystem environment. You can connect up to four systems in a complex.

Cryptographic Facility [1.1.5]

The Cryptographic Facility is a hardware feature that provides a method of protecting data in computer and communication systems from unauthorized disclosure. It involves the process of transforming plain-text into cipher-text (encipherment) and the reverse process of turning cipher-text into plain-text (decipherment).

The Cryptographic Facility is available only in ESA virtual machines and only if a real Cryptographic Facility is installed on the processor. VM/ESA does not provide a software simulation of this hardware.

Data Migration for 3990 Model 6 Consideration [1.1.5, 1.2.0, 1.2.1]

3990 model 6 storage control unit does not support 3380 track compatibility mode on 3390 DASD. 3390 DASD attached to a 3990 model 6 control unit can operate only in native mode.

DBCS Console Input Support [1.1.5]

VM/ESA now has input support for DBCS. A terminal user can input mixed DBCS data on the input line of the terminal screen. Some CP commands have been enhanced to accept mixed DBCS data. Existing CMS commands that accepted mixed DBCS data in fullscreen CMS can now be entered in line mode.

DFSMS/VM——Storage Management [1.1.5]

DFSMS/VM is an additional facility supplied with VM/ESA that simplifies data storage management. It offers tools that allow you to convert data to new storage devices quickly and easily. It includes a high-performance data mover, an automated move process, and an online interactive user interface. DFSMS/VM also provides functions to help you manage data more efficiently on your current storage devices as well as on new devices you acquire.

DFSMS/VM also provides system management for SFS files. It is designed to make more efficient use of storage by automatically and periodically performing functions that:

- Move low-activity files from user storage to DFSMS/VM-owned storage
- Eliminate aged files that are no longer required by the general user

The transfer of control for files in primary user storage to DFSMS/VM-owned storage is called *migration*.

The DFSMS/VM space management functions are:

- Automated space management
- Commands for space management, when you do not want automated space management
- Expiration of old files
- Expiration of files that DFSMS/VM has migrated
- Automatic recall of files that DFSMS/VM has migrated

Refer to the DFSMS/VM library for more details about DFSMS/VM and its functions.

Distributed IUCV [1.1.5, 1.2.0, 1.2.1]

Applications within an ISFC collection now have the option of using distributed IUCV with the addition of a new SYSTEM CONFIG statement, DISTRIBUTE. Distributed IUCV allows an application or system to fully exploit distributed IUCV capabilities by either of the following:

- Explicitly specifying which system its intended target resides on when invoking an IUCV CONNECT
- Allowing full transparent distributed IUCV, which means that IUCV will first attempt to satisfy a CONNECT on the local system and then will attempt to locate the target on a system within the CS collection

IUCV applications will behave the same in a distributed environment as they do on a local system with the following exceptions:

- PURGE and REJECT will only be honored on the local system. Once a message is sent to the other system, it is considered delivered.
- The PRIORITY and MSGLIMIT directory specifications must be present on both systems if they are to be honored.
- The maximum data length will be limited to 1M per message.

For information on the DISTRIBUTE system configuration statement, see the *VM/ESA: Planning and Administration*.

Dump Space Allocation [1.1.5]

Spool space can now be reserved exclusively for dumps. If you are using a SYSTEM CONFIG file, specify the DUMP operand on the CP_OWNED statement.

If you are using HCPSYS, another way to reserve space is by specifying the new D operand on the SYSCPVOL macro to indicate that the spool space on a volume is to be reserved exclusively for dumps. The SYSCPVOL macro replaces the SYSOWN macro of VM/ESA 1.1.5 370 Feature.

Dump space can be allocated in discontiguous sets of cylinders or extents in spooling space. CP can also allocate dump space dynamically during system operation.

Dynamic CP Exits [2.2.0, 2.3.0]

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 it's ability to influence subsequent processing in the module containing the exit point is limited. 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.

Enterprise Systems Connection Architecture® Support [1.1.5]

Enterprise Systems Connection Architecture (ESCON) includes I/O channels that transmit data through fiber optic cables, providing higher data transfer rates. ESCON hardware architecture provides the functions and protocols that let channels and control units support the ESCON interface.

The dynamic switch is an integral part of the attachment of devices to systems with ESCON. Dynamic switches connect multiple control units and channels that implement ESCON. Control units can be dynamically switched among connecting channels within a processor complex or in different processor complexes. Depending on the configuration requirements, channels and control units can be attached to the dynamic switch in any combination.

This support affects external interfaces as follows:

- QUERY VIRTUAL SWITCHES is a new command.
- The SET MITIME command can monitor dynamic switch devices.
- The IUCV user directory control statement supports a new value on the *cpsyserv* parameter: *CONFIG.

  Note: Use *CONFIG only if your virtual machine is running the Enterprise Systems Connection Manager Services (ESCM) licensed program.
- The OPTION user directory control statement supports a new parameter, RMCHINFO.
- The following commands have new responses or have a new field in existing responses: ATTACH, DETACH *real device*, DETACH VIRTUAL, HALT, QUERY MITIME, QUERY *real device*, QUERY VIRTUAL ALL, REDEFINE, and RESET.
- The SET CPTRACE command can trace a channel subsystem I/O.
- New DIAGNOSE codes `X'A4'` and `X'A8'` support I/O being issued on a specific path.
- The RDEVICE system configuration statement recognizes dynamic switch types. Specify the operands TYPE UNSUPPORTED DEVCLASS SWITCH, as follows:

  ```
  rdevice rdev type unsupported devclass switch
  ```

- The RDEVICE macro in HCPRIO recognizes dynamic switch types supported by VM/ESA 2.4.0. (HCPRIO replaces DMKRIO.)
- Monitor record MRSYTCOM has new data.

Extended Count-Key Data Channel Programs [1.1.5]

VM/ESA provides support for extended count-key data (ECKD) channel programs allowing DASD to be attached at distances greater than 400 feet from the central processing unit. This method of I/O subsystem communication is required for new and future direct access devices, storage controllers, and channels. ECKD channel programs are for use by both CP and guest virtual machines.

Note: DASD on ESCON channels must be formatted without filler records.

FBA DASD Support [1.1.5]

VM/ESA provides full support for the following FBA DASD: 3370, 9332, 9335, and 9336. (Note that 3370s and 9332s are not supported as installation devices.)

Full MP Capability for VMCF [1.2.0, 1.2.1, 1.2.2, 2.1.0]

The Virtual Machine Communications Facility (VMCF) no longer has to run on the master processor. This may result in improved performance and capacity for systems that run applications (such as TCP/IP, OV/VM, and RACF/VM) that use VMCF.

Guest Support for the Storage-Protection-Override Facility [1.1.5]

VM/ESA supports the storage-protection-override facility for guest use. This enables CICS/ESA® on MVS or OS/390 to take advantage of the improved reliability that this facility provides. VM/ESA enables guests to use this facility when installed on the machine. For details on this facility, see the *IBM ESA/390 Principles of Operation*.

Guest Cryptographic Support [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]

This support provides VM/ESA guest virtual machine access to the S/390 CMOS cryptographic coprocessor by extending the existing VM/ESA 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/ESA. ISCF is currently the only IBM application program interface to the S/390 CMOS cryptographic hardware.

Guest Support for QDIO Facility [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]

VM/ESA provides guest support for the Queued-Direct-I/O (QDIO) Facility on S/390 CMOS processors that support this new I/O architecture. VM/ESA provides guest (dedicated device) support and dynamic I/O support for OSA Direct Express, which uses the QDIO Facility for data transfer between the S/390 processor and a Local Area Network.

IEEE Floating Point Support [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]

New function has been added to support IEEE Floating Point Hardware on IBM S/390 G5 Servers. The VM/ESA Control Program was 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, which are provided with the IEEE Floating Point hardware first introduced on the S/390 G5 Server.

This support includes preservation and restoring 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.

Integrated Adaptor Support [1.1.5]

VM/ESA supports I/O devices that are attached to 9221 processors through an integrated adapter mechanism instead of a connection through a channel. This support provides a conversion path to the ESA/370 and ESA/390 versions of the 9221 family of processors from various System/370 architected systems, including 43xx and 9370 processors. Significant changes to the I/O equipment or attachment mechanism are not required. VM/ESA 2.4.0 provides support that extends the IPL function to 9335 and 9336 FBA DASDs.

Note: 3370 and 9332 FBA DASD are also supported by VM/ESA except as an installation device.

ISFC Broadcast Routing [1.1.5, 1.2.0, 1.2.1]

Broadcast Routing removes the requirement that all VM systems be directly connected in a CS collection. APPC and IUCV traffic between a source and target system is now forwarded through the intermediate systems. Global *IDENT Resource and Gateway identifies and revokes are also broadcast throughout the CS collection. This reduces the number of CTC links in collections of 3 or more VM systems. Note that fully connected collections will still work.

ISFC will always choose the shortest path when routing communications. Existing conversations are not re-routed when a shorter path becomes active between a source and target system. However, any new conversations will use the shorter path.

When an ISFC link goes down, existing conversations over the affected path are terminated. ISFC does not attempt to re-route these conversations over an alternate path. If the conversation is re-started and an alternate path exists, the conversation will succeed.

Your CS collection can contain pre-VM/ESA 1.2.2 systems in addition to VM/ESA 1.2.2 and later systems. However, care must be taken to ensure that full connectivity is maintained, because the pre-VM/ESA 1.2.2 systems will not forward communications.

Note: Access to private resources using an LU_name_qualifier of *USERID and originating on a pre-VM/ESA 1.2.2 system requires a direct link to the target system. This is because the pre-VM/ESA 1.2.2 system does not provide enough information for the VM/ESA 1.2.2 and later systems to properly route private server connections. A direct link would not be required if the pre-VM/ESA 1.2.2 system uses the System Gateway of the target system as the LU_name_qualifier instead of *USERID.

VM/ESA 1.2.0 and VM/ESA 1.2.1 will allow their Global *IDENT resources and gateways to be forwarded by a VM/ESA 1.2.2 or later system. VM systems running the PWSCF PRPQ connected to a VM/ESA 1.2.2 or later system will forward their Global *IDENT resources and gateways to the VM/ESA 1.2.2 or later system. However, the VM/ESA 1.2.2 or later system will NOT forward these resources and gateways to other VM systems in the CS collection.

Logical Device Limit Relief [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

The maximum number of logical devices that can exist concurrently on VM/ESA has been increased from 4096 to 32,768. The default is 4096. The new CP SET MAXLDEV and QUERY MAXLDEV commands allow you to change or determine the current maximum.

LOGONBY Support [1.1.5, 1.2.0, 1.2.1]

Support was added to allow authorized users to logon to another user ID. The authorization is given using the LOGONBY Directory Control Statement. After using the BY *byuserid* operand on the LOGON command, the user uses their own password to log on and use the virtual machine.

Minidisk Cache Default Setting at Initialization [1.1.5, 1.2.0]

Updated DEFNUC Parameter [1.1.5]: The default for the BUFFSIZ parameter of the CMS nucleus generation macro (DEFNUC) has been increased from 12 to 20. This parameter sets the default maximum size of the buffer for read/write cache for files residing on SFS.

New DEFNUC Parameter [1.1.5, 1.2.0]: A new parameter, MDBUFSZ, has been added to the DEFNUC macro to let you change the default maximum size of the buffer for read/write cache for files residing on minidisks. See the *VM/ESA: Planning and Administration* book for more information on CMS nucleus generation and the DEFNUC macro.

Minidisk Cache Enhancements [1.1.5, 1.2.0, 1.2.1]

With the enhancements to the CP minidisk cache feature, the following are potential items to consider when migrating:

- Remove expanded storage from system if added specifically for minidisk cache.
- Review storage allocation for minidisk cache.
- Use SET MDCACHE or SET RDEVICE commands instead of SET SHARED to enable minidisk cache on volumes.
- Enable caching for minidisks that were poor candidates in the past.
- Disable caching for minidisks that are poor candidates.
- Disable minidisk cache fair share limit for key users.
- Reformat some minidisks to smaller blocksize.
- Prepare for minidisk caching on devices shared between first and second level systems.
- Avoid mixing standard format and non-standard format records on the same cylinder.

For more information on CP Minidisk Cache, see the *VM/ESA: Planning and Administration*.

Multiple Levels of CMS on VM/ESA [ALL]

In order to ease conversion to VM/ESA 2.4.0, IBM provides limited support for multiple levels of CMS on VM/ESA 2.4.0. You can continue to use your production CMS with the CP component from VM/ESA 2.4.0, and then gradually convert 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 VM/ESA 2.4.0.

Note: The CMS component from VM/ESA 1.1.5 370 Feature is supported on VM/ESA 2.4.0 in a 370 virtual machine. Only CMS levels prior to CMS Level 12 can run in a 370 virtual machine.

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.

When IBM discontinues service of a particular VM/ESA release, then support for the CMS component of that release under this offering also ends.

Note: VM/ESA 2.4.0 does not support running the VM/ESA 2.4.0 CMS on a backlevel CP.

Object Directory Compatibility [ESA]

Object directory compatibility is supported when the object directory created with the VM/ESA 2.4.0 DIRECTXA MODULE is used by currently supported releases of CP (except VM/ESA 1.1.5 370 Feature).

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 VM/ESA 2.4.0 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 VM/ESA 2.4.0 HCPDIR ASSEMBLE module, which is used to create DIRECTXA MODULE.

Open Systems Adapter Support Facility (OSA/SF) for VM/ESA [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

OSA/SF for VM/ESA is an optionally-installed facility supplied with VM/ESA that allows you to customize the modes of operation of an Open Systems Adapter (OSA). An OSA is a unique S/390 channel type that connects the processor either directly to a LAN or by LAN emulation to an Asynchronous Transmission Mode (ATM) based network.

An OSA has been supported in a VM/ESA environment since VM/ESA 1.2.1. Prior to VM/ESA 2.2.0, however, OSA/SF services have been available only from OSA/SF for MVS/ESA™ or from OSA/SF as a base, non-exclusive element of OS/390.

OSA/SF for VM/ESA runs in its own virtual machine and therefore has its own user ID. It also needs an administrative user ID to handle its installation. After OSA/SF for VM/ESA is installed and operational, CMS user IDs can access it to manage the OSAs in the VM/ESA environment. Alternatively, the OSA/SF OS/2® interface files can be downloaded to a programmable workstation on which OS/2 WARP is running.

POSIX User Data Base Support [1.1.5, 1.2.0, 1.2.1, 1.2.2]

New CP directory statements were added to allow you to define POSIX groups and specify the POSIX information for a user entry. The new statements are: POSIXGLIST, POSIXGROUP, POSIXINFO, and POSIXOPT.

Real Multi-Path Lock Facility (MPLF) Support [1.2.0, 1.2.1, 1.2.2, 2.1.0]

In addition to the MPLF simulation support for TP guests, the real Multi-Path Lock Facility (MPLF) support provided by CP enables virtual machines to use the real MPLF facilities of a D/T 3990 model 6 DASD control unit.

Removable Media Services (RMS) Free Drive Support [1.2.0, 1.2.1, 1.2.2, 2.1.0]

Diagnose Code X'254', Access Real Subsystem, allows the DFSMS/VM RMS virtual machine to issue nondrive dependent I/O to a 3494 or 3495 tape library subsystem, without requiring the machine to have a CP-attached tape drive.

REXX/EXEC Migration Tool for VM/ESA [1.1.5]

The REXX/EXEC Migration Tool for VM/ESA (VM/ESA MIGR) helps you convert REXX, EXEC 2, DMKRIO, and DMKSYS 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 information about these changes through Help panels.

The *VM/ESA: REXX/EXEC Migration Tool for VM/ESA* shows how to use this tool.

Saved Segment and Named Saved System Enhancements [1.1.5, 1.2.0]

Changes to CP allow some functions to run concurrently that previously had to run consecutively:

- Multiple saved segment loads, finds, and purges (using DIAGNOSE code X'64' or functions that call DIAGNOSE code X'64') can run at the same time or while a SAVESEG or SAVESYS command is being processed. Previously (on VM/ESA 1.2.0), only one saved segment or saved system operation could occur on the system at any given time.
- Users can IPL saved systems and load saved segments into their virtual machines while other users are doing the same. Previously (on VM/ESA

1.2.0), more than one user could not IPL saved systems or load saved segments at the same time.

- Except in rare cases, saved segments and saved systems can be loaded, saved, or purged while the SPTAPE command is being used to dump or load a saved segment or saved system.
- A user having the proper authority can interrupt a SAVESEG or SAVESYS operation with the PURGE NSS command. Previously (on VM/ESA 1.2.0), the purge had to wait for the save to complete.

TCP/IP Feature for VM/ESA [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

TCP/IP Feature for VM/ESA contains the functions available in IBM TCP/IP Version 2 Release 4 for VM and provides the following enhancements:

Note: TCP/IP Feature for VM/ESA does not run on VM releases prior to VM/ESA 2.3.0.

- NFS support of Byte File and Shared File Systems (BFS, SFS)

  In addition to minidisk files, the Network File System (NFS) allows a client to access the CMS Byte File System (BFS) and Shared File System (SFS) as if they were local to the client. Clients can now access these files using the enhanced MOUNT and MOUNTPW commands.

- Expanded CMS/internet Mail capability

  This support lets CMS end users send mail and file objects through a TCP/IP network using enhanced CMS utilities. While a similar capability exists today through separate, TCP/IP-provided utilities, this support incorporates the capability in existing CMS facilities (NOTE and SENDFILE).

- Line Printer Client

  The current Line Printer Client (LPR) operations have been enhanced to allow the end user to have print data processed by the RSCS server. Specifically, this enhancement allows users to select either asynchronous printing via (RSCS) or the currently supported synchronous delivery. Additional flexibility is provided with support for the specification of user-defined nicknames.

- Configuration of Servers

  A new server definition methodology provides a way to easily define, or change the attributes of the set of server virtual machines used by TCP/IP. This implementation focuses on a standard method of server configuration and consolidates that configuration into a single file. The ability to simplify the replication of TCP/IP servers is further complemented by the ability to add new application protocols by simply defining them in a CMS NAMES file.

- SMTP Performance enhancement

  The SMTP (Simple Mail Transfer Protocol) performance has been improved by increasing the throughput of the server. The throughput gains are realized by reducing synchronous minidisk I/O, converting to asynchronous spool services, and improved use of data-in-memory techniques. All current SMTP server functions are still supported with these enhancements. Robust debugging support through extensive tracing capability is also provided in the server.

- Base Improvements

Many new functions improve the usability, function, performance and serviceability of TCP/IP in the VM environment. These include multiple logical hosts, selective trace, performance monitoring, TN3270E printing, multi-port Telnet server, and improved server inactivity detection.

TCP/IP Feature Enhancements [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]

TCP/IP Feature for VM/ESA has been enhanced as follows:

- Expanded RouteD Capability

  This enhancement is seen as a major upgrade in function and reliability of the RouteD server in TCP/IP. The routing daemon (RouteD) is a server that implements the Routing Information Protocols (RIP). The RouteD server dynamically creates and maintains network routing tables using RIP. Enhancements being provided over the existing support include:

  – RIP Version 2. This is an extension of the RIP 1 protocol and provides the following features:
    - Route Tags to provide EGP-RIP and BGP-RIP interactions
    - Immediate Nets Hop for shorter paths
    - Authentication of RIP-2 packets for routing update security
    - Configuration switches for RIP-1 and RIP-2 packets
    - Supernetting support

  – Variable Subnetting

  – Virtual IP Addressing (VIPA)

  – Enhanced messages and debugging

- NFS Version 3 support

  The following changes may be of interest to end users:

  – NFS Version 3 is supported allowing for improved efficiency of communications between NFS client and server.

  – Your administrator can configure a list of EXPORTed file systems, allowing Windows® clients, for example, to use point and click to get access to VM file systems.

  – New parameters were added to the MOUNT command allowing you to use your LOGON BY privileges as well as specifying both a logon password and a minidisk link password in the same mount string.

  – A new SMSG QUERY RESOURCE command allows a CMS user to see what file systems are active for a given VMNFS server. A new SMSG QUERY CONFIG command shows you the configuration parameters used when the VMNFS server was started.

  NFS administrators will find that:

  – More of the VMNFS server machine's parameters can be tailored and you can refresh server configuration parameters without stopping the server.

  – NFS security has also been enhanced with two new exits VMNFSMON and NFSSMSG. These exits provide additional MOUNT options and the screening of SMSG requests.

- FTP Enhancements

  A new SMSG FTPSERVE command which allows for communication with the FTP Server is provided along with three new user exits. In addition, a site specific “Welcome Banner” can now be displayed when connecting to the FTP

Server. Also new is FTP to Reader Support which allows the FTP Server to manipulate VM reader files. Basically this allows the user to:

  - CD to a VM user ID reader rather than minidisk, SFS, or BFS directory.
  - PUT a file to the reader without having to know the password for that user ID.
  - List files in the reader, if authorized.
  - Delete one or more files from the reader, if authorized.

- TCP/IP Customization

  This support adds dynamic expansion of TCP/IP stack pool sizes, minimizing the need to restart the stack as use increases. It also eliminates the need to restart the stack to make certain kinds of configuration changes.

- Enhanced User Authorization

  A new CP Diagnose interface will be used by VM's FTP and NFS servers to allow access to user resources without requiring more than one password to be supplied and without requiring LOGONBY base passwords to be revealed. Three new CMS CSL routines DMSESM, DMSPWCHK, and DMSLINK are also provided for this support.

- Native ATM for TCP/IP

  This support enables TCP/IP to use an Open Systems Adapter configured to support Native ATM. The DEVICE and LINK statements can designate an Open Systems Adapter to operate in Native ATM mode.

- SMTP support

  Two new service extensions have been added. The 8-bit Multipurpose Internet Mail Extension (MIME) allows for the exchange of messages in which the content body consists of an 8-bit MIME message containing arbitrary octet-aligned material. The SMTP Message Size Declaration service extension allows for some communication regarding message size between the client and server SMTP. The use of this extension allows the server SMTP to identify mail that is too large to be processed (based on the client's estimated size) before any transfer is attempted. Additionally, the SOURCEROUTES configuration statement has been modified providing the ablility to remove source routes from the RCPT TO or the MAIL FROM statements.

Virtual Disks in Storage [1.1.5, 1.2.0]

VM/ESA now supports a new type of temporary FBA minidisk known as a virtual disk in storage. Virtual disks in storage are allocated in ESA/370 address spaces in system storage instead of on DASD. Because I/O overhead is avoided, virtual disks in storage may be faster to use than other minidisks. Having a real FBA DASD in the system configuration is not required.

The new VDISK operand of the FEATURES statement in the system configuration file contains default settings for the maximum storage available for all virtual disks in storage on the system and the maximum storage available for virtual disks in storage created by a single user with the DEFINE command. New QUERY VDISK and SET VDISK commands are provided to display information about existing virtual disks in storage and storage limits and to set new storage limits.

Virtual disks in storage are defined, linked, and detached like other minidisks and must be formatted when first created. Virtual disks in storage defined by MDISK statements in the directory can be shared with other virtual machines. As with permanent minidisks, access is controlled by link mode passwords or an external security manager. A virtual disk in storage is created when the first user links to it and destroyed when the last user detaches it or logs off. Virtual reserve/release is supported for virtual disks in storage, which allows data sharing among multiple guests.

Virtual Machine Size [1.1.5]

In VM/ESA 1.1.5 370 Feature, maximum virtual machine size was 16MB. In VM/ESA 2.4.0, maximum virtual machine size (primary address space) is 2047MB. Also, the sizes of named saved systems and saved segments can now be 2047MB. You can now run applications that require very large storage sizes in virtual machines.

To accommodate this new limit, the following external interfaces are changed in an upwardly compatible manner:

- The DEFINE STORAGE and LOGON commands
- The USER user directory control statement
- DIAGNOSE code X'84'

The following new commands also accommodate the new virtual machine size: DEFSEG, DEFSYS, QUERY NSS ALL MAP, and XAUTOLOG.

VM Data Spaces Support [1.1.5]

The new ESA/390 architecture allows an XC virtual machine to create data spaces (nonprimary address spaces). With data spaces, virtual storage is no longer limited to 2047MB; it is virtually unlimited.

Direct access to a data space is limited to local XC machines. Local 370 and XA virtual machines access a data space with a DIAGNOSE instruction.

SFS takes advantage of the new VM data spaces facility. Use of data spaces is transparent to users, although it should provide faster service. The SFS administrator declares which directories are eligible for mapping to data spaces, and the number of data spaces available to the server machine is determined by its directory entry.

Use of VM data spaces can particularly help performance when used with data that is frequently read, but infrequently changed.

Certain CP commands, DIAGNOSE codes, and CP data space macros can access address spaces other than the virtual machine’s primary address space. Use of this function requires an XC virtual machine in access register (AR) mode.

A new DIAGNOSE function, X'250', performs block I/O synchronously or asynchronously between DASD and buffers in host address spaces.

Notes:

1. Some DIAGNOSE codes require that certain operands be in the primary address space.
2. Some of the DIAGNOSE codes do not use data spaces, but have been enabled to run in an XC virtual machine in AR mode.

The Dump Viewing Facility commands and subcommands can process dumps of VM data spaces.

VM/ESA Guest Coupling Simulation Support [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

VM/ESA guest coupling simulation provides for the simulation of one or more complete parallel sysplex systems within a single VM/ESA system image. This environment allows the testing and debugging of guest operating systems while running under VM/ESA. Guest coupling simulation is supported only 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

VM/ESA 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 MSGPROC
- QUERY VIRTUAL MSGDEVICES
- 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 environment. Also, a new CP SET VTOD command has been added to allow Year 2000 testing of a parallel sysplex environment on VM/ESA.

Year 2000 Support [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

Default Date Formats for Command Output [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

The following commands, which provide date output, have been enhanced to provide new date formats that include 4-digit years:

CP Commands	CMS Commands
QUERY CPLEVEL	RDRLIST
QUERY NLS	FILELIST
QUERY NSS	LISTFILE
QUERY READER/PRINTER/PUNCH ALL	IDENTIFY
QUERY TIME	
QUERY TRFILE	
QUERY UCR	

The capability to set a default date format for these commands is provided on a system-wide basis and also on a user (virtual machine) basis.

Date formats are defined as follows:

Name	Format
SHORTDATE	*mm/dd/yy*, *mm/dd*, or *yy/mm/dd* (depending on function)
FULLDATE	*mm/dd/yyyy* or *yyyy/mm/dd*
ISODATE	*yyyy-mm-dd*
VMDATE	Use the virtual machine's default setting.
SYSDEFAULT	Use the system-wide default setting.

Note: SHORTDATE represents the format of an existing date output provided by a particular command. FULLDATE is an extension of existing date formats to include 4-digit years.

To maintain compatibility with prior VM releases, the system-wide and user default settings are initialized as follows:

Type	Default Date Format
System-wide	SHORTDATE
User (virtual machine)	SYSDEFAULT

A new system configuration file statement, SYSTEM_DATEFORMAT, sets the system-wide default date format. If this statement is not included in the system configuration file, the default date format for the system is SHORTDATE.

A new directory control statement, DATEFORMAT, sets the default date format for a user (virtual machine). It can be set to a specific format or to SYSDEFAULT. If the user's default is set to a specific format, it overrides the system-wide setting. If this statement is not included in a user's directory entry, the default date format for the user is SYSDEFAULT.

Two new CP commands, SET DATEFORMAT and QUERY DATEFORMAT, set and query the default date formats for the system and individual users. The user's default can be set to a specific format or to SYSDEFAULT. SET DATEFORMAT can be used to override the directory and configuration file settings.

Default date formats for the FILELIST and RDRLIST commands can also be set with the CMS DEFAULTS command. The default (initial) setting is VMDATE.

If a user's default date format is set to (or defaults to) the system-wide default, and the system-wide default is changed while the user is logged on, the user's default does not change until the user either logs off and then logs back on or issues the SET DATEFORMAT SYSDEFAULT command.

A user's default date format can be changed to a specific date format with the SET DATEFORMAT command at any time while the user is logged on. This change takes effect immediately.

The date format priorities for the CP and CMS commands that support multiple date formats, in order from highest priority to lowest, are:

- Command operand or option
- CMS DEFAULTS command setting (for FILELIST and RDRLIST only)
- User (virtual machine) default
- System-wide default

The following is an example of this behavior:

```
set dateformat isodate
Ready;
q time
TIME IS 13:26:52 EDT THURSDAY 1996-08-29
CONNECT= 26:02:23 VIRTCPU= 002:12.11 TOTCPU= 002:21.52
Ready;
q time shortdate
TIME IS 13:27:11 EDT THURSDAY 08/29/96
CONNECT= 26:02:42 VIRTCPU= 002:12.11 TOTCPU= 002:21.52
Ready;
```

This behavior could have an impact on current programs that issue these commands (without specifying the date format) and expect the date to come back in SHORTDATE format. This concept of changing your virtual machine's date format may help you to identify those programs that rely on a date and which may need updating to support the year 2000 and beyond.

Tools for Converting Date Formats and Testing Year 2000 Readiness [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

The new DateTimeSubtract CSL routine computes differences of time. As part of the calculation process, DateTimeSubtract performs time format conversions and time zone conversions. Because of the format conversions that DateTimeSubtract performs, conversion from a format with a 2-digit year to a format with a 4-digit year is possible. This conversion can assist users in modifying their user-written programs to work with 4-digit years.

SETDATE is an unsupported tool provided for testing purposes. It allows you to change the date and time of last update for individual minidisk and SFS files. The file must be an existing file and cannot be empty. For SFS files, the directory does not have to be accessed.

Year 2000 Support, Phase 2 [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

Additional Date Format Support [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

Support for specifying a date format has been extended to the CMS LISTDS and NETDATA commands. The default date format for NETDATA can also be set with the CMS DEFAULTS command. The NETDATA, NOTE, RECEIVE, and SENDFILE commands support 4-digit-year date formats for entries in the *userid* NETLOG file. For an existing NETLOG file, unless a different default date format is specified, entries are written to the file in SHORTDATE format. However, for a new NETLOG file, if a different default date format is not specified, entries are written to the file in ISODATE format.

Additional Tools for Converting Date Formats and Testing [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

The following CMS commands allow you to do date format corrections:

FIXCENT Sets the correct century for all the files on a particular minidisk

FILESERV FIXCENT Sets the correct century for all SFS objects in a particular file pool

NETLCNVT Converts dates in an existing *userid* NETLOG file from 2-digit-year format to 4-digit-year, or from 4-digit-year format to 2-digit-year

The new CMS Pipelines DATECONVERT stage allows you to do date format conversions and date validations.

The DateTimeSubtract CSL routine now also supports POSIX time conversions.

Service [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

Advanced Digital Delivery Support [1.1.5, 1.2.0, 1.2.1, 1.2.2]

ITNVTSTR EXEC has been added to support Advanced Digital Delivery. For more complete information about this offering, see the *Advanced Digital Delivery User's Guide*. The ITNVTSTR EXEC takes as input a product's electronically delivered service order code from the A-disk. This order's ENVELOPE file is examined to determine how to process the order's corresponding PACKAGE file. The order's PACKAGE file is separated into its component parts; these parts are untersed and stored on some specified target disk for later service application.

S/390 Service Update Facility [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

The S/390 Service Update Facility (SUF) is an internet-based S/390 software service tool that makes ordering and receiving OS/390, 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 (satelite 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 simulaneously.

Details regarding prerequisites, entitlement, and how to obtain SUF can be obtained through the internet from the following URL:

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

VMSES/E [1.1.5]

The VMSES/E component has been substantially rewritten and improved. Especially important improvements include:

- Performance
- A software inventory for storing information about products installed and PTFs applied and requisite relationships among them
- A new utility for managing the software inventory
- A new exec, VMFINS, helps you install
- True exclude list processing
- Automatic identification of parts that have had service applied and that must be built
- Automated building of “select and copy” parts such as execs, XEDIT macros, and HELP files.
- Automatic build of CSLLIBs, DOSLIBs, and generated objects such as text decks

For more information on install and service changes, see Chapter 10, “Preinstallation and Installation Changes and Conversion Considerations” on page 209 and Chapter 11, “Service Changes and Conversion Considerations” on page 249.

VMSES/E Enhancements [1.2.0]

- Two new CSLGEN command options, FILETYPE and NOAUTO, let you specify a file type other than TEXT for text files supplied as input to the build process. When NOAUTO is in effect, the file type specified with the new FILETYPE option becomes the default file type for all ROUTINE, TEXT, and INCLUDE statements.

  The TEXT statement in the CSLCNTRL file supports file types other than TEXT.

  The ROUTINE statement supports new options, FILETYPE and MAP. FILETYPE lets you specify a file type. MAP indicates that a load map file is to be created for the routine.

- The CSLGEN command and the ROUTINE and ALIAS statements support a new COPYTYPE option. The COPYTYPE option lets the library builder specify a file type other than TEXT for text files generated by CSLGEN as a result of using the COPY option.
- New part handlers, invoked by VMFBLD, are supported:
  - VMFBDCLB builds a callable services library.

  - VMFBDDLB builds a DOSLIB, a CMS/DOS phase library.
  - VMFBDGEN builds generated objects such as text decks.

- A new High Level Assembler is supported. The VMFHLASM exec lets you use the High Level Assembler.

- Local modification support introduces a new option, LOGMOD on the VMFSIM CHKLVL command. This new option enables the automatic updating of local version vectors using the information from the AUX files.

VMFREPL EXEC [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

The VMFREPL EXEC is used to support 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 levels of a part

VMSES/E Enhancements [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

- 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 two 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 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.

VMFREM Command [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]

The VMFREM command 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.

Administration [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

Byte File System (BFS) Support [1.1.5, 1.2.0, 1.2.1, 1.2.2]

File pool administration commands and CSL routines have been enhanced to provide limited support for the OpenEdition for VM/ESA Byte File System.

CMS File Pool Enhancements [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

AUDIT Enhancement [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

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.

DIRLIST Enhancements [1.1.5, 1.2.0]

- **[1.1.5]** The CMS DIRLIST command now allows nicknames in the user ID portion of the directory ID. DIRLIST also accepts a nickname defined for a group of users. You use the new NICK option for the new nicknames support.
- **[1.1.5]** DIRLIST also can display linked minidisks using the new MDISK option.

- **[1.1.5, 1.2.0]** DIRLIST supports new options: OWNER, READWRITE, and READONLY. If a temporary access to an SFS directory is needed, these new options do the following:
  - OWNER: If you are the owner of the directory, the directory is automatically accessed as read/write. If you are not the owner of the directory, the directory is automatically accessed as read-only. This is the default, and this is how DIRLIST worked in your old release.
  - READWRITE: You get read/write access to the directory, regardless of whether you are the owner or not. If the directory is already accessed, this option is ignored.
  - READONLY: You get read-only access to the directory, regardless of whether you are the owner or not. If the directory is already accessed, this option is ignored.

New File Pool Administrator Commands [1.1.5, 1.2.0]

- **[1.1.5]** The DATASPACE command is used to make a directory control directory eligible for use in a data space.
- **[1.1.5]** The FILEPOOL FILELOAD command is used to restore base files from a copy of storage group data created by FILEPOOL BACKUP.

  Note: If your installation has applied the SFS File Level Restore between-release enhancement (APAR VM44507), then your system already has the new FILEPOOL FILELOAD command.
- **[1.1.5]** The FILEPOOL RENAME command allows an administrator to change the owning user ID of an SFS file space without affecting the directory structure, files, authorities, and aliases of the file space. This function replaces the SFSTRANS and REGRANT sample execs that are shown in the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book and provided on the 193 disk.
- **[1.1.5]** The FILEPOOL ENABLE and FILEPOOL DISABLE commands can be used as alternatives to:
  - The DMSDISSG, DMSENASG, DMSDISFS, and DMSENAFS CSL routines
  - The ENABLE and DISABLE SFS operator commands
- **[1.1.5]** The QUERY FILEPOOL DISABLE command can be used as an alternative to the QUERY DISABLE operator command.
- **[1.1.5]** The FILEPOOL CONTROL BACKUP command allows an administrator to control when a control data backup is scheduled without logging on to an SFS operator console.
- **[1.1.5]** The FILEPOOL MINIDISK command allows an administrator to add minidisks to storage groups in a file pool while the server continues processing user requests. Minidisks can still be added to storage groups while in dedicated maintenance mode processing using the FILESERV MINIDISK command.
- **[1.1.5, 1.2.0]** The FILEPOOL LIST BACKUP command lists the SFS objects that are contained in a user storage group backup file. It works for backups created in any release of VM.
- **[1.1.5, 1.2.0]** The FILEPOOL LIST MINIDISK command lists the SFS base files that have at least one data block on a specific SFS file pool storage group minidisk. It works when the SFS file pool server is at VM/ESA 1.2.0 or later.

All of these new commands are described completely in the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book.

Enhanced File Pool Administrator Commands [1.1.5, 1.2.0, 1.2.1]

- **[1.1.5, 1.2.0]** The SET THRESHOLD command now supports a new FOR *userid* operand that an administrator can use to set the threshold of a user. (*userid* can be a user ID or a nickname.)
- **[1.1.5, 1.2.0, 1.2.1]** The FILEPOOL RENAME command now supports renaming user IDs not enrolled in the file pool. Explicit authorizations and current locks are transferred to the new user ID. All locks held by the new user ID are released. When the new and old user IDs are authorized to use the same object, the new user ID is granted the higher of the two authorizations.

FILELIST Enhancements [1.1.5, 1.2.0]

- **[1.1.5]** The CMS FILELIST command now accepts SFS directory IDs in addition to file modes.
- **[1.1.5, 1.2.0]** FILELIST supports new options: OWNER, READWRITE, and READONLY. If a temporary access to an SFS directory is needed, these new options do the following:
  - OWNER: If you are the owner of the directory, the directory is automatically accessed as read/write. If you are not the owner of the directory, the directory is automatically accessed as read-only. This is the default, and this is how FILELIST worked in your old release.
  - READWRITE: You get read/write access to the directory, regardless of whether you are the owner or not. If the directory is already accessed, this option is ignored.
  - READONLY: You get read-only access to the directory, regardless of whether you are the owner or not. If the directory is already accessed, this option is ignored.

File System Performance [1.1.5]

The SFS default cache size has changed from 12KB to 20KB. In most environments, this reduces response times and processor usage by requiring fewer APPC server calls and reduced file block I/Os.

Note that you can change the cache size by changing DMSNGP.

Enhanced Date Support for FILELIST and LISTFILE [1.1.5]

- The new ALLDATES option of the FILELIST and LISTFILE commands allows the user to obtain the new extended information for files in a directory. The extended information includes: Date of Creation, Time of Creation, Date of Last Reference, Date of Last Update, and Time of Last Update.
- The new BEFORE, AFTER, and TODAY options of the FILELIST and LISTFILE commands allow a user to list files that were last written to before a specified date, after a specified date, or on the current date.

Enhanced SFS Asynchronous Interfaces [1.1.5]

CSL routines that communicate with the SFS file pool server have the capability of operating asynchronously.

SFS QUERY FILEPOOL Enhancements [1.1.5]

New QUERY FILEPOOL commands have been added to display the output of the QUERY FILEPOOL STATUS command in a more readable format:

- QUERY FILEPOOL AGENT
- QUERY FILEPOOL CATALOG
- QUERY FILEPOOL COUNTER
- QUERY FILEPOOL CRR
- QUERY FILEPOOL MINIDISK
- QUERY FILEPOOL LOG
- QUERY FILEPOOL OVERVIEW
- QUERY FILEPOOL REPORT
- QUERY FILEPOOL STORGRP

SFS Catalog Reorganization Reduction [1.1.5]

The SFS file pool server now dynamically reclaims unused catalog index blocks. The FILESERV REORG command does not have to be issued as often, which means that the amount of time the SFS file pool server is unavailable to the end user is reduced.

SET and QUERY FILESPACE Commands [1.1.5, 1.2.0, 1.2.1]

- The new SET FILESPACE *userid* command lets you set a default file space ID, which is in effect until the CMS session ends.
- The new QUERY FILESPACE command displays the default filespace that you set using SET FILESPACE.

Using Special Characters in SFS Directory Names [1.1.5]

The following characters can now be used in SFS directory names: @, _, #, and $. Also, SFS directory names no longer have to start with a letter.

I/O Assist Enhancements [1.1.5]

You can now designate individual devices as ineligible for I/O assist. You use the new NOIOASSIST parameter on the ATTACH command or on the DEDICATE control statement. You can use the new NOIOASSIST parameter for devices that normally use I/O assist but do not benefit by it.

Also, some conditions for I/O assist eligibility have been added. These new eligibility conditions make some devices that used to be eligible for I/O assist no longer eligible. Only devices that do not benefit from I/O assist should be affected.

Link and Access Enhancements - VMLINK Command [1.1.5, 1.2.0, 1.2.1]

The VMLINK CMS command provides the functions of the CMS ACCESS and RELEASE commands and the CP LINK and DETACH commands. The user does not need to provide virtual device numbers and access modes.

VMLINK:

- Finds free virtual device numbers and access modes

- Links and accesses minidisks and Shared File System directories
- Recognizes nicknames for minidisks and directories
- Provides a rich assortment of facilities for programmers

Minidisk Link Enhancements [1.1.5]

New link access modes help improve the data integrity of applications that depend on the stability of a minidisk. The new link access modes are:

- “stable”—the minidisk is locked against all write link requests
- “exclusive”—the minidisk is locked against all link requests

Changes include:

- The LINK command has new link access modes for the *mode* operand.
- DIAGNOSE code X'E4' has new link access modes for the *mode* field in the parameter list.
- LINK and MDISK user directory control statements have new suffixes for the *mode* operand.
- The OPTION user directory control statement has new operands, LNKSTABL and LNKEXCLU.
- Responses have changed for QUERY *virtual device* and XLINK DISPLAY.
- Parameters have been added to QUERY LINKS.

Product Enablement Support [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

A new CP SET PRODUCT command and new PRODUCT configuration statement were added to define a product or feature to your VM/ESA system and 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 were 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.

Productivity Aids [1.1.5, 1.2.0, 1.2.1]

DISK LOAD Command Enhancements [1.1.5, 1.2.0]

The DISK LOAD command has been enhanced to load files to file modes other than A.

NAMEFIND Command Enhancements [1.1.5]

The following enhancements were made to the NAMEFIND command:

- The SIZE option was changed to “*,” which means NAMEFIND will allocate as much storage as necessary to buffer the names file.
- The maximum number of tags limit was eliminated.
- The new STEM option can be used to return more than 255 characters for tags and values in the names file. However, CMS commands, such as NOTE and TELL, are limited to 255 character tag values because these values must be returned through the CMS stack.
- NAMEFIND now attempts to allocate its buffer storage above the 16MB line.

NAMES Command Enhancements [1.1.5, 1.2.0, 1.2.1]

The following enhancements were made to the NAMES command:

- **[1.1.5]** The NAMES command now supports different panels to create and maintain your names files. By choosing one of the following new options —MAIL, ALTMAIL, COMDIR, SERVER, PANEL— you will see a different panel to work with to modify a “userid NAMES” file, a “$SERVER$ NAMES” file, or a communications directory. See the *VM/ESA: CMS Command Reference* for examples of these panels.

  MAIL is the default panel. On this panel, the nickname field has been extended to 21 characters and the Userid and Node fields have been extended to 70 characters. The positions of these fields have been rearranged to incorporate these new field lengths.

- **[1.1.5]** The NAMES PF keys have been changed.

  To allow you to specify any number of tags for a nickname, you can now scroll between multiple panels (PF10 and PF11). The Delete PF key has been reassigned from PF10 to PF9.

  Also, the PF5 Find support has changed. It now searches the names file for all occurrences, and allow you to scroll through the matches. The PF5 key is changed to FindQuit while using the support, so you may quit at any time.

- **[1.1.5, 1.2.0, 1.2.1]** The Userid and Node fields will no longer be uppercased.

NOTE, SENDFILE, and NETDATA SEND Enhancements [1.1.5, 1.2.0, 1.2.1]

A new option, CLASS, on the NOTE, SENDFILE, and NETDATA SEND commands lets you specify which spool class you want. The default spool class is A.

A note can have a new USEROPTIONS: line between the OPTIONS: line and the Date: line. This can be used by user applications to list their unique options. The NOTE command does not create a USEROPTIONS: header record. The application or the user must add this record. The SENDFILE command ignores the USEROPTIONS: line.

RECEIVE Command Changes [1.1.5, 1.2.0]

When receiving multiple DISK DUMP files, all files get created on the specified file mode. Also, an entry in the user's NETLOG file is added for each file created. In the old release, just the first file was created and logged in the user's NETLOG file. Also, when receiving DISK DUMP files to a file mode other than A, it is no longer required that you have the A file mode accessed as read/write.

Saved Segment Mapping Interface [1.1.5]

The VMSES/E VMFSGMAP EXEC displays a map of all the discontiguous saved segments, segment spaces, member saved segments, and saved systems defined on the system. This includes saved segments defined to VMSES/E but not yet built. You can select a saved segment from the map and display its definition. The VMSES/E definition of a saved segment includes its DEFSEG parameters, whether it can be defined above the 16MB line, whether it contains logical saved segments, the build parameters, and other information. You can add, change, and delete saved segment definitions. The mapping interface allows you to view saved segment layouts prior to actually building them on the system. You can define multiple saved segment configurations.

System Console Support [1.1.5, 1.2.0, 1.2.1]

System Console support allows the system console to be used as a logon console. The OPERATOR_CONSOLES and EMERGENCY_MESSAGES_CONSOLES SYSTEM CONFIG statements have a new operand, SYSTEM_CONSOLE, which provides the system with System Console support.

SYSTEM SEGID Enhancements [1.1.5, 1.2.0, 1.2.1]

After using the CMS SEGGEN command or the VMFBLD EXEC to build a CMS logical saved segment, copying the system segment identification file (SYSTEM SEGID) to the CMS system disk and resaving CMS is no longer required in every case.

Copying the SYSTEM SEGID to the system disk is *not* required if the contents of an existing logical saved segment are modified (data is added, deleted, or changed).

The SYSTEM SEGID file *must* be copied to the system disk if:

- A new logical or physical saved segment is created
- An existing logical or physical saved segment is deleted
- The relationship between the logical and physical saved segments is changed, such as moving or copying a logical saved segment from one physical saved segment to another

User Class Restructure for Individual QUERY and SET Operands [1.1.5]

You can use the CLASS OVERRIDE file to change the privilege class of individual operands of the QUERY and SET commands. In VM/ESA 1.1.5 370 Feature, you could not change the privilege class of individual QUERY and SET operands. Also, the COMMANDS command and the QUERY COMMANDS command now display information about individual QUERY and SET operands.

Application Programming [ALL]

370-Only CMS Applications in XA or XC Virtual Machines [1.1.5, 1.2.0]

Most CMS applications that used to run only in a 370 virtual machine can now run in an XA or XC virtual machine if you enter the CP SET 370ACCOM ON command. Also, modules generated with the 370 option of the GENMOD command can run in XA or XC virtual machines by entering the CMS SET GEN370 OFF command.

This support adds the following:

- A new operand for the CP SET command: 370ACCOM
- New information about the setting of 370ACCOM in the CP QUERY SET response
- A new operand for the CMS SET and QUERY commands: GEN370

See the *VM/ESA: CP Programming Services* book for more information about 370-only CMS applications in XA or XC virtual machines.

Callable Services Library (CSL) Enhancements [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

The following enhancements have been made to the callable services libraries:

- **[1.1.5]** The direct call interface allows an application or system programmer to call CSL routines directly by routine name rather than using a CALL to DMSCSL. The direct call interface provides performance comparable to the CSL fastpath.
- **[1.1.5]** An interactive mapping command, CSLMAP, has been added. This mapping command provides RTNMAP function in a display format like CSLLIST.
- **[1.1.5]** Many CSL routines have been enhanced to work on minidisk files.
- **[1.1.5]** Two new CSL routines, DMSQFPDS and DMSQFPDD, can be used to query if a file pool is disabled.
- **[1.1.5]** A new callable service library, VMMTLIB, is added. VMMTLIB provides routines that enable application multitasking.
- **[1.1.5, 1.2.0]** The unique ID, which is returned by DMSEXIST, DMSEXIFI, DMSGETDI, DMSGETDF, and DMSGETDX, can be specified on calls to DMSEXIST to query the existence of a particular file, directory, or external object.
- **[1.1.5, 1.2.0, 1.2.1, 1.2.2]** Many CMS record file system CSL routines (located in VMLIB) provide limited support for manipulating Byte File System (BFS) files and top directories.
- **[1.1.5, 1.2.0, 1.2.1, 1.2.2]** VMMTLIB has been moved into the CMS nucleus.

  The MTSEG and USEMTSEG parameters in the CMS nucleus generation macro, DEFNUC, are disabled.
- **[1.1.5, 1.2.0, 1.2.1, 1.2.2]** VMMTLIB now includes routines that call OpenEdition for VM/ESA (POSIX) services. See “OpenEdition for VM/ESA [1.1.5, 1.2.0, 1.2.1, 1.2.2]” on page 194.
- **[1.1.5, 1.2.0, 1.2.1, 1.2.2]** A new callable services library, VMGUILIB, is added. VMGUILIB contains Distributed GUI Toolkit (DT) routines for creating GUI applications.
- **[1.1.5, 1.2.0, 1.2.1, 1.2.2]** A new CSL routine, WorkstationGetAddress, gets the value set for the workstation display address.
- **[1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]** Many CMS record file system (VMLIB) CSL routines have been enhanced to provide support for dates with 4-digit years.
- **[2.1.0, 2.2.0, 2.3.0]** New OpenEdition for VM/ESA callable services have been added. See “OpenEdition for VM/ESA Enhancements [2.1.0, 2.2.0, 2.3.0]” on page 196.

CMS Application Multitasking [1.1.5]

CMS allows an application to divide itself into multiple units or executions and provides the ability of these units, called threads, to run on multiple CPUs simultaneously. The multitasking facilities are available only at the application programming level as routines in a new callable services library, VMMTLIB.

Note: If you have PRPQ P81089 Server Tasking Environment/VM (STEVR/VM), your system already supports application multitasking. Differences between the

PRPQ and CMS multitasking are described in the *VM/ESA: CMS Application Multitasking* book.

CMS Application Multitasking Enhancements [1.2.0, 1.2.1, 1.2.2, 2.1.0]

Thread Blocking I/O [1.2.0]

Thread Blocking I/O is new support that allows CMS multitasking applications to better use CMS/SFS I/O requests with CMS Multitasking *event services*. It lets applications detect the completion of an asynchronously issued CSL routine.

MonitorBufferGet Routine [1.2.0, 1.2.1, 1.2.2]

The new MonitorBufferGet routine obtains the address of the CMS monitor data area, where information about threading operations and POSIX processes is stored.

DateTimeSubtract Routine [1.2.0, 1.2.1, 1.2.2, 2.1.0]

The new DateTimeSubtract routine computes differences of time. As part of the calculation process, DateTimeSubtract performs time format conversions and time zone conversions. Because of the format conversions that DateTimeSubtract performs, conversion from a format with a 2-digit year to a format with a 4-digit year is possible. This conversion can assist users in modifying their user-written programs to work with 4-digit years.

Reserved System Events [1.2.0, 1.2.1, 1.2.2]

The following new reserved system event names have been added:

- **[1.1.5, 1.2.0]** VMSFSASYNC - Shared File System asynchronous request completion
- **[1.1.5, 1.2.0, 1.2.1, 1.2.2]** VMCON1ECB - Input available at the console
- **[1.1.5, 1.2.0, 1.2.1, 1.2.2]** VMPOSGNL - Generation or delivery of a POSIX signal

CMS Pipelines Enhancements [1.1.5, 1.2.0, 1.2.1, 1.2.2]

New Assembler Macro Interface [1.1.5, 1.2.0, 1.2.1, 1.2.2]

The new Assembler macros for pipelines are:

PIPCMD
PIPCOMMT
PIPDESC
PIPEPVR
PIPINPUT
PIPLOCAT
PIPOUTP
PIPSEL
PIPSEVER
PIPSHORT
PIPSTRNO
PIPSTRST

All macros contain a copyright statement in the prologue enabling source to be changed by customers.

The execs to create filter packages have been renamed and updated to encompass Assembler user-written stages.

The changed Filter Package execs are:

PIPGFTXT EXEC replaces PIPGREXX EXEC.
PIPGFMOD EXEC replaces PIPLNKRX EXEC.

The PIPGREXX EXEC and the PIPLNKRX EXEC built filter packages from an input file containing only REXX stage names. Such an input file can be updated to include Assembler stage names and the PIPGFTXT and PIPGFMOD execs will create a new filter package with all the stages listed in that input file.

All CMS/TSO Pipelines Stages, Commands, and Subcommands Are Supported [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

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

For more information, see the *CMS/TSO Pipelines: Author's Edition*.

Console Output Support [1.1.5]

Your application can now issue a line-mode write operation using a virtual start (SIO, SIOF, or SSCH) to write double-byte character set (DBCS) data to the console. In VM/ESA 1.1.5 370 Feature, you could do this only with DIAGNOSE code X'58'.

In VM/ESA 1.1.5 370 Feature, single-byte character set (SBCS) and DBCS data was formatted for display based on the physical size of the screen. VM/ESA now displays SBCS and DBCS data based on the smaller of:

- The physical size of the screen
- The logical line size as set by the TERMINAL LINESIZE command

In your current logon session, if you have issued TERMINAL LINESIZE OFF or have not issued the TERMINAL LINESIZE command, then your new VM/ESA system is compatible with your VM/ESA 1.1.5 370 Feature system in this aspect.

CONSOLE Macro Usability Enhancements [1.1.5, 1.2.0]

The CONSOLE WRITE and CONSOLE READ macros support specifying a register as (*reg*) for the OPTIONS parameter. *reg* is a register that contains the options to be used with the CONSOLE WRITE or CONSOLE READ macro.

CONSOLE WRITE supports a new BRKKEY= parameter to identify break key handling requests.

DIAGNOSE Code X'254' (Access Real Subsystem) [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

DIAGNOSE code X'254' issues a limited set of channel programs to select tape control unit subsystems. It provides a virtual machine with asynchronous access to a subsystem even if all devices in that subsystem are dedicated to other virtual machines or to CP.

DIAGNOSE Code X'270' (Pseudo Timer Extended) [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

DIAGNOSE code X'270' causes CP to store the number of bytes of time information specified in the Ry register at the address specified in the Rx register. 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.

DIAGNOSE Code X'27C' (Product Enablement Verification) [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

DIAGNOSE code X'27C' performs product enablement verification. It requests information about the enablement status of a single product or feature, such as:

- Whether the product is defined to the system
- Whether the product is enabled or disabled
- The contents of the optional DESCRIPTION operand specified on the PRODUCT statement or SET PRODUCT command

EXEC 2 Support for 4-Digit Years [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

Two predefined variables have been added to the EXEC 2 language:

- &FULLDATE is the date in Universal Coordinated Time (UTC) in the form *yyyy/mm/dd*.
- &ISODATE is the date in UTC in the form *yyyy-mm-dd*.

Extract Support for File System Information [1.1.5]

Your application can extract information about space availability for storing files. This is supported for both SFS top directories and minidisks.

New information names have been added for Year 2000 support (4-digit years): FILE_DATE_CENTURY, ACT_FILE_DATE_CENTRY (note that U is omitted), and YEAR2000_SUPPORT.

See the *VM/ESA: CMS Application Development Reference* for details.

GCS Enhancements [1.1.5, 1.2.0, 1.2.1]

Single User Environment Support [1.1.5]

GCS now supports a single user environment as well as the group environment. The single user environment allows multiple users to IPL the same named GCS saved system. Each virtual machine operates independently. Applications that do not require group communications are able to IPL and run without the overhead of group initialization and multiple virtual machines. In VM/ESA 1.1.5 370 Feature, GCS supported only multiple user virtual machine groups.

GCS Command Function Support [1.1.5]

GCS supports the CLEAR command, which permits you to clear the virtual console. Also GCS supports the EXECIO command giving your programs the ability to:

- Read records from a disk or virtual reader into the program stack or a program variable
- Write records from the program stack or a variable to a CMS file or virtual device
- Cause execution of CP commands and the recovery of the resulting output

New GCS Macros [1.1.5, 1.2.0, 1.2.1]

GCS supports new macros:

- **[1.1.5]** DEVTYPE
- **[1.1.5]** ESPIE
- **[1.1.5, 1.2.0, 1.2.1]** GCSTOKEN
- **[1.1.5, 1.2.0, 1.2.1]** QUERY MODDATE
- **[1.1.5]** RDJFCB

GCS Trace Selectivity Support [1.1.5]

This support allows GCS to selectively trace supervisor events while using the ITRACE command.

GCS Trace Entries Enhancements [1.1.5]

Enhancements have been made to:

- GETMAIN trace entries
- Formatted SVC 202 trace entries

GCS Dynamic Address Resolution [1.1.5]

The QUERY ADDRESS command now provides the real address of selected GCS nucleus modules and tables. While these data structures used to be available only in the GCS load map, they can now be dynamically displayed to aid in problem solving.

Dynamic Time Zone Support [1.1.5, 1.2.0]

The SET TIMEZONE command can be used by an operator to change the time in the system. (The SET TIMEZONE command was supported in VM/ESA 1.2.0.)

The CVT macro has a new field that contains the difference between local time and Coordinated Universal Time (UTC), based on what the operator set using the SET TIMEZONE command. An application can now retrieve the latest offset from UTC using the new field in the CVT macro rather than DIAGNOSE code 0. The field in the CVT macro is updated each time the operator issues a SET TIMEZONE.

GCS Query Address Enhancements [1.1.5, 1.2.0, 1.2.1]

The GCS QUERY ADDRESS command now returns the name of, and displacement into, a module or table for a specified address. GCS QUERY MODDATE command is added to return the compilation date of a specified module or a list of modules having the most recent compilation date.

GCS VM/ESA Data Spaces Support [1.1.5, 1.2.0, 1.2.1]

GCS supports VM/ESA Data Spaces through use of the CP macro interfaces. GCS will save and restore your access registers through a call to the GCS supervisor.

GCS Diagnostic Enhancements [1.1.5, 1.2.0, 1.2.1]

GCS has enhanced diagnostic capabilities provided for Data Space support through the SDUMPX macro and by filling in of fields in the SDWA with the access registers at the time of a dump.

GCS Name/Token Support [1.1.5, 1.2.0, 1.2.1]

The GCS Name/Token support, having the GCSTOKEN macro as a user interface, provides a way for programs to share data between two programs running under the same task, or between two or more tasks running in the same virtual machine or in different virtual machines in a single GCS GROUP.

Guest Support for Concurrent Copy Sessions [1.1.5]

VM/ESA provides guest support for Concurrent Copy (CONCOPY) sessions on DASD attached to the 3990 Model 3 Storage Control with the CONCOPY feature installed. CONCOPY sessions permit backing up and copying of data while it is actively being updated. CONCOPY does this by effectively eliminating the time during which updates are not processed. The CONCOPY session captures a “snapshot” of the data. This process is transparent to the end user or application interface. Access to the data appears uninterrupted.

The DASDSYS CONCOPY parameter on the new STDEVOPT directory statement authorizes a guest virtual machine to control and process CONCOPY sessions. The new CP commands QUERY CONCOPY and CONCOPY TERMINATE allow the host system operator to detect and terminate CONCOPY sessions if the guest becomes nonfunctional. Use of the following existing CP commands may indirectly terminate CONCOPY sessions:

- DEFINE (Virtual Device)
- DEFINE STORAGE
- DETACH (Real Device)
- DETACH (Virtual Device)
- FORCE
- IPL
- LOGOFF
- REDEFINE
- SET MACHINE
- SHUTDOWN
- SYSTEM CLEAR
- SYSTEM RESET

Use of the following existing CP commands halts CONCOPY sessions on the subsystem:

- SET CACHE SUBSYSTEM FORCEOFF
- SET CACHE SUBSYSTEM OFF

The halted CONCOPY sessions are not terminated. They must be terminated and then restarted by the applications.

Inter-System Facility for Communications (ISFC) [1.1.5]

ISFC provides high-speed communications between programs running on VM host systems. When inter-connected, these VM systems form a communication services (CS) collection. ISFC support is provided in CP.

Programs can be written to the SAA* CPI Communications interface or the APPC/VM assembler language programming interface.

In a CS collection, VM/ESA systems using ISFC are called VM domain controllers; these domain controllers must be connected by channel-to-channel (CTC) links.

For more information about ISFC, see the *VM/ESA: Connectivity Planning, Administration, and Operation* book.

Java™ and NetRexx™ Support [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

VM/ESA now has the capability of developing, compiling, and running Java™ and NetRexx applications. You run the applications in the OpenEdition 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, Microsoft® Windows 95, or Windows NT®.

Language Environment [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

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

MQSeries® Client for VM/ESA [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

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

OpenEdition for VM/ESA [1.1.5, 1.2.0, 1.2.1, 1.2.2]

The term **OpenEdition for VM/ESA** refers to the VM/ESA implementation of industry-standard open systems interfaces for applications and users. OpenEdition for VM/ESA consists of four parts:

- OpenEdition for VM/ESA Services
- OpenEdition Shell and Utilities Feature for VM/ESA
- OpenEdition for VM/ESA Sockets
- OpenEdition Distributed Computing Environment (DCE) Feature for VM/ESA

OpenEdition for VM/ESA Services is the VM/ESA implementation of three Institute of Electrical and Electronics Engineers (IEEE) Portable Operating System Interface for Computer Environments (POSIX) standards:

- POSIX 1003.1 (known as POSIX.1) - System Interfaces
- POSIX 1003.1a (known as POSIX.1a) - Extensions to POSIX.1
- POSIX 1003.1c (known as POSIX.1c) - Threads

The POSIX.1, POSIX.1a, and POSIX.1c interfaces are provided as C library routines in the IBM C for VM/ESA (C/VM™) run-time library. (The C/VM run-time library is included in Language Environment, which is supplied with VM/ESA.) For programs written in other languages, a language-neutral version of a subset of the

POSIX functions is provided in VM/ESA as a set of callable services library (CSL) routines known as the OpenEdition for VM/ESA callable services. These routines are called by the C POSIX library routines to provide the functions, but are also available to other applications. Programming language binding files are provided for REXX and Assembler applications. In addition, a REXX subcommand environment, ADDRESS OPENVM, is provided so the routines can be invoked as REXX functions. For more information about these routines, see the *IBM OpenEdition for VM/ESA: Callable Services Reference*.

Included in the OpenEdition for VM/ESA Services is a POSIX.1 compliant file system known as the OpenEdition for VM/ESA Byte File System (BFS). BFS is a companion to the CMS Shared File System (SFS) that provides a byte-stream view of files. BFS allows data to be organized and used in a UNIX® style and format.

Like SFS files, BFS files are organized in a hierarchical directory structure and stored in CMS file pools. While supporting the POSIX file system functions and rules, BFS also takes advantage of administration and system management facilities that it shares with SFS. These include space allocation, backup, and DFSMS/VM file migration, as well as other administrative functions.

CMS provides a set of commands, known as the OPENVM commands, that allow users to manage their BFS directories and files and control their related permission and ownership attributes. CMS Pipelines additionally provides the ability to use BFS from pipeline programs.

OpenEdition Shell and Utilities Feature for VM/ESA provides application development tools and an interactive environment in support of the POSIX application environment. OpenEdition Shell and Utilities Feature for VM/ESA is the VM/ESA implementation of the IEEE POSIX 1003.2 Shell and Utilities standard (known as POSIX.2). OpenEdition Shell and Utilities Feature for VM/ESA provides a large set of utilities that aid in program development and in porting applications from other open systems. It also provides a UNIX-like interactive user environment. Users of the OpenEdition shell environment have access to both the shell command set (built-in commands and utilities) and the full CP and CMS command set, as well as both OpenEdition and non-OpenEdition applications. For information about using the OpenEdition shell, see the *IBM OpenEdition for VM/ESA: User's Guide*. For descriptions of the shell commands and OPENVM commands, see the *IBM OpenEdition for VM/ESA: Command Reference*.

OpenEdition for VM/ESA sockets are a set of C language functions that provide an industry-accepted protocol for client/server communication. OpenEdition for VM/ESA sockets correspond closely to the sockets used by UNIX applications. For more information, see *IBM OpenEdition for VM/ESA: Sockets Reference*.

The optional OpenEdition DCE Feature for VM/ESA provides support for distributed computing by providing an authenticated remote procedure call mechanism and the DCE threads package. It also provides distributed directory support for DCE clients. OpenEdition for VM/ESA provides a library of publications on DCE. To gain familiarity with DCE concepts, see *IBM OpenEdition DCE for VM/ESA: Introducing the OpenEdition Distributed Computing Environment*.

OpenEdition for VM/ESA Enhancements [2.1.0, 2.2.0, 2.3.0]

Additional C/VM Library Routines [2.1.0, 2.2.0]

Over 150 C library routines have been added to the C/VM run-time library. See the *IBM C for VM/ESA: Library Reference*.

Additional OpenEdition for VM/ESA Callable Services [2.1.0, 2.2.0]

The following callable services have been added:

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)

The OpenEdition for VM/ESA implementation of the fork (BPX1FRK) service has some limitations not found in other implementations. In certain situations, you may need to modify your application to accomodate these limitations. To avoid the limitations of fork (BPX1FRK), you should consider modifying your application to use spawn (BPX1SPN). To understand the OpenEdition for VM/ESA implementation of fork (BPX1FRK), and to get information about the other callable services and macros that were added, see the *IBM OpenEdition for VM/ESA: Callable Services Reference*. For information about converting **fork()** usage in a C/VM application to **spawn()**, see the *VM/ESA: CMS Application Development Guide*.

Additional OpenEdition for VM/ESA Callable Services [2.1.0, 2.2.0, 2.3.0]

The following callable services have been added:

Service	Purpose
realpath (BPX1RPH)	Find the absolute path name for a relative path name

For more information about these services, see the *IBM OpenEdition for VM/ESA: Callable Services Reference*.

Additional OPENVM Commands [2.1.0, 2.2.0]

If you determine that the processing provided by OpenEdition for VM/ESA implementation of the fork (BPX1FRK) callable service is sufficient for your needs, you must use the new OPENVM SET FORK command to explicitly turn that processing ON before running your program. You can use the OPENVM QUERY FORK command to determine the current setting. For more information about these OPENVM commands, see the *IBM OpenEdition for VM/ESA: Command Reference*.

Additional OpenEdition Shell Commands [2.1.0, 2.2.0]

The following OpenEdition shell commands have been added:

- **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, see the *IBM OpenEdition for VM/ESA: Command Reference*.

Enhanced openvmf (BPX1VM5) Callable Service [2.2.0, 2.3.0]

The openvmf (BPX1VM5) callable service has been enhanced with the new VM5_RESOLVE_PATH function. This function resolves a BFS path name to its full-qualified BFS system root. An equate for this function has been added to the BPXYVM5 mapping macro. For more information about openvmf and BPXYVM5, see the *IBM OpenEdition for VM/ESA: Callable Services Reference*.

REXX Enhancements [1.1.5, 1.2.0, 1.2.1, 1.2.2]

APILOAD Function [1.1.5]

The APILOAD function has been added. It is used to include binding files in REXX/VM execs.

REXX Exploitation of XA [1.1.5]

REXX/VM exploits XA architecture.

REXX Level 2 Support [1.1.5, 1.2.0]

REXX/VM has been enhanced so that customers can take advantage of the Level 2 functions available on other SAA platforms.

Non-I/O Support [1.1.5]: Non-I/O REXX Level 2 support includes support of binary literal strings and binary conversion functions. Also included are enhancements to the VALUE function, which include access to CMS global variables, improved variable support in the DROP and PROCEDURE instructions, and enhancements in parsing templates. Arithmetic operations have been improved so that they are now more accurate.

Some instructions and functions have been changed.

Stream I/O Support [1.1.5, 1.2.0]: REXX Level 2 Stream I/O support adds new functions to help process CMS files and streams of data. Previously, users had to rely on CMS or XEDIT services to do I/O. The new CHARIN, CHAROUT, CHARS, LINEIN, LINEOUT, and LINES functions allow you to process CMS files one line at a time or character by character. The new STREAM function allows the general user to obtain state of the stream and to send commands to it.

The CALL and SIGNAL instructions include the NOTREADY condition, which is raised if an error occurs during an input or output operation.

The PARSE instruction has a new variant, PARSE LINEIN, which parses the next line from the default input stream.

REXX and DBCS Enhancements [1.1.5]

DBCS characters, which used to be allowed only in strings and comments, are now allowed in symbols as well.

If ETMODE is enabled, some SO/SI grouping checks are done on comments. Previously no checks were done on comments.

ETMODE is now only recognized if it is the first instruction in a program. If it is not the first instruction and you use it later in the program, you will get DMSREX488E Error 33. If you do place it as the first instruction of your program, all subsequent uses will be ignored.

REXX Global Exits [1.1.5]

VM/ESA supports new REXX global exits. See the *VM/ESA: REXX/VM Reference* for details.

ADDRESS OPENVM Instruction [1.1.5, 1.2.0, 1.2.1, 1.2.2]

The ADDRESS OPENVM instruction has been added. It is used to call OPENVM-type CSL routines, such as OpenEdition for VM/ESA callable services. OPENVM routines may not follow the usual CSL conventions, such as providing return and reason codes as the first two parameters.

REXX Sockets API [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

VM/ESA supports an API for socket applications written in REXX for the TCP/IP for VM environment. The new REXX/VM built-in function, SOCKET, maps socket calls from the C programming language to REXX. This allows you to use REXX to implement and test TCP/IP applications.

The REXX Sockets API provides socket functions to:

- Process socket sets
- Initialize, change, and close sockets
- Exchange data for sockets
- Resolve names for sockets
- Manage configurations, options, and modes for sockets

For more information, see the *VM/ESA: REXX/VM Reference*.

SFS Limit Processing Enhancement [1.1.5, 1.2.0]

When using an FS macro, the EXECIO command, or OS/MVS PUT or WRITE macros to write to an SFS directory in a VM/ESA 2.4.0 file pool, a disk full error is reported when the file space limit is exceeded. See “Changes to SFS Space Processing [1.1.5, 1.2.0]” on page 300 for more details.

TVISECT Enhancements [1.1.5, 1.2.0]

Four new fields are added to TVISECT to provide additional tape information to the customer-supplied DMSTVI exit interface module. The new fields are:

TVIFLAGS	If on, indicates that the new fields are available
TVIRECFM	Indicates RECFM values
TVILRECL	Indicates LRECL values
TVIBLKSI	Indicates BLKSIZE values

VM/ESA Graphical User Interface (GUI) Facility [1.1.5, 1.2.0, 1.2.1, 1.2.2]

The VM/ESA Graphical User Interface (GUI) Facility lets you run programs on your VM/ESA host system that display on a workstation using a graphical user interface. The VM/ESA GUI Facility consists of a programming interface called Distributed GUI Toolkit (DT) and the CMS Desktop.

To use the VM/ESA GUI Facility, you must:

1. Configure your workstation with an appropriate operating system and connectivity product.

2. Download and install the VM/ESA GUI Facility workstation agent on your workstation.

3. Start the workstation agent.

4. Prepare your VM/ESA session to run GUI programs.

5. Start a VM/ESA GUI program (such as the CMS Desktop).

See the *IBM VM/ESA: Graphical User Interface Facility* for details on how to set up and use the VM/ESA GUI Facility.

Distributed GUI Toolkit (DT): DT is an object-based programming interface that can be used from REXX, C++, C, or Assembler languages through CSL routine calls. DT supports the display of visual controls and elements found in many GUI workstation applications. DT programs can create windows, menu bars, menus, radio buttons, check boxes, entry fields, and list boxes. With DT, programs can also interact with the workstation clipboard, invoke workstation commands, and upload and download files. However, DT leaves the details of a window's appearance to the native GUI. See the *IBM VM/ESA: Distributed Graphical User Interface Toolkit* for more information on using DT.

CMS Desktop: The CMS Desktop is a subset of CMS function that uses DT to let you perform tasks through a graphical user interface. You can start the CMS Desktop by entering the CMSDESK command. The CMS Desktop performs the following functions:

- Manipulates data files on your minidisks and SFS directories and in your reader
- Searches data files on your minidisks and SFS directories
- Maintains your NAMES file
- Edits host files
- Adds and deletes objects from your own desktop
- Sends notes to other VM/ESA users
- Includes integrated online help

New Commands: The following new commands have been added:

CMSDESK
Starts the CMS Desktop.

QUERY WORKSTATION
Displays information about the workstation display address setting.

SET WORKSTATION
Sets the workstation display address.

VM/ESA GUI Facility Enhancements [2.1.0]

A toolbar is now available on the CMS Desktop. For each application on the CMS Desktop, you can choose to display a toolbar provided or you can create your own bitmaps and display them on the toolbar. See the *IBM VM/ESA: Graphical User Interface Facility* for more information.

A new DT routine, DtBitmap, allows you to import bitmaps from a file into an application.

VM/ESA added support for the following workstation operating environments and their appropriate communications product:

- Microsoft Windows with APPC support
- Windows 95 with TCP/IP support

VM/ESA GUI Facility Enhancements [2.2.0]

The VM/ESA GUI Facility is now part of the CMS component.

The VM/ESA GUI Facility now includes the following:

- Windows NT workstation environment support

- 32-bit support for Windows 95 and Windows NT

- Source Help files (IPF files) to allow users to customize their online help information.
- Security enhancements for the workstation agents. You can now identify the user IDs authorized to display GUI windows on your workstation. You can also specify whether you want to display the security window when you start these GUI applications.

See the *IBM VM/ESA: Graphical User Interface Facility* for more information.

The following new DT routines have been added:

DtAdapter03
: Converts the value of a DtBool object to an object of data type DtInt32 or converts the value of a DtInt32 object to an object of data type DtBool.

DtAdapter05
: Converts the value of a DtColorobject to an object of data type DtInt32 or converts the value of a object of data bype DtInt32 to a DtColor object.

DtAdapter06
: Determines the length of an input string.

DtAdapter07
: Manipulates data strings by comparing the values of two strings.

DtAdapter09
: Dynamically creates a character string.

DtAdapter10
: Extracts substrings from a character string.

DtAdapter11
: Finds a string within another string.

DtAdapter12
: Builds a table of strings from a group of individual strings.

DtAdapter13
: Splits a table of strings into a group of individual strings.

DtCounter
: Controls a value by increasing or decreasing the value, setting the value to zero, or setting the value to another value.

DtScrollHorz
: Creates a vertical scroll bar.

DtScrollVert
: Creates a horizontal scroll bar.

See the *IBM VM/ESA: Distributed Graphical User Interface Toolkit* for more information.

Year 2000 API Support [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

VM/ESA APIs that accept date input or provide date output have been changed to support 4-digit years:

Note: Some of these functions may not exist on your current VM/ESA release. For more information about the specific changes, see Part 3, “Compatibility Tables” on page 313.

- CP DIAGNOSE codes:
  - DIAGNOSE code X'00'
  - DIAGNOSE code X'14'
  - DIAGNOSE code X'84'
  - DIAGNOSE code X'BC'

  - DIAGNOSE code X'D8'

- CSL routines:
  - DMSCLBLK
  - DMSCLDBK
  - DMSCLOSE
  - DMSCRDIR
  - DMSCRFIL
  - DMSCROB
  - DMSENUSR
  - DMSERP
  - DMSEXIDI
  - DMSEXIFI
  - DMSEXIST
  - DMSGETDA
  - DMSGETDF
  - DMSGETDI
  - DMSGETDS
  - DMSGETDX
  - DMSOPBLK
  - DMSOPDBK
  - DMSOPEN
  - DMSRDCAT
  - DMSTRUNC

- CMS macros:
  - DIRBUFF
  - EXSBUFF
  - FSSTATE
  - FSTD

- CMS OS simulation macro:
  - TIME

- CMS Pipelines stages:
  - AFTFST
  - FMTFST
  - STATE
  - STATEW

- REXX functions:
  - CMSFLAG
  - DATE
  - DIAG/DIAGRC

- GCS macro:
  - TIME

The following new interfaces are provided:

- DIAGNOSE code X'270' - Pseudo Timer Extended - supports 4-digit years. It replaces DIAGNOSE code X'0C' for newer applications.

- DateTimeSubtract CSL routine performs time format conversions and time zone conversions.

- &FULLDATE and &ISODATE EXEC 2 statements return the date in Coordinated Universal Time (UTC) in the specified 4-digit-year form.

Real System Operation [1.1.5, 1.2.0, 1.2.1]

Clean Start Support [1.1.5, 1.2.0, 1.2.1]

A CLEAN start IPLs the system without attempting to recover spool files and system data files that existed prior to system shutdown. Clean start restores the least amount of the system's environment.

GIVE Command [1.1.5]

The CP GIVE command allows a user or a system resource operator to transfer a dedicated tape drive to another user on the system. In VM/ESA 1.1.5 370 Feature, the operator had to first detach the tape drive specifying the LEAVE option, then enter the ATTACH command to attach the tape drive to the intended receiver. Another user could have attached to the tape drive before the operator could attach it to the intended receiver. With the GIVE command you can be sure the intended receiver gets the tape drive.

INDICATE Command Enhancements [1.1.5, 1.2.0]

The CP INDICATE command supports new operands:

NSS To display information about named saved systems and saved segments that are loaded in the system

SPACES To display information about address spaces

Also, the INDICATE USER command supports a new operand, EXPANDED, that gives a response with additional information, including information about primary spaces, private spaces, shared space, private paging, and shared paging.

SPXTAPE Command [1.1.5, 1.2.0, 1.2.1]

The CP SPXTAPE command dumps standard spool files and system data files from the spooling system onto tape and loads SPXTAPE-format files from tape into the spooling system. Compared with the SPTAPE command, SPXTAPE provides faster performance, reduced tape volume usage, better error handling, and increased flexibility and usability. For example, SPXTAPE:

- Can dump files to, or load files from, multiple tape drives simultaneously
- Can process files that are larger than a single tape volume
- Creates log files to record command status and a history of the files that have been processed
- Attempts to continue processing if it encounters an I/O error
- Skips over standard tape labels

In addition, there are the following important operational differences between SPTAPE and SPXTAPE:

- SPTAPE uses a single real tape drive addressed by its real device number. SPXTAPE uses one or more real tape drives attached as virtual devices to the virtual machine issuing the command.

- The SPTAPE command for ending processing at the completion of the current file is STOP. The SPXTAPE command for the same general function is END.

- The default tape disposition operand for SPTAPE is LEAVE. The default for SPXTAPE is RUN.
- When processing standard spool files, SPTAPE by default selects only files with a hold status of NOHOLD. SPXTAPE by default selects files regardless of their hold status.
- SPXTAPE requires the USER operand if a class D or E user selects standard spool files or system trace files by user ID.
- On SPTAPE DUMP, the MODE *nnnn* operand selects the recording format on the tape drive, which varies by device type. SPXTAPE DUMP uses the default recording format of the tape drive. The MODE operand on SPXTAPE DUMP indicates only whether the hardware compaction feature is to be used (COMP|NOCOMP).
- If a tape contains multiple tape files, each one written by a separate SPTAPE DUMP command, all the tape files can be processed by a single SPTAPE LOAD command. If the tape files were written by separate SPXTAPE DUMP commands, each tape file must be processed by a separate SPXTAPE LOAD command.

Terminal Hang User Control [1.1.5]

Users can have control over certain hang conditions. The following enhancements have been provided:

- End-user breakout capability for hangs which occur during console I/O and DIAGNOSE code `X'A4'` and DIAGNOSE code `X'A8'` synchronous DASD I/O. The following commands allow you to break out of these hangs: IPL, LOGOFF, SYSTEM RESET, and SYSTEM CLEAR.
- New operands on the FORCE command allow the operator to select whether to force the user to logoff or to disconnect. In VM/ESA 1.1.5 370 Feature, the FORCE command always forced the user to logoff.
- There is a new operand, HERE, on the LOGON command. This operand lets a logged-on user disconnect from their current terminal session and reconnect to the terminal where the LOGON HERE was entered.

Warm, Cold, Force Starts Improvement [1.1.5, 1.2.0]

The operator is given the option to continue whenever files will be lost if the IPL continues. This is true for warm, cold, and force starts.

Virtual Machine Operation [1.1.5]

Hardware System Console Simulation [1.1.5]

A new System/370 processor interface allows an operating system to use the hardware system console as an IPL and error recovery console. VM/ESA enables operating systems that exploit this new interface to run in virtual machines. This allows MVS operating systems to remove the requirement for a non-SNA operator console for message exchange during IPL and error recovery.

The VM operator's console (the logon terminal) simulates the hardware system console functions for the guest operating system. Message traffic destined for the hardware system console is intercepted by CP from the guest operating system and displayed at the VM operator's console.

Diagnosis [1.1.5, 1.2.0, 1.2.1, 1.2.2]

Control Unit Reconfiguration [1.1.5, 1.2.0, 1.2.1]

Control Unit Reconfiguration (CUIR) is a hardware serviceability feature supported for 3990 models 6 and 7 storage control units. It allows a service representative to initiate channel reconfiguration requests from the control unit. This frees the operator of having to manually issue reconfiguration commands from the system console.

Minidisk Corruption Detection Enhancement [1.1.5, 1.2.0]

The CMS file system has been enhanced to detect more types of minidisk corruption. Two new error messages, DMS1305T and DMS1306T, help diagnose these file system errors.

POSIX Support [1.1.5, 1.2.0, 1.2.1, 1.2.2]

The following DIAGNOSE codes have been added for POSIX support:

- X'280' - Set POSIX security values
- X'29C' - Set POSIX IDs
- X'2A0' - Query POSIX IDs
- X'2A4' - POSIX process ID (PID) services

Recording of Symptom Records [1.1.5]

Symptom records indicate the state of CP at the time of a dump. In VM/ESA 1.1.5 370 Feature, they are contained only in dumps. Now, CP or a virtual machine can record symptom records in CMS files separately from dumps or error records. In addition, more information is provided in the symptom record to allow better identification of duplicate problems.

A new Dump Viewing Facility command, VIEWSYM, lets you view, compare, and select symptom records from one of these CMS files. (Dump Viewing Facility replaces IPCS.)

Symptom records can be automatically sent to authorized virtual machines, which can retrieve them using the RETRIEVE SYMPTOM utility. Records are sent and retrieved through a new CP system service, *SYMPTOM. The mechanism is similar to the one used by the accounting and EREP virtual machines.

In addition to the Dump Viewing Facility VIEWSYM command, the RETRIEVE SYMPTOM utility, and the *SYMPTOM system service, this support adds the following external interfaces:

- A new operand of the RECORDING command, SYMPTOM
- A new function on the QUERY RECORDING command response, SYMPTOM
- A new Dump Viewing Facility DUMPSCAN subcommand, SYMPTOM
- Checkpointing of symptom records by the SHUTDOWN command
- An HCPSYS macro, SYSSYMP

Note: In a SYSTEM CONFIG file, the SYSTEM_USERIDS statement replaces the SYSSYMP macro.

- A new parameter of the IUCV user directory control statement, *SYMPTOM

DIAGNOSE code X'94' is changed to allow an IBM program or a guest operating system to supply a symptom record to CP, either by itself or in conjunction with a dump.

SPXTAPE Command [1.1.5, 1.2.0, 1.2.1]

The CP SPXTAPE command allows the class G user to save on tape and restore the user's own reader, printer, and punch spool files and the user's own system trace files.

Storage Dump Enhancement [1.1.5, 1.2.0, 1.2.1]

You can use the new SNAPDUMP command to get a new type of CP dump, which is called a snapdump. A snapdump can also be created by the system when it encounters an internal error. See the *VM/ESA: CP Command and Utility Reference* and the *VM/ESA: Diagnosis Guide* for more information.

Trace Enhancements [1.1.5, 1.2.0, 1.2.1]

- **[1.1.5, 1.2.0]** The TRSAVE command supports two new operands, DEFERIO and FRAMES.

  DEFERIO allows the delay of recording:

  – Trace data to DASD until the *traceid* is disabled
  – Trace table data until TRSAVE CP OFF is issued

  FRAMES refers to the number of frames allocated from the system dynamic paging area for enabled DEFERIO *traceids* and for DEFERIO CP system tracing.

- **[1.1.5, 1.2.0]** The TRACERED command of the Dump Viewing Facility provides more diagnostic information when viewing trace data. Also, it supports CMS files as input.

  A new Dump Viewing Facility subcommand, TRSAVE, displays:

  – Enabled TRSOURCE *traceids*
  – Active TRSAVEs for CP from a storage dump

- **[1.1.5, 1.2.0, 1.2.1]** The TRSOURCE command supports four new selectivity options for the TYPE DATA datatrace function: IF, THEN, ELSE, and ENDIF.

- **[1.1.5, 1.2.0, 1.2.1]** The Dump Viewing Facility commands ASID, ACCLIST, DUMPID, and FINDUSER have been enhanced to support snapdumps.

See the *VM/ESA: CP Command and Utility Reference* and the *VM/ESA: Dump Viewing Facility* book for more information about these trace enhancements.

Viewing and Printing CP and CMS Control Blocks [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

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

http://www.ibm.com/s390/vm/pubs/

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

In addition, VM/ESA 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.

Virtual Machine Dump Enhancements [1.1.5]

Multiple virtual machine dumps can be sent to a single file, dumps can be sent in a continuous mode, similar to the continuous mode allowed by spooling commands, and address spaces other than the primary address space can be dumped. Also, virtual machine dumps occupy system data files rather than spool files, reducing processing overhead and DASD space usage.

These enhancements add the following:

- New operands on the VMDUMP command and new parameters on DIAGNOSE code X'94' for:
  - Setting and resetting continuous output mode
  - Closing or purging an open virtual machine dump file
  - Dumping several address spaces
- A new command, QUERY VMDUMP.
- The DUMPLOAD utility can now recognize and handle virtual machine dump files containing the output of multiple VMDUMP requests or containing the output from several address spaces.

Chapter 10. Preinstallation and Installation Changes and Conversion Considerations

This chapter discusses conversion considerations you should think about before installing your new system and changes you may encounter during your installation.

This chapter may contain information for any or all of the conversions documented in this book. For definitions of the conversion identifiers used in this chapter, see “How to Identify the Topics That Apply to Your Conversion” on page 3.

This chapter contains the following sections:

Major Changes in Installation [ALL]

This section describes the major changes in installation.

CD-ROM Install Support [1.1.5, 1.2.0]

The install process supports VM/ESA installation from CD-ROM in addition to tape. The IBM S/370 and S/390 Optical Media Attach/2 (OMA/2) is required to install from a CD-ROM disk. OMA/2 allows a PS/2 with a CD Reader attached to emulate a 3422 tape control unit.

High Level Assembler Support [1.1.5, 1.2.0, 1.2.1, 1.2.2]

The IBM High Level Assembler (HLASM) is the only assembler supported on VM/ESA. Both the Version 1 Release 1 and Version 1 Release 2 HLASM products are supported by VM/ESA commands. HLASM is a co-requisite for VM/ESA and is no longer automatically included on the VM/ESA System DDR.

INSTALL EXEC [1.1.5]

To install VM/ESA, you use an installation process that features an INSTALL exec with a panel interface. You are able to select what components to load and there is flexibility in where you can place those components on your DASD. This installation process minimizes contiguous DASD requirements and DASD device type restrictions.

INSTALL EXEC Enhanced [1.2.0, 1.2.1]

The INSTALL EXEC has been enhanced as follows:

- **[1.2.0]** A minidisk recover option has been added. This option lets you load the contents of a minidisk from the VM/ESA System DDR tape or CD-ROM disk.
- **[1.2.0, 1.2.1]** The LOAD MENU now includes the option of loading SFS and Uppercase English HELP (UCENG).

ITASK EXEC Deleted [1.1.5, 1.2.0]

ITASK is no longer shipped with the VM/ESA product.

New Execs [1.1.5, 1.2.0, 1.2.1, 1.2.2]

The following execs were added in VM/ESA 2.1.0:

- INSTPLAN EXEC selects items to load and the DASD type on which to install.
- INSTIIS EXEC formats and labels your installation DASD and restores the IIS.
- INSTDIR EXEC selects the appropriate directory for your installation DASD type.
- POSTLOAD EXEC performs clean-up tasks depending on the items you have loaded.

New and Enhanced Execs [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

The new INSTPOOL EXEC generates the file pool servers if needed and autologs the server user IDs to start the servers.

The INSTDIR EXEC has been enhanced to generate a directory tailored to your choice of items selected and DASD placement defined.

New and Enhanced Execs [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

The new INSTDEF EXEC customizes CMS, rebuilds CMS, CP, and GCS, and moves selected items to SFS.

The INSTDEF2 EXEC performs clean-up and backup tasks after customizing and rebuilding the nucleus.

Product Tape Installation Procedures Deleted [1.1.5, 1.2.0]

Due to the ease and flexibility of installing with a VM/ESA systems DDR image, the Product Tape installation procedure is no longer available.

Preinstallation Considerations [ALL]

Before you install your new system, you may want to consider:

- **[1.1.5]** CP message command function user modifications may no longer work
- **[1.1.5, 1.2.0, 1.2.1]** Considerations for converting 3390 data to 3390 Model 9
- **[1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]** Considerations for converting your user directory
- **[1.2.0, 1.2.1, 1.2.2]** Considerations for CP Exits

CP Message Function User Modifications May No Longer Work [1.1.5]

VM/ESA has improved support for processing message commands with an installation-wide exit, HCPMSU. HCPMSU is an assemble module that allows customization of the message command functions. User CP modifications based on earlier message-command-processing code no longer work. *VM/ESA: Planning and Administration* describes HCPMSU.

Considerations for Converting 3390 Data to 3390 Model 9 [1.1.5, 1.2.0, 1.2.1]

There are considerations for moving data from other 3390s to a 3390 Model 9. The 3390 Model 9 allocation map is extent-based rather than cylinder-based. All other 3390s are cylinder-based.

If you use the DDR command to copy data from another 3390 to a 3390 Model 9, the source 3390's cylinder-based allocation map is also copied if the DDR copy includes cylinder 0. If this happens, and you are using the device as a CP volume, only the first 4096 cylinders of the 3390 Model 9 are used. Also, if you use DDR to copy data from a 3390 Model 9 to another 3390, the 3390 Model 9's extent-based allocation map is copied if the DDR copy includes cylinder 0. The target DASD will not be supported as a CP volume. To correct this situation, reallocate the target 3390 using the ICKDSF CPVOLUME ALLOCATE command.

Note: The 3390 Model 9 does not support 3380 track compatibility mode and is not recommended for CP system areas. Also, the 3390 Model 9 is not supported as an installation device.

Considerations for Converting Your User Directory [ALL]

Considerations for converting your user directory are:

- Object directory compatibility
- Getting around directory differences
- User directory control statement changes

Object Directory Compatibility [ESA]

Object directory compatibility is supported when the object directory created with the VM/ESA 2.4.0 DIRECTXA MODULE is used by **currently supported releases of CP (except VM/ESA 1.1.5 370 Feature)**. 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 VM/ESA 2.4.0 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 VM/ESA 2.4.0 HCPDIR ASSEMBLE module, which is used to create the DIRECTXA MODULE.

Getting Around Directory Differences [ALL]

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

- **[1.1.5]** Avoid using any directory statements or options that were introduced after your VM/ESA 1.1.5 370 Feature system.
- **[1.2.0, 1.2.1]** Be careful not to include any directory statements or options beyond those that are supported by MIXED directory APARs that have been applied to your system. The following MIXED directory APARs exist:
  - **[1.2.0]** APAR VM55600 for VM/ESA 1.2.1 statements and options
  - **[1.2.0, 1.2.1]** APAR VM58022 for VM/ESA 1.2.2 statements and options

  MIXED directory APARs do not exist for releases after VM/ESA 1.2.2. To include statements or options that were introduced after VM/ESA 1.2.2, you must maintain two source directories. The statements and options that are unsupported by your system should be included only in the source directory for the VM/ESA 2.4.0 system.

- **[Other Unsupported ESA Releases]** 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 system should be included only in the source directory for the VM/ESA 2.4.0 system.
- **[Supported ESA Releases]** You can do either of the following:
  - Use the VM/ESA 2.4.0 DIRECTXA MODULE 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 [ALL]

For a list of control statement changes, refer to the control statements compatibility table (if any) for your conversion. See Part 3, “Compatibility Tables” on page 313.

Considerations for CP Exits [1.2.0, 1.2.1, 1.2.2]

Certain system records which are accessed by *ACCOUNT, *LOGREC, *SYMPTOM and *CONFIG system services are saved in the checkpoint area across IPLs. The format and size of these records changed in VM/ESA 2.4.0. VM/ESA 2.4.0 will automatically convert records from the previous releases to the new release format. However, should you wish to run the previous release level once the records have been converted you will require APAR VM59271 to be applied to the previous release which will convert the records back to a format which can be processed by the previous release. APAR VM59271 is available for VM/ESA 1.2.0, VM/ESA 1.2.1, and VM/ESA 1.2.2.

Installation Considerations [ALL]

This section describes differences you may encounter during your VM/ESA installation. It contains the following topics:

- “Installation Process Overview [ALL]”
- “Preparing the CP-Owned DASD [1.1.5]” on page 214
- “Alternate CP Nucleus Support Differences [1.1.5]” on page 216
- “Converting System Generation Files [1.1.5]” on page 216
- “Configuring I/O [1.1.5]” on page 217
- “Re-IMLing the Processor to ESA/370 or ESA/390 Mode [1.1.5]” on page 226
- “Installing the Printer Image Library [1.1.5]” on page 227
- “Installing GCS [1.1.5]” on page 227
- “Connectivity Conversion Considerations [1.1.5]” on page 227
- “Installing CMS [1.1.5]” on page 229
- “Building Saved Segments [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]” on page 230

Installation Process Overview [ALL]

VM/ESA Version 2 Release 4.0 features an automated installation process that uses panel interface driven execs. You are able to select the items to load and there is flexibility in where you can place the items on your DASD. This installation process minimizes contiguous DASD requirements and DASD type restrictions.

This process offers a choice of three procedures for installing your VM/ESA system. All three use the VM/ESA System DDR that includes the Initial Installation System (IIS). The IIS is a functional VM system used during installation of VM/ESA. You will restore the IIS and then continue to load minidisks from the VM/ESA System DDR prepared for your particular DASD type. The installation procedures allow for:

- Flexibility in which items to load
- Mixed DASD support
- Automated directory build using one of the following three methods:
  - The same DASD type and density for all installation packs. Requires the entire pack. The pack labels (240W01, ...) are defined by the installation execs, and the placement of minidisks is generated by the installation exec.
  - Any combination of DASD types/models supported by your installation (3380, 3390, 9345, FBA). The extents used on each pack may be limited by the customer, the pack labels are defined by the customers, and the placement of items is generated by the installation execs.
  - Any combination of DASD types/models supported by your installation (3380, 3390, 9345, FBA). The extents used on each pack may be limited by the customer, the pack labels are defined by the customers, and the placement of items is controlled by the customer.
- Layout of your system during installation planning.

The following briefly describes the three procedures:

- **Procedure 1** is a first level installation. This means there is no VM system running on the processor or LPAR on which you are installing VM/ESA Version 2 Release 4.0.

  You must use this procedure if:

  - No VM system is running in the processor or LPAR on which you are installing VM/ESA Version 2 Release 4.0
  - Or your VM system does not support the DASD on which you are installing VM/ESA Version 2 Release 4.0.
- **Procedure 2** is a second level installation. This means there is a VM system running on the processor or LPAR on which you are installing VM/ESA Version 2 Release 4.0, and you will install VM/ESA Version 2 Release 4.0 from a user ID on that system.

  You may use this procedure if:

  - You have an ESA VM/ESA system running on the processor or LPAR on which you are installing VM/ESA Version 2 Release 4.0
  - And your VM system supports the DASD on which you are installing VM/ESA Version 2 Release 4.0.

- **Procedure 3** is a mixture of first and second level installations. This procedure begins the installation in a virtual machine on your VM system (second level) and completes the installation first level.

  You may use this procedure if:

  – You have a 370 VM/ESA system running on the processor or LPAR on which you are installing VM/ESA Version 2 Release 4.0
  – And your current system supports the DASD on which you are installing VM/ESA Version 2 Release 4.0

Preparing the CP-Owned DASD [1.1.5]

ESCON consideration: When you are doing the conversion, you must be careful not to place CP-owned 3380 DASD volumes on channels that use Enterprise Systems Connection Architecture (ESCON) unless the volumes have been formatted without filler records using the Device Support Facility program (ICKDSF). When you format a DASD volume with ICKDSF, you receive messages indicating whether you are formatting with or without filler records. You can use the NFIL option of ICKDSF to ensure that your DASD is formatted without filler records. Any other ECKD capable DASD (such as 3390s and 9345s) are always formatted without filler records.

Also note that any minidisks on ESCON channels which are being mapped to virtual storage (such as SFS DIRCONTROL directories or SFS minidisks participating in virtual machine data space mappings) must be formatted without filler records. This is because CP paging routines are used to perform I/O to mapped minidisks. If mapped minidisks are formatted with filler records, CP paging routines forces CKD channel program usage. CKD channel program usage causes severe performance degradation on ESCON channels. You can use the CMS FORMAT command to format CMS minidisks without filler records.

Non-ESCON considerations: You must reallocate, but not reformat, your paging and spooling packs, except for cylinder 0 or pages 0-3. You must reformat cylinder 0 or pages 0-3.

Note: When you reformat cylinder 0 or pages 0-3, you lose any information in cylinder 0 or pages 0-3, such as pointers to the nucleus and the user directory. Note also that VM/ESA reserves pages 0-3 of FBA DASD for system use. In VM/ESA 1.1.5 370 Feature, it reserved pages 0-1.

There is no longer a stand-alone format/allocate program in VM/ESA as there was in VM/ESA 1.1.5 370 Feature. Instead, you can:

- Use the stand-alone Device Support Facilities (ICKDSF) program, Release 11 or later for CKD or ECKD, and Release 14 or later for FBA. In fact, using the ICKDSF is the recommended method for allocating and reformatting DASD volumes for CP used. See *ICKDSF User's Guide and Reference* for more information on using ICKDSF.

- Use the CPFMTXA utility of VM/ESA See *VM/ESA: CP Command and Utility Reference* for more information.

CPFMTXA utility does not let you query allocation on an R/O volume, and it requires you to have write access to a volume in order to read or write the allocation map. You can load the CPFMTXA utility and the ICKDSF program from the VM/ESA IIS onto your VM/ESA 1.1.5 370 Feature system to reformat and reallocate DASD for use on your new VM/ESA system.

The following table compares the VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0 options for allocation space. You should not allocate cylinder 0 as TDSK.

Table 18. Comparison of VM/ESA 1.1.5 370 Feature Allocation Options to VM/ESA 2.4.0

VM/ESA 1.1.5 370 Feature Allocation Option	**VM/ESA 2.4.0 Counterpart**	**Differences**
DRCT	Same	No differences.
DUMP	SPOL	In the new VM/ESA, you use SPOL for dump space in addition to spooling and paging activity.
END	Same	No differences.
OVRD	———	Override files are stored as system data files in the new VM/ESA rather than on DASD. Therefore, the OVRD allocation is unsupported.
PAGE	Same	In VM/ESA 1.1.5 370 Feature, you usually use PAGE (plus further definition by the SYSPAG macro) for paging and swapping; you usually use TEMP for spooling and overflow paging. In the new VM/ESA, you use PAGE for all paging activity; you use SPOL only for the overflow paging.
PERM	Same	Using CPFMTXA, the default allocation type is TEMP under VM/ESA 1.1.5 370 Feature but PERM under the new VM/ESA.
TDSK	Same	No differences.
TEMP	SPOL	In the new VM/ESA, you use SPOL for spooling and overflow paging activity (and also for dump space).

If You Are Ready to Move to the New VM/ESA System

You must reformat cylinder 0 of devices (and reallocate the devices) that currently have a System/370-style allocation bit map if you intend to use that volume as a CP-owned volume under the new VM/ESA. You do not need to reformat cylinder 0 and reallocate DASDs that are not CP-owned. Also, you do not have to reformat and reallocate pages 0-3 of FBA DASD. They can be used in the same format on the new VM/ESA.

Note: When you reformat cylinder 0, you lose any information in cylinder 0, such as pointers to the nucleus and the user directory.

Because you cannot reformat and reallocate paging and spooling packs to which you are currently attached, you must:

1. Shut down the system.
2. Re-IPL it with the CP-owned DASDs that you plan to reformat offline.
3. Attach the DASDs to a virtual machine.
4. Reformat cylinder 0 or pages 0-3 and reallocate. Note that the new VM/ESA reserves pages 0-3 of FBA DASD for system use. In VM/ESA 1.1.5 370 Feature, it reserved pages 0-1.

What If I Forgot to Reformat Cylinder 0 or Pages 0-3 and Reallocate CP-Owned Packs?

If you forgot to reformat cylinder 0 or pages 0-3 and reallocate your CP-owned DASD, you will not receive an error message telling you that the DASD is formatted or allocated incorrectly. The DASD will be brought up online but will not be CP-owned (even if some or all of the DASD is allocated as PERM).

If you enter a command to attach the DASD, such as:

```
cp att 453 system specx5
```

you receive the following error message:

```
HCPATS128E DASD 0453 ERROR READING ALLOCATION RECORD
```

However, after reformatting cylinder 0 or pages 0-3 and reallocating the pack, you will be able to attach the pack to the system.

If the DASD is not CP-owned, you do not need to reformat cylinder 0 or pages 0-3 and reallocate. You may use the DASD in the same format as you did on VM/ESA 1.1.5 370 Feature.

Alternate CP Nucleus Support Differences [1.1.5]

The alternate CP nucleus support in VM/ESA 2.4.0 differs from that in VM/ESA 1.1.5 370 Feature, as follows:

- In VM/ESA 1.1.5 370 Feature, you got different user class restructure override data and EREP data when you IPLed an alternate nucleus than when you IPLed the primary nucleus. In VM/ESA 2.4.0, you get the same data because the data is not saved to specially allocated areas of DASD; the user class restructure data resides in system data files and the EREP data is sent to a virtual machine.
- VM/ESA 1.1.5 370 Feature abended if a valid directory was not found; you had to recover the directory while the system was not running. In VM/ESA 2.4.0, operation continues if a valid directory is not found; you have the choice of attempting to recover the directory while the system is running or after shutting it down. See “What If I Cannot IPL Because of a Problem with the User Directory? [ALL]” on page 112 for more information.

Although alternate nucleus support exists in VM/ESA 2.4.0, you should be aware that a CP nucleus can also be stored as a module. Many different CP modules can be stored on one or more CMS minidisks, and the module to be loaded can be identified when the system is IPLed. This makes the alternate nucleus support obsolete. For more information, see the *VM/ESA: Planning and Administration*.

Converting System Generation Files [1.1.5]

There are four system generation files you need to convert when moving to VM/ESA 2.4.0: DMKSYS, DMKRIO, DMKBOX, and DMKSNT. A new system configuration file named SYSTEM CONFIG replaces DMKSYS (or SYS OVERRIDE) and DMKRIO (or RIO OVERRIDE). A file named LOGO CONFIG is used in place of DMKBOX (or BOX OVERRIDE). CP commands replace DMKSNT (or SNT OVERRIDE).

During installation, you can choose to exploit the new CP system configuration files immediately, or you can choose to convert your DMKSYS, DMKRIO, and DMKBOX files to HCPSYS, HCPRIO, and HCPBOX files. IBM recommends using the new CP system configuration files, as described in "Converting to System Configuration Files from HCPRIO and HCPSYS [ESA]" on page 79. Note that you still need HCPRIO and HCPSYS files with the following entries:

System File	Minimum Entry
HCPSYS	SYSEND
HCPRIO	RIOGEN CONS=DYNAMIC

HCPSYS and HCPRIO files with these entries are shipped on the VM/ESA System DDR tapes. They are also included in the CP loadlist and should not be removed.

Note: If you are using an HCPRIO other than the one supplied by IBM, HCPRIO entries are also needed for dedicated devices of V=R guests that use V=R recovery.

Converting DMKSNT

VM/ESA 2.4.0 does not have an equivalent system definition file for the systems name table. Instead, you generate saved systems, saved segments, and image libraries using CP commands. As a result, converting DMKSNT statements to the equivalent VM/ESA 2.4.0 commands (see Table 19) is an administrative task rather than an installation task. This is discussed in "Saved Segments and Named Saved Systems Differences [1.1.5]" on page 254.

Table 19. Defining Saved Systems, Saved Segments, and Image Libraries

DMKSNT Macro	VM/ESA 2.4.0 Counterpart
NAMESYS	DEFSYS and DEFSEG commands.
NAMENCP	Not supported.
NAME3800	IMAGELIB and IMAGEMOD utilities.
NAMELANG	NAMELANG is not supported. However, the function is now included in the CMS LANGGEN command.

Configuring I/O [1.1.5]

This section describes the major I/O differences between System/370 architecture and ESA/370 or ESA/390 architecture. Also discussed is how to create an IOCDS, recommendations for configuring I/O, and vary channel path support.

How System/370 I/O Differs from VM/ESA 2.4.0 I/O

One of the major differences between System/370 architecture and ESA/370 or ESA/390 architecture is in the way I/O is handled. System/370 uses channels. ESA/370 and ESA/390 use a channel subsystem, the same channel subsystem first used by 370-XA architecture.

The main differences between System/370 channels and the ESA/370 or ESA/390 channel subsystem are:

- ESA/370 and ESA/390 have Enterprise Systems Connection Architecture (ESCON). For information about ESCON, see *Introduction to Enterprise Systems Connections*.
- The channel subsystem is designed so all processors in the system can access all the devices attached to the system.

- In ESA/370 and ESA/390, hardware handles path management. In System/370, software handled path management.
- ESA/370 and ESA/390 device numbers do not need to correspond to the physical attachment of the I/O device. (For backout purposes, you should make the VM/ESA 2.4.0 device numbers correspond to the VM/ESA 1.1.5 370 Feature addresses.)

  In System/370, you relate devices to control units and channels. For instance, I/O set up for device number 342 would be as shown in Figure 22.

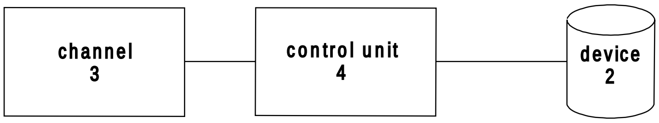


Figure 22. System/370 Rationale for Device Numbers

  In ESA/370 and ESA/390, you relate devices to subchannels. Because each subchannel is uniquely associated with one I/O device and that I/O device is uniquely associated with that one subchannel, the subchannel becomes the addressable unit.

The channel subsystem is more efficient than the System/370 channels because it moves I/O management functions out of the Control Program and into the hardware. This makes your job easier, because you no longer need to define all the paths to a device in the system definition file (that is, DMKRIO or HCPRIO) or in the SYSTEM CONFIG file.

Figure 23 on page 219 and Figure 24 on page 220 provide a conceptual overview of the differences.

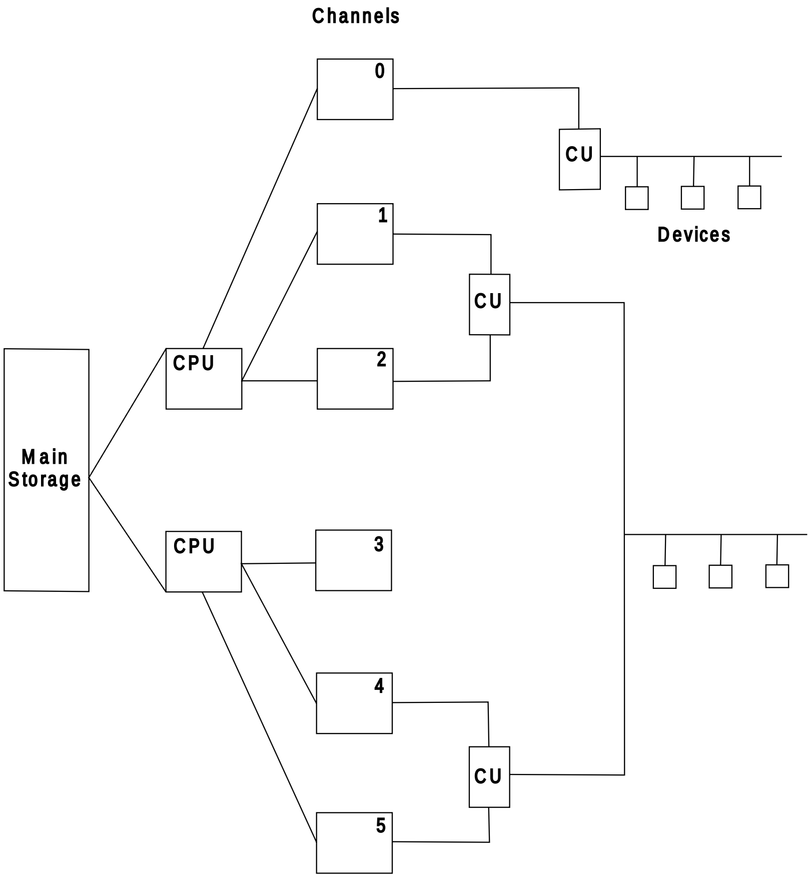


Figure 23. System/370 I/O Structure

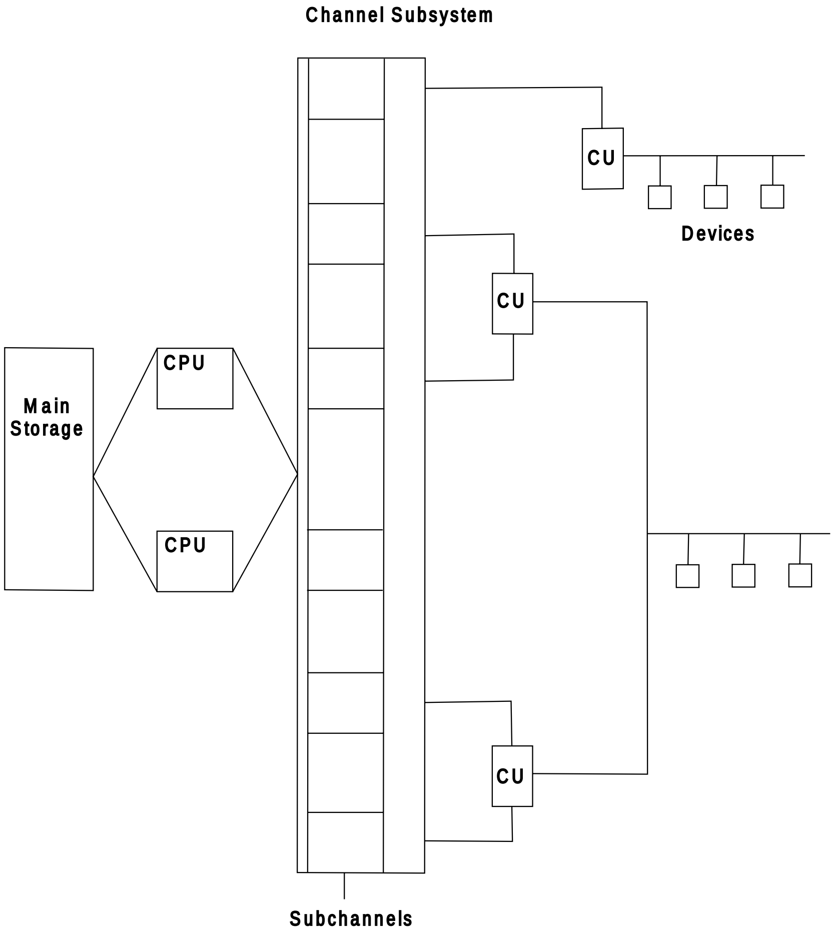


Figure 24. ESA/370 and ESA/390 I/O Structure

Consideration for Creating an IOCDS for Old System and New System

If you are running VM/ESA 1.1.5 370 Feature on a 308X, 3090, or ES/9000, you must have defined the real I/O configuration to the hardware. Using the Input/Output Configuration Program (IOCP), you would have created an input/output configuration data set (IOCDS). VM/ESA 2.4.0 requires an IOCDS for ES/3090 and ES/9000 processors.

You can use the same IOCDS for VM/ESA 2.4.0 that you use for VM/ESA 1.1.5 370 Feature—unless you defined your I/O in an unconventional manner in VM/ESA 1.1.5 370 Feature. For instance, if you used expedient means in VM/ESA 1.1.5 370 Feature, such as assigning two addresses to the same device or defining each path to a control unit as a separate control unit (so you could share dedicated devices or attach a single tape unit to several virtual machine guests), you have to create a new IOCDS. In VM/ESA 2.4.0, the same expedient means may cause the system to hang or data integrity to be lost.

Even if your current IOCDS does not have these problems, you may benefit from reconfiguring your I/O. During the conversion period, however, it is important to have an IOCDS that can be used in 370, XA, and ESA virtual machines. Therefore, you might consider reconfiguring your I/O in a manner that can be used for both the VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0 systems.

For an example of converting your IOCP, see “Example of Converting Your IOCP” on page 222. For an example of using the same IOCP in VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0, but converting from DMKRIO to SYSTEM CONFIG, see “Example of IOCP and DMKRIO Files for System/370” on page 224.

Recommendations for Configuring the I/O

You may want to consider the following:

- When configuring the I/O for the VM/ESA 2.4.0 system, ensure that the I/O configuration described in IOCDS matches the configuration described in the SYSTEM CONFIG file (for devices that need to be defined) or is sensed by CP. (Note that the SYSTEM CONFIG file is similar to DMKRIO.) There is a one-to-one correspondence between real I/O devices, IOCDS IODEVICE definitions, and RDEVICE statements and devices sensed by CP. For every real device there must be one IODEVICE definition. For every IODEVICE definition, CP must generate the real device block dynamically or there must be an RDEVICE statement.

  At system initialization, CP sends warning messages to the primary system operator about devices in the CP_OWNED list or USER_VOLUME_LIST it finds offline. CP does this by checking the SYSTEM CONFIG file (and any definitions in the HPCRIO file) and sensing the devices dynamically. If you have assured that SYSTEM CONFIG and the IOCDS are compatible, the primary system operator receives warning messages only for devices that are intended to be online but were inadvertently left offline.

- To make it easier to convert from your VM/ESA 1.1.5 370 Feature system to your VM/ESA 2.4.0 system, you should assign the same device numbers in your VM/ESA 2.4.0 system that you use in your VM/ESA 1.1.5 370 Feature system.

- Splitting DASD strings between two different control units (defining the same devices as primary to one control unit and alternate to the other) is not necessary and can cause problems.

- When running a System/370 operating system as a guest on VM/ESA 2.4.0, you do not need to define alternate channels and alternate control units.

- See “Converting DMKRIO and DMKSYS Macros to SYSTEM CONFIG File Statements” on page 348 for more information on converting DMKRIO to SYSTEM CONFIG statements.

Example of Converting Your IOCP

Figure 25 is an example of a System/370 configuration:

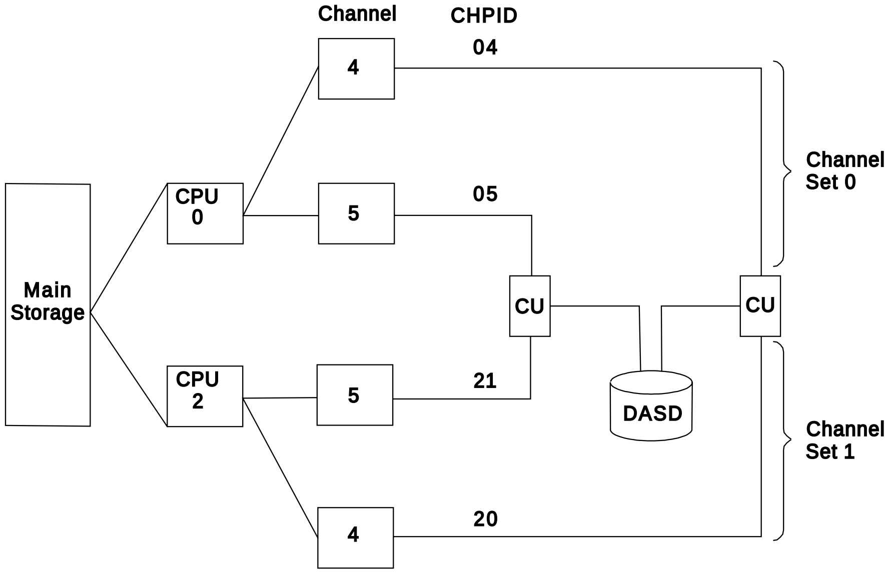


Figure 25. Example of a System/370 I/O Configuration

The IOCP that defines the configuration in Figure 25 follows:

```
CHPID     PATH=((04,4,0)),TYPE=BL
CHPID     PATH=((05,5,0)),TYPE=BL
CHPID     PATH=((20,4,1)),TYPE=BL
CHPID     PATH=((21,5,1)),TYPE=BL
CNTLUNIT  CUNUMBR=042,PATH=(04,20),SHARED=N,UNIT=3880,
      PROTOCL=S,UNITADD=((20,32))
CNTLUNIT  CUNUMBR=052,PATH=(05,21),SHARED=N,UNIT=3880,
      PROTOCL=S,UNITADD=((20,32))
IODEVICE  ADDRESS=(420,32),CUNUMBR=(042,052),UNIT=3380
```

Figure 26 on page 223 is an example of a ESA/370 or ESA/390 configuration:

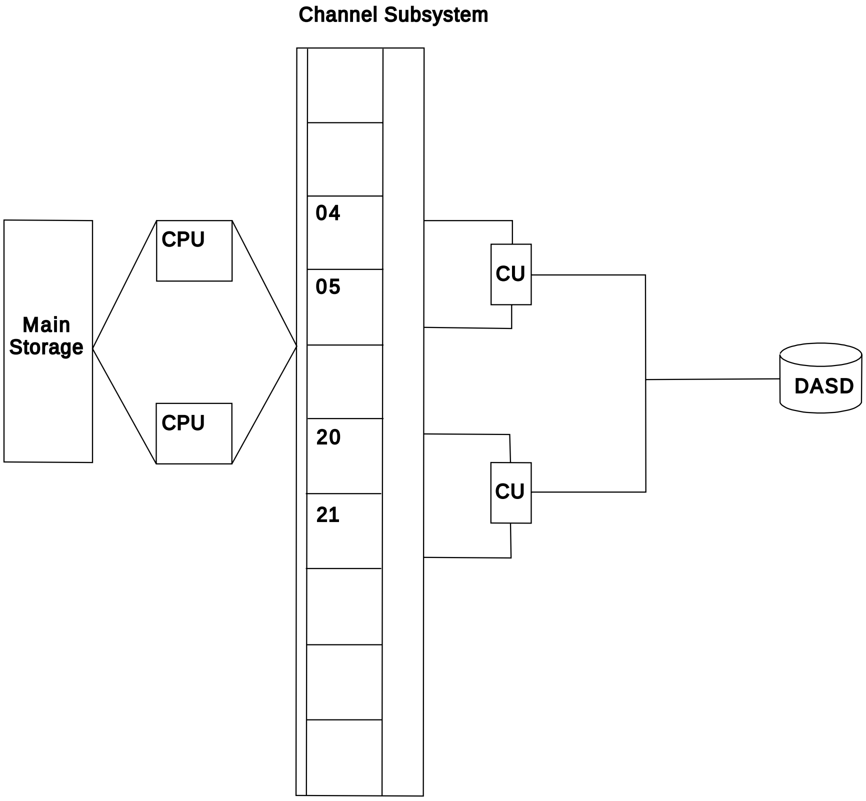


Figure 26. Example of a ESA/370 or ESA/390 I/O Configuration

The IOCP that defines the configuration in Figure 26 follows:

```
CHPID     PATH=((04)),TYPE=BL
CHPID     PATH=((05)),TYPE=BL
CHPID     PATH=((20)),TYPE=BL
CHPID     PATH=((21)),TYPE=BL
CNTLUNIT  CUNUMBR=042,PATH=(04,20),SHARED=N,UNIT=3880,
      PROTOCL=S,UNITADD=((20,32))
CNTLUNIT  CUNUMBR=052,PATH=(05,21),SHARED=N,UNIT=3880,
      PROTOCL=S,UNITADD=((20,32))
IODEVICE  ADDRESS=(420,32),CUNUMBR=(042,052),UNIT=3380
```

Example of IOCP and DMKRIO Files for System/370

Figure 27 is an example of a System/370 I/O configuration.

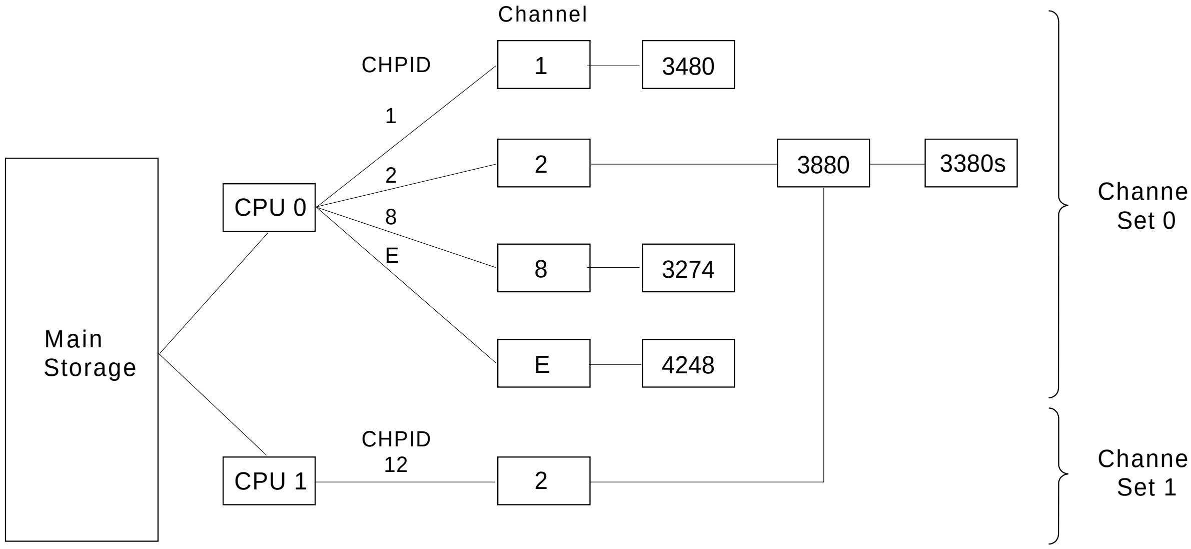


Figure 27. Example of a System/370 I/O Configuration

IOCP and DMKRIO files that define the configuration in Figure 27 follow. Note that the same IOCP can be used in VM/ESA 2.4.0 as shown in “Example of IOCP and SYSTEM CONFIG Files for ESA/370 or ESA/390” on page 225.

IOCP file:

```
CHPID PATH=((01,1,0)),TYPE=BL
CHPID PATH=((02,2,0),(12,2,1)),TYPE=BL
CHPID PATH=((08,8,0)),TYPE=BL
CHPID PATH=((0E,E,0)),TYPE=BL
CNTLUNIT CUNUMBR=100,PATH=(01),SHARED=N,UNIT=3480,
      UNITADD=((00,16)),PROTOCL=S
CNTLUNIT CUNUMBR=280,PATH=(02,12),SHARED=N,UNIT=3880,
      UNITADD=((80,32)),PROTOCL=S
CNTLUNIT CUNUMBR=8E0,PATH=(08),SHARED=YB,UNIT=3274,
      UNITADD=((E0,32)),PROTOCL=D
CNTLUNIT CUNUMBR=E0E,PATH=(0E),SHARED=N,UNIT=4248,
      UNITADD=((0E)),PROTOCL=D
IODEVICE ADDRESS=(100,16),CUNUMBR=100,UNIT=3480
IODEVICE ADDRESS=(280,32),CUNUMBR=280,UNIT=3380
IODEVICE ADDRESS=(8E0,7),CUNUMBR=8E0,UNIT=3279,
      MODEL=3
IODEVICE ADDRESS=(8E7,1),CUNUMBR=8E0,UNIT=3287,
      PATH=08
IODEVICE ADDRESS=(E0E),CUNUMBR=E0E,UNIT=4248,STADET=Y,
      PATH=0E
```

DMKRIO file:

```
RDEVICE ADDRESS=(100,16),DEVTYPE=3480
RDEVICE ADDRESS=(280,32),DEVTYPE=3380
RDEVICE ADDRESS=(8E0,7),DEVTYPE=3279
RDEVICE ADDRESS=(8E7,1),DEVTYPE=3287
RDEVICE ADDRESS=(E0E,1),DEVTYPE=4248,CLASS=(A)
```

*

```
         RCTLUNIT ADDRESS=100,CUTYPE=3480
         RCTLUNIT ADDRESS=280,CUTYPE=3880
         RCTLUNIT ADDRESS=8E0,CUTYPE=3274
         RCTLUNIT ADDRESS=E0E,CUTYPE=4248
*
         RCHANNEL ADDRESS=1,CHTYPE=BLKMPXR
         RCHANNEL ADDRESS=2,CHTYPE=BLKMPXR
         RCHANNEL ADDRESS=8,CHTYPE=BLKMPXR
         RCHANNEL ADDRESS=E,CHTYPE=BLKMPXR
```

In the IOCP and DMKRIO files you can see that:

- Each channel must be assigned to a specific processor.
- Each piece of hardware must have its own definition.
- There can be multiple paths to a DASD through multiple address ranges. In this example, there are two paths to one DASD through two address ranges.

Example of IOCP and SYSTEM CONFIG Files for ESA/370 or ESA/390

A corresponding ESA/370 or ESA/390 I/O configuration would look as shown in Figure 28.

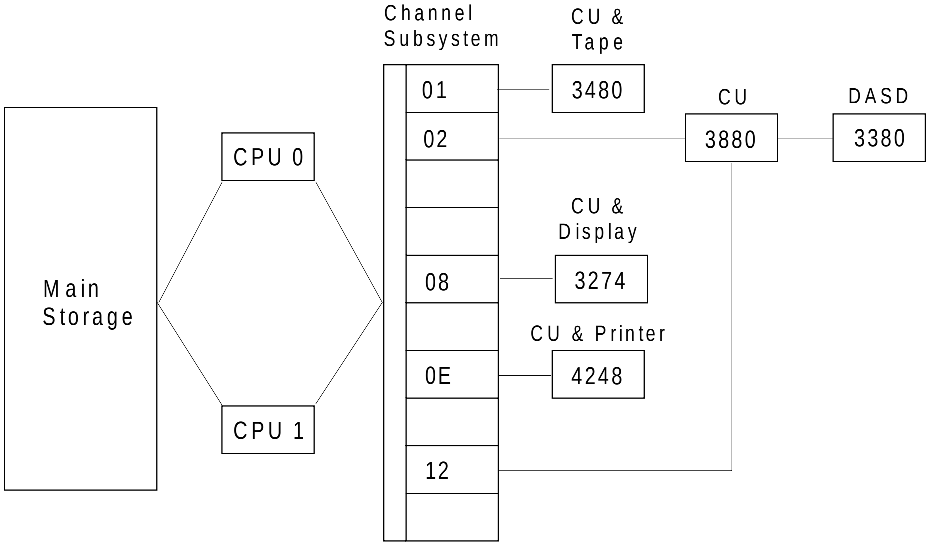


Figure 28. Example of an ESA/370 or ESA/390 I/O Configuration

In this example, the same IOCP file can be used in VM/ESA 2.4.0. See the IOCP file in “Example of IOCP and DMKRIO Files for System/370” on page 224 for details. The corresponding VM/ESA 2.4.0 SYSTEM CONFIG files would look as follows:

```
/*******************************************************
*                                                      *
*   Sample Rdevice Config File for 9121 Sample System  *
*                                                      *
*     The devices contained within this config file    *
*  map those devices that were in the DMKRIO file.     *
*                                                      *
*     Please note that the 3480s, 3380s and 3274       *
*  device definitions can be determined dynamically    *
*  so no additional device definitions are needed      *
*  for them.  The 4248 Printer device definition can   *
*  be determined dynamically but the class and fcb     *
*  information are needed to start the printer by CP.  *
*                                                      *
*******************************************************/

   Rdevice E0E Type Impact_Printer  Class A FCB FCB8
```

In the IOCP and SYSTEM CONFIG files you can see that:

- Each channel can be accessed by any processor.
- There can be a logical control unit made up of two real control units.
- There can be multiple paths to a DASD through a single address range. In this example, there are two paths to one DASD through only one address range.

Vary Channel Path Support

VM/ESA 2.4.0 provides commands to vary channel paths online and offline logically. The VARY PATH command allows the system operator to logically make a path to one or more devices available or unavailable for system use. Without this support, you would have to reload (IPL) the entire VM/ESA 2.4.0 system or vary *all* devices offline, then online, just to gain the one or more paths desired. The ability to vary channel paths allows you to isolate control units for maintenance or to dynamically add a new channel path without disrupting the entire system's I/O configuration.

This support also includes commands that allow the system operator to obtain path allocation information. The QUERY PATHS command displays the logical online or offline status of all paths leading to a specified device. The QUERY CHPID command allows the operator to find all the devices accessible through a specified channel path and the logical online and offline status of each device on that path.

Re-IMLing the Processor to ESA/370 or ESA/390 Mode [1.1.5]

Before you IPL VM/ESA 2.4.0, you must re-IML the processor to ESA/370 or ESA/390 mode, depending on the processor.

As mentioned previously, you can use the same IOCP files (IOCDS). However, to take advantage of the ESA/370 or ESA/390 architecture, you will most likely want to redefine your I/O and recreate IOCP files.

Installing the Printer Image Library [1.1.5]

VM/ESA 2.4.0 handles the printer image library differently from VM/ESA 1.1.5 370 Feature, as shown in the following table.

Table 20. Differences Installing the Printer Image Library

VM/ESA 1.1.5 370 Feature	VM/ESA 2.4.0
You need an image library for 3800 printers. You do not need an image library for impact printers.	You need an image library for both 3800 printers and impact printers.
After creating or updating image libraries, you must re-IPL the system.	You can create or update image libraries dynamically under CMS; you do not need to re-IPL.
Uses assemble files: DMKFCB ASSEMBLE DMKUCS ASSEMBLE DMKUCB ASSEMBLE DMKUCC ASSEMBLE DMKPIA ASSEMBLE DMKPIB ASSEMBLE.	Uses image libraries. IMAG*xxxx* IMG is the default image library for printer type *xxxx*. Sample files are provided on the VM/ESA System DDR tapes for creating image libraries. You create image libraries for: • 3800 printers using text decks and control files • Impact printers using assemble files, text decks, and control files.
For manipulating image libraries, use the GENIMAGE, IMAGELIB, and IMAGEMOD commands, and the NAME3800 macro.	For manipulating image libraries, use the GENIMAGE, IMAGELIB, and IMAGEMOD utilities, and the QUERY IMG and PURGE IMG commands.

Installing GCS [1.1.5]

When installing GCS, you no longer have to define the GCS saved system in DMKSNT. GCS is automatically defined and saved for you when you IPL it.

Connectivity Conversion Considerations [1.1.5]

This section describes conversion considerations for GCS and SNA applications. It also describes changes you may have to consider if you are using a TSAF collection.

GCS and SNA Applications (Such As VTAM)

As in VM/ESA 1.1.5 370 Feature, you can use the GCS component to implement a native SNA communications network. The differences are found in:

- Storage allocation for VSAM and BAM
- Restricted access
- Saved segment definition
- Virtual machine modes
- SNA applications

Also, GCS now exploits XA architecture, which helps relieve virtual storage constaints.

Storage Allocation for VSAM and BAM: In VM/ESA 1.1.5 370 Feature, the system reserved storage for the VSAM and BAM segments, thus requiring virtual machine size to be defined below the VSAM and BAM segments. In VM/ESA 2.4.0, this requirement no longer exists. Storage is reserved or gets allocated dynamically.

In VM/ESA 2.4.0, user IDs of virtual machines requiring storage for VSAM and BAM can be specified using the GCS GROUP exec during system installation. If the user ID was specified, storage is reserved.

After the IPL, when VSAM services are requested, GCS checks if storage is already reserved. If storage was not reserved, GCS checks whether storage is available. If storage was reserved or is available, the segment is loaded. If storage is not available, a message is issued requesting that the virtual machine re-IPL. When the virtual machine re-IPLs, the storage is reserved so the segment can be loaded when VSAM is requested.

Also, CMSBAM and CMSVSAM must be placed in a segment space. In VM/ESA 1.1.5 370 Feature, CMSBAM and CMSVSAM could be placed in DCSSs. See "Saved Segments and Named Saved Systems Differences [1.1.5]" on page 254 for a description of saved segments in VM/ESA 2.4.0.

Restricted Access: In VM/ESA 1.1.5 370 Feature, a virtual machine accesses the GCS supervisor by a link (the LINK command or LINK user directory control statement) to the GCS system disk. In VM/ESA 2.4.0, you control this access with the GCS GROUP exec and by the NAMESAVE user directory control statement. A new question in the GROUP exec asks if you want the GCS named saved system to be restricted. The default is that it is restricted. The NAMESAVE user directory control statement should be placed in the directories of virtual machines that are allowed to access the GCS named saved system.

Saved Segment Definition: In VM/ESA 1.1.5 370 Feature, you used the DMKSNT file to define the name and location of SNA application saved segments (such as for VTAM). In VM/ESA 2.4.0, this is done automatically.

Virtual Machine Modes: In VM/ESA 2.4.0, your GCS system must run in an XA, ESA, or XC virtual machine. The type of virtual machine in which your GCS system is allowed to run depends on what type of virtual machine you were in when you IPLed the recovery server. The following table shows where GCS can run based on the type of virtual machine that you were in when you IPLed the recovery server.

Virtual machine at Recovery Server IPL time:	All virtual machines in the group must be:
XA	XA, ESA, XC
ESA	XA, ESA, XC
XC	XA, ESA, XC

If you IPLed the recovery server in an XA, ESA, or XC virtual machine, then your GCS system cannot run in 370, and all virtual machines in the group must be XA, ESA, or XC.

Also, you receive message 134I (`*sysname* has *nnnnn* Kb of available common free storage above/below the 16 Mb line`) when you IPL the recovery machine. In VM/ESA 1.1.5 370 Feature, you received this message when you IPLed the card reader to build and save your system.

GROUP Exec: You must run the GROUP exec and answer all questions and go through all panels. In most cases, if you take the defaults, you remain compatible with your old release.

SNA Applications: SNA applications running in a GCS group must run in the same type of virtual machine. Applications that exploit or tolerate XA can run with GCS in an XA virtual machine. VTAM is an example of an SNA application that exploits 370-XA architecture.

Refer to the GCS commands and macros compatibility table for your conversion to see if there are any changes. See Part 3, “Compatibility Tables” on page 313.

TSAF and APPC

The following systems can participate in a TSAF collection:

- VM/SP Release 5
- VM/SP Release 6
- VM/SP HPO Release 5
- VM/SP HPO Release 6
- All VM/ESA releases.

There is one TSAF-related difference in VM/ESA 2.4.0 CP. In VM/ESA 1.1.5 370 Feature, the *CRM system service has a limit of 200 global resources and gateways per node. In VM/ESA 2.4.0, the limit is 500 global resources and gateways per node, and 65535 local resources per node.

Installing CMS [1.1.5]

This section includes information about such things as installing the CMS named saved system and changes to the CMS nucleus.

Installing the CMS Named Saved System

In VM/ESA 2.4.0, CMS is installed as a single named saved system. The SAMPNSS EXEC, used during installation, uses the DEFSYS command with the MACHMODE option to specify that the CMS named saved system can be IPLed from an XA or an XC virtual machine.

Note: The SAMPNSS EXEC did not exist in VM/ESA 1.1.5 370 Feature.

Back up the CMS named saved system on tape. Because CP uses system spooling space to store named saved systems and because there is always the possibility that you will not be able to recover spooling space after a CP abnormal termination, you should always keep backup copies of named saved systems on tape. The CP SPTAPE command enables you to do this. If you IPL VM/ESA 2.4.0 with a cold start, named saved systems are not purged.

CMS Nucleus Changes

The CMS nucleus in VM/ESA 2.4.0 is substantially larger than in VM/ESA 1.1.5 370 Feature and extends above the 16MB line. Figure 29 shows the location of the VM/ESA 2.4.0 CMS nucleus.

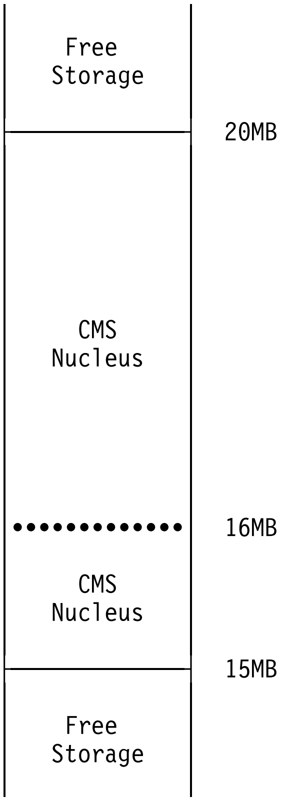


Figure 29. VM/ESA 2.4.0 CMS Nucleus

If you want to IPL CMS from a DASD address rather than from a named segment, such as IPL 190, then you have to define your virtual machine storage to a minimum of 20MB.

Using the CMS Component from the 370 Feature

You can use the VM/ESA 1.1.5 370 Feature CMS instead of the CMS component from VM/ESA 2.4.0 so that your conversion takes place in stages; first to the new CP component, and then to the new CMS component.

Note: VM/ESA 2.4.0 does not support multiple or back levels of CP.

See “Multiple Levels of CMS on VM/ESA [ALL]” on page 169 for more considerations.

Building Saved Segments [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

In VM/ESA 2.4.0, saved segments are preinstalled on the VM/ESA system DDR tapes or CD-ROM. Also, installation of saved segments has been automated as a function of the VMSES/E component. See “Saved Segments and Named Saved Systems Differences [1.1.5]” on page 254 for a description of saved segments in VM/ESA 2.4.0.

Default installation information is supplied for the saved segments such as the following:

- CMSBAM
- CMSDOS
- CMSFILES
- CMSPIPES
- CMSVMLIB
- DOSINST
- INSTSEG
- HELPSEG
- SVM

CMSBAM and CMSDOS are defined as members of a segment space called DOSBAM.

CMSVMLIB is a physical saved segment that contains the VMLIB logical saved segment.

INSTSEG is a physical saved segment that contains the CMSINST logical saved segment. HELPSEG is a physical saved segment that contains the HELP logical saved segment. The DCSSGEN and SAVEFD commands are no longer used to install CMSINST and HELP.

The VMSES/E VMFSGMAP EXEC is used to view the supplied saved segment definitions. VMFSGMAP can also be used to add or delete saved segment definitions. The VMSES/E VMFBLD EXEC is then used to build the saved segments.

For general information about using VMSES/E to view, modify, and build saved segments, see *VM/ESA: Planning and Administration*. For information about the VMFSGMAP EXEC and other parts of the VMSES/E saved segment support, see the *VM/ESA: VMSES/E Introduction and Reference*.

FBA DASD Conversion Considerations [1.1.5]

Considerations for FBA DASD migration include:

- Temporary use of 370 DASD allocations for CP-owned volumes (PAGE, PERM, TDSK, DRCT, DUMP, OVRD, and TEMP space). This is supported only for FBA. See “Temporary Use of 370 DASD Allocations for VM/ESA 2.4.0 CP-Owned Volumes [1.1.5]” on page 232.
- You can use your old system's TEMP space as SPOL space on VM/ESA 2.4.0, but your spool files remain incompatible. See “TEMP Space as SPOL Space -- Spool Files Remain Incompatible [1.1.5]” on page 233.
- To be eligible for minidisk cache, which is supported with Expanded Storage, your FBA minidisks must be aligned on 4KB boundaries. See “Aligning Minidisks on 4KB Boundaries for Minidisk Cache and for SFS [1.1.5]” on page 233.
- FBA DASD allocation record. See “FBA DASD Allocation Record [1.1.5]” on page 236.
- Accounting record type 3 (temporary disk space) changes. See “Device Class Field in Accounting Record 3 Changed [1.1.5]” on page 299.
- 3370s and 9332s are not supported as installation devices.

Temporary Use of 370 DASD Allocations for VM/ESA 2.4.0 CP-Owned Volumes [1.1.5]

VM/ESA 2.4.0 can use FBA DASD volumes that were formatted by VM/ESA 1.1.5 370 Feature. The old space allocation types are mapped to VM/ESA space allocation types. See “DASD Space Allocation Mappings between Your Old Release and VM/ESA 2.4.0” for these mappings. This can help during migration to VM/ESA 2.4.0. It is recommended, however, that FBA DASD volumes be reformatted using ICKDSF, Release 14 or later, before bringing your new VM/ESA 2.4.0 system into production. Reasons for this recommendation:

- Performance. Changes to allocations of certain DASD space types, especially PAGE and SPOL, may enhance the performance of page and spool subsystems. See “DASD Space Allocation Mappings between Your Old Release and VM/ESA 2.4.0” for more information on PAGE and SPOL space allocations and *VM/ESA: Planning and Administration* for information about DASD space requirements.
- To enhance efficient use of DASD space. OVRD space that was “active” in your old system is not used by the VM/ESA 2.4.0 system. To recover this space you need to reformat the DASD.
- To take advantage of new function. You should consider allocating some space for the new allocation type, PARM, to take advantage of the new CP configurability function.

Once you have used ICKDSF, Release 14 or later, to reformat your FBA DASD with VM/ESA allocation types, you cannot use that FBA DASD with your old system. You have to reformat the FBA DASD if you want to reuse your old system's allocation types. Consider saving your 370 allocations, even if just on paper, in case you need to recreate them for backout to your old system.

Note: Temporary use of 370 allocation types is supported only for FBA DASD.

DASD Space Allocation Mappings between Your Old Release and VM/ESA 2.4.0

The following table shows how VM/ESA 2.4.0 allocates DASD space for allocations you used with your old system.

Table 21 (Page 1 of 2). Mapping of 370 DASD allocation types to VM/ESA DASD allocation types

370 Allocation Type	VM/ESA Allocation Type	Considerations
PAGE	PAGE	None
PERM	PERM	None
TDSK	TDSK	None
DRCT	DRCT	None

Table 21 (Page 2 of 2). Mapping of 370 DASD allocation types to VM/ESA DASD allocation types

370 Allocation Type	VM/ESA Allocation Type	Considerations

DUMP	SPOL	To set aside a volume to be used for DUMP space, use the DUMP option of the SYSCPVOL macro in HCPSYS in combination with the SET DUMP DASD command. In VM/ESA 2.4.0, SYSCPVOL replaces SYSOWN.
		For example, in HCPSYS, you could set up a volume called DUMP1 for dumps only:
		`SYSCPVOL acpvol,(dump1,d),cpvol2`
		If you are using the new system configuration statements instead of an HCPSYS file, use the CP_OWNED statement. In the following example, a volume named DUMP1 is set up for dumps only at slot 7 in the CP-owned volume list:
		`cp_owned slot 007 dump1 dump`

OVRD	SPOL	In VM/ESA 2.4.0, override information is kept in system data files. Therefore, there is no corresponding allocation type. “Inactive” OVRD space is used. “Active” OVRD space is not used and should be reformatted as SPOL space.
TEMP	SPOL	If your old system is using TEMP space for all paging and spooling, you should use ICKDSF to reformat and reallocate these FBA DASD volumes before using your VM/ESA 2.4.0 system in production to improve performance: • Reallocate the TEMP space as separate PAGE space and SPOL space • If possible, place PAGE and SPOL on separate volumes. TEMP space can be used for VM/ESA SPOL space. However, the spool files themselves remain incompatible. See “Spool File Conversion Considerations and Procedures [S370]” on page 88 for information about spool file incompatibilities and how to migrate your spool files.

TEMP Space as SPOL Space -- Spool Files Remain Incompatible [1.1.5]

TEMP space can be used for VM/ESA SPOL space. However, the spool files themselves remain incompatible. See “Spool File Conversion Considerations and Procedures [S370]” on page 88 for information about spool file incompatibilities and how to migrate your spool files.

Aligning Minidisks on 4KB Boundaries for Minidisk Cache and for SFS [1.1.5]

To be eligible for minidisk cache, which is supported with Expanded Storage, minidisks must be aligned on 4KB boundaries and formatted as 4KB.

FBA minidisks set up for the SFS file pool must also be aligned on 4KB boundaries and formatted for 4KB. At this time, only storage group minidisks are required to be aligned on 4KB boundaries. Note that control minidisks should continue to be 512-byte aligned.

If minidisks for the SFS file pool are not aligned on 4KB boundaries, errors or warning messages occur when the administrator or the operator enters one of the following commands:

- FILESERV START
- FILESERV GENERATE
- FILESERV MINIDISK
- ENABLE GROUP
- FILEPOOL RESTORE
- FILEPOOL ENABLE GROUP (new)
- FILEPOOL MINIDISK (new)

Steps to Follow to Prepare FBA Minidisks for Minidisk Cache

The steps to follow are:

1. Save the data from the minidisks you are aligning. You can use DDR to do this.
2. Align the minidisks as defined in the directory. See “Aligning Minidisks on 4KB Boundaries” for an example of how to do this.
3. Format the minidisks for 4KB. See “Formatting Minidisks for 4KB” on page 235 for a discussion on this.
4. Restore the data to the minidisks.

Note that there may be trade offs between optimization of DASD space and faster I/O response.

Steps to Follow to Prepare FBA Minidisks for SFS File Pool Use

The steps to follow are described in complete detail in the section about replacing SFS and CRR file pool minidisks in the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book. The steps outlined include:

1. Saving the data from the minidisks.
2. Aligning the minidisks as defined in the directory. “Aligning Minidisks on 4KB Boundaries” shows an example of how to do this in this situation.
3. Replacing the minidisks.
4. Restoring the data to the minidisks.

Aligning Minidisks on 4KB Boundaries

Make sure the data in these minidisks is saved before changing the alignment. Then, after the minidisk has been aligned and formatted or replaced, restore the data. You can use DDR to save and restore the data.

In the user directory, make sure the MDISK statements of minidisks on FBA DASD define the minidisks to start and end on 4KB boundaries. To do this, make sure that both of the following can be divided evenly by eight:

- The starting block
- The ending block plus one or the total number of blocks being defined.

For example, consider the following SFS storage group minidisk definition:

```
MDISK 199 9332 6000 18250 SFSVOL R PURPLE ORANGE
```

6000 / 8 = 750. The starting block is aligned on a 4KB boundary. 18250 / 8 = 2281.25. 18250 is the number of blocks being defined. Because this number does not divide evenly by eight, the ending block is not aligned on a 4KB boundary.

To align the ending block on a 4KB boundary, you could round 2281.25 up to the nearest integer, 2282, then multiply by eight. 2282 x 8 = 18256. Actually, if you are aligning this minidisk so that it can work with your VM/ESA 2.4.0 SFS file pool server, you must round up. To replace a minidisk used with SFS, the new minidisk must be the same size or bigger than the original minidisk. So, to make this minidisk 4KB aligned, it could be redefined like this:

```
MDISK 199 9332 6000 18256 SFSVOL R PURPLE ORANGE
```

Note that this would work only if there is not another minidisk that you would be overlapping.

In the above example, if you were aligning this minidisk so it could work with minidisk cache, it may be easiest to simply round down rather than round up. In this way, you could avoid overlapping minidisks. For example, in the previous situation, you would round 2281.25 down to the nearest integer, 2281, then multiply by eight. 2281 x 8 = 18248. So, if it were not an SFS minidisk, you could redefine the minidisk like this:

```
MDISK 199 9332 6000 18248 SFSVOL R PURPLE ORANGE
```

In order to get your newly aligned minidisks to all fit together, you may have to move some of them.

Formatting Minidisks for 4KB

To take advantage of minidisk cache, minidisks must be formatted for 4KB. Also, as always, SFS storage group minidisks must be formatted for 4KB. This is done when you format and reserve the minidisk, which is part of what you do when you execute step 3 described under "Steps to Follow to Prepare FBA Minidisks for SFS File Pool Use" on page 234. This is fully explained in the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book.

When using FORMAT, the default for FBA is 1KB. The default for CKD or ECKD minidisks is 4KB. These defaults usually help optimize DASD space. (A usage note for the FORMAT command in the *VM/ESA: CMS Command Reference* talks about choosing an appropriate block size depending on the intended use of the minidisk.) For any minidisks not formatted as 4KB or aligned on 4KB boundaries, you must decide whether you want to take advantage of:

- Faster I/O response (by making them 4KB minidisks so they can use the minidisk cache)
- Optimizing your DASD space (by taking the default for FORMAT, or by changing the default depending on the type of files the minidisk is to contain).

If you want to take advantage of minidisk cache to improve I/O response time, then you and your users must reformat your FBA minidisks for 4KB. For example:

```
format 191 a (blksize 4k
```

FBA DASD Allocation Record [1.1.5]

Applications that depended on the size and format of the CP DASD allocation record in the DASD allocation block (ALOCBLOK, which is HCPALOC in VM/ESA 2.4.0) in your old system must change to run on VM/ESA 2.4.0. These applications must recognize space allocation codes for FBA DASD formatted by VM/ESA.

Applications That Use Device Class Field of Accounting Record 3 [1.1.5]

If you have applications that depend on the device class field of accounting record 3, you may have to make changes. See “Device Class Field in Accounting Record 3 Changed [1.1.5]” on page 299 for details.

LOAD LIST Message Change for VMFBLD, VMFINS BUILD, and HCPSADMP [1.2.0]

The LOAD LIST message, which may result from issuing VMFBLD, VMFINS BUILD, or HCPSADMP execs, has changed. It now includes the time and date of the file that was used as the load list.

For example, the LOAD LIST message in your old release might have looked like this:

```
LOAD LIST:  $$$TLL$$ EXEC    A1 (MNT191)
```

This message in VM/ESA 2.4.0 looks like this:

```
LOAD LIST:  $$$TLL$$ EXEC    A1 02/24/93 09:25 (MNT191)
```

Converting Your SFS File Pool Servers [ALL]

You will probably want to have the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book available as you work through the steps of converting to VM/ESA 2.4.0 SFS file pool servers.

Using Two System Images during System Conversion

For this type of conversion, you have both the old system and the new system running at the same time during the conversion. You use two physical processors, a logical partition, or a second level system. See Chapter 5, “Determining Which Conversion Strategies You Want to Use” on page 21 for more information on what the conversion strategies are and how to select a strategy.

You can make the file pool server conversion in one of 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.

  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 *VM/ESA: 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.

Steps to Follow if You Are Using Two System Images during System Conversion

1. **[1.1.5]** Apply APAR VM53385 to your old CMS code.

   This APAR allows you to convert your server back to your old release if you have to back out.

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

   See the section on backing up the user data in the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book for help on doing this if you need it. This backup can be used on your VM/ESA 2.4.0 server. It can also be used if you need to back out to your old release.

   Notes:

   a. **[1.1.5]** Backup files created on your new system are not supported on the old system.

   b. **[1.2.0, 1.2.1, 1.2.2, 2.1.0]** Control data backup files created on your new system are not supported on the 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 VM/ESA 2.4.0 system.

   During the install, make sure the VM/ESA 2.4.0 CMS code gets loaded, but do not install or set up the SFS servers. The VM/ESA 2.4.0 CMS code contains the SFS server code.

5. Set up a file pool server machine.

   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 “System Directory Entry Considerations and Changes for Servers” on page 240.

   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 or DDR command to move the contents of the old server's minidisks onto the new server's minidisks. See the *VM/ESA: 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 *VM/ESA: CMS File Pool Planning, Administration, and Operation* book for details on how to do this.

6. **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 *VM/ESA: 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.

7. Log on to the server machine. Make sure it uses the VM/ESA 2.4.0 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.

8. IPL the new CMS.

9. 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 *VM/ESA: CMS File Pool Planning, Administration, and Operation*. This can help you avoid potential virtual storage problems.

10. 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
    ```

11. 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
```

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

    Notes:

    a. 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.

    b. **[1.1.5]** Neither control data nor storage group data backup files created on your new system are supported on the old system.

    c. **[1.2.0]** Control data backup files created on your new system are not supported on the old system. Storage group data backup files created on your new system are supported on the old system.

13. If you encounter problems and need to back out to your old system, see "Converting a CMS File Pool Server from VM/ESA 2.4.0 Back to Your Old Release" on page 242.

Using the 'Cut and Go' Strategy during System Conversion

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. See Chapter 5, "Determining Which Conversion Strategies You Want to Use" on page 21 for more information on what the conversion strategies are and how to select a strategy.

Steps to Follow If You Are Using the 'Cut and Go' Strategy

1. **[1.1.5]** Apply APAR VM53385 to your old CMS code.

   This APAR allows you to convert your server back to your old release if you have to back out.

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

   See the section on backing up the user data in the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book for help on doing this if you need it. This backup can be used on your VM/ESA 2.4.0 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.

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

4. 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
```

5. Install the VM/ESA 2.4.0 system.

   During the install, make sure the VM/ESA 2.4.0 CMS code gets loaded, but do not install or set up the SFS servers. The VM/ESA 2.4.0 CMS code contains the SFS server code.

6. Make sure the server virtual machine has access to the disks where the VM/ESA 2.4.0 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.

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 *VM/ESA: CMS File Pool Planning, Administration, and Operation* book. 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. If you encounter problems and need to back out to your old system, see “Converting a CMS File Pool Server from VM/ESA 2.4.0 Back to Your Old Release” on page 242.

System Directory Entry Considerations and Changes for Servers

You may have to convert some of your SFS file pool servers' directory entries.

If you have not already, make the following changes:

- **USER directory control statement:**
  - Make sure the minimum and maximum virtual storage is at least 32MB.
  - **[1.1.5]** Remove the priority 1, if present, after the BG.
- **OPTION directory control statement:**
  - Remove BMX operand; it is not applicable in VM/ESA 2.4.0.
  - 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 step 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.

    Note that QUICKDSP replaces the SET QDROP OFF command.

   - 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 the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book. 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 VM/ESA dispatch queue.

     SHARE REL 1500 replaces the SET SHARE *userid* REL 1500 command.

- **MACHINE directory control statement:**
   - If not done, replace 370 with XA or XC.

     SFS file pool servers should use XC if you have an ES/9000 processor. CRR recovery servers do not exploit data spaces and, therefore, should be set to XA. Note that if you do not have an ES/9000 processor, CP allows you to specify XC, but automatically places the machine in XA mode.

- **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 running on an ES/9000 processor.

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

CMS File Pool Features You Should Not Exploit until Your VM/ESA 2.4.0 System Is Stable

IBM recommends that you do not exploit and do not allow your users to exploit new VM/ESA 2.4.0 CMS file pool features until you are certain that you will not need to convert the file pool server back to your old release. New features that you should avoid include:

- **[1.1.5, 1.2.0, 1.2.1]** Unresolved aliases
- **[1.1.5]** Empty files.
- **[1.1.5]** External objects.
- **[1.1.5]** DFSMS/VM management of SFS file pools.
- **[1.1.5]** Expanded character set for VM/ESA 2.4.0 directory names.

Additional Considerations [1.1.5]

If you exploit SFS VM data spaces support and then convert your server back to VM/ESA 1.1.5 370 Feature, you will notice a performance degradation.

There are other new VM/ESA 2.4.0 SFS features not listed above that do not affect your capability to convert the SFS file pool server back to your old system. Use of these features in applications may, however, impact these applications when you convert the SFS file pool server back to your old system.

Converting a CMS File Pool Server from VM/ESA 2.4.0 Back to Your Old Release

VM/ESA 2.4.0 does not support the conversion of CMS file pools and servers back to a previous release. However, IBM does recognize that special situations exist where such conversion may be desired. The following guidance is given to assist you with that conversion. Please read this entire section before converting your VM/ESA 2.4.0 file pool server back to your old system.

Some of the problems you may encounter if you do not convert your file pool server back properly include:

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

Steps to Convert Your File Pool Server Back

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

1. If required, perform the activities described in “What If You Have Exploited New VM/ESA 2.4.0 SFS Features?” on page 244 to remove the effects of exploiting new SFS features:

   - **[1.1.5, 1.2.0, 1.2.1]** Unresolved aliases
   - **[1.1.5]** Empty files
   - **[1.1.5]** External objects
   - **[1.1.5]** DFSMS/VM management of SFS file pools
   - **[1.1.5]** Expanded character set for VM/ESA 2.4.0 directory names

2. Shut down the VM/ESA 2.4.0 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 conversion 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 *VM/ESA: CMS File Pool Planning, Administration, and Operation* for LUNAME start-up parameter considerations.

3. Convert back to your old release.

   Additional Considerations [1.1.5]

   To do this step in a conversion from VM/ESA 1.1.5 370 Feature, you may have to:

   - Review and change any start-up parameters as needed.

     VM/ESA 2.4.0 has one new start-up parameter: DFSMS or NODFSMS. If you have not already removed it, remove this start-up parameter. Even if you have not exploited DFSMS/VM, you have to remove the NODFSMS start-up parameter if it is in your DMSPARMS file.

   - Remove the ESECURITY start-up parameter from the *serverid* DMSPARMS file if you are using SFS external security manager support in VM/ESA 2.4.0.

     Also, the recommended values for certain SFS start-up parameters differ between VM/ESA 2.4.0 and VM/ESA 1.1.5 370 Feature. Review the VM/ESA 1.1.5 370 Feature recommended values for the following start-up parameters before converting back to VM/ESA 1.1.5 370 Feature:

     – CATBUFFERS
     – CTLBUFFERS
     – USERS.

   - Prepare the user directory for use on your old system.

     Use the recommended values for user directory control statements for file pool servers. See your old *VM/ESA: CP Planning and Administration for 370* book for these recommended values.

     In addition, you must consider user directory control statements and operands:

     – Introduced to support XC mode
     – Not supported by CP370

     These new control statements and operands must be removed or replaced by appropriate VM/ESA 1.1.5 370 Feature control statements and operands.

   - Make sure the FORMAT or NOFORMAT start-up parameter is NOFORMAT. The default is FORMAT. Doing this saves time because you are using existing logs, which have already been formatted and reserved.

4. On the old system:

   a. **[1.1.5]** Apply APAR VM53385 to your old system, if you have not already.

   b. **[1.1.5]** Execute FILESERV LOG *vdev1 vdev2*. For example:

   ```
   fileserv log 302 303
   ```

   To find out what *vdev1* and *vdev2* should be, you can look at the DDNAME=LOG1 and DDNAME=LOG2 records in the *filepoolid* POOLDEF file. For example:

```
   .
   .
   .
DDNAME=LOG1        VDEV=302
DDNAME=LOG2        VDEV=303
   .
   .
   .
```

c. 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
```

Control backup files created by VM/ESA 2.4.0 cannot be processed by earlier releases. Also, control backup files created by earlier releases cannot be processed by VM/ESA 2.4.0.

d. 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
```

e. **[1.1.5]** Use the FILEPOOL BACKUP command to create new user data backup files. See your old *VM/ESA: CMS Planning and Administration Guide for 370* for information on how to do this.

User data backup files created by VM/ESA 2.4.0 cannot be processed by your old system. After you have converted back to your old system, back up all user storage groups as soon as possible.

If you have not yet created user data backup files with VM/ESA 2.4.0, that is, your current user data backup files were created under your old system, you do not have to re-create user data backup files.

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

What If You Have Exploited New VM/ESA 2.4.0 SFS Features?

Problems you may have to address if you or your SFS users have exploited new VM/ESA 2.4.0 features include:

- The time and effort to remove or change the effects of any usage may be quite extensive.
- SFS users may already be accustomed to, and like, the new VM/ESA 2.4.0 SFS features.

Unresolved Aliases [1.1.5, 1.2.0, 1.2.1]: If you have used unresolved aliases, they must be erased before migrating the SFS server back to your old release. See the information about recovery procedures in the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book for how to do this.

Empty Files or External Objects [1.1.5]: You have to locate and eliminate all empty files and external objects before converting back to your old system.

Note: If DFSMS/VM is managing the file pool, it can create empty files if the file expiration attribute, *expiration disposition*, of a Management Class in the configuration file has a value of DATA.

For empty files, you can do either of the following to eliminate them:

- Erase them
- Make them nonempty by adding records to them.

For external objects, you must erase them.

You can use the following procedure to locate and eliminate empty files and external objects:

1. Determine which files are empty or if there are external objects.

   To locate empty files or external objects you can use the ALLEO EXEC shown in Appendix D, “Sample Exec for Finding Empty Files and External Objects in SFS Directories” on page 643. The ALLEO EXEC creates a file called ALLEO FILES that lists the directories and the empty files and external objects.

   If you need to, see “How to tell if a file is empty” on page 246 or “How to tell if a file is an external object” on page 246.

2. Erase all empty files and external objects.

   You can use the results of the ALLEO EXEC, which show you the directories and the empty files and external objects. Do one of the following to get rid of all empty files and external objects:

   - Use the ERASE command. For example:

     ```
     erase empty file cupr.fourq.documentation
     ```

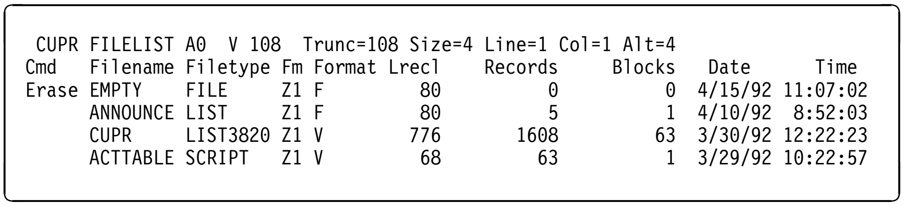
   - If a directory containing an empty file or external object is accessed read/write, erase the file from the FILELIST screen using ERASE or DISCARD. For example:

     ```
      BLACKBUL FILELIST A0  V 108  Trunc=108 Size=4 Line=1 Col=1 Alt=4
     Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date      Time
     Erase EMPTY    FILE     Z1 F         80          0          0  4/15/92 11:07:02
           ANNOUNCE LIST     Z1 F         80          5          1  4/10/92  8:52:03
           CUPR     LIST3820 Z1 V        776       1608         63  3/30/92 12:22:23
           ACTTABLE SCRIPT   Z1 V         68         63          1  3/29/92 10:22:57
     ```

   - If you do not have the directory accessed read/write but are authorized for read/write, reaccess the directory or subdirectory using the FORCERW option of the ACCESS command. For example, on the DIRLIST screen do the following:

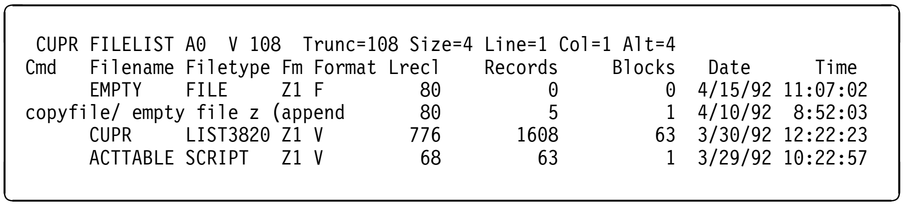
     ```
      CUPR DIRLIST  A0  V 139  Trunc=319 Size=11 Line=1 Col=1 Alt=1
          J  SERVER3:CUPR.
          -  SERVER3:CUPR.CUSTOMERS
          -  SERVER3:CUPR.FOURQ
          -  SERVER3:CUPR.FOURQ.ACTIONITEMS
     access/ z (forcerwUPR.FOURQ.DOCUMENTATION
          -  SERVER3:CUPR.FOURQ.SERVICE
          -  SERVER3:CUPR.INTRODUCTION
          -  SERVER3:CUPR.ONEQ
          -  SERVER3:CUPR.ONEQ.ACTIONITEMS
          -  SERVER3:CUPR.ONEQ.DOCUMENTATION
          -  SERVER3:CUPR.ONEQ.OVERALL
     ```

     Then use FILELIST, and use ERASE. For example:



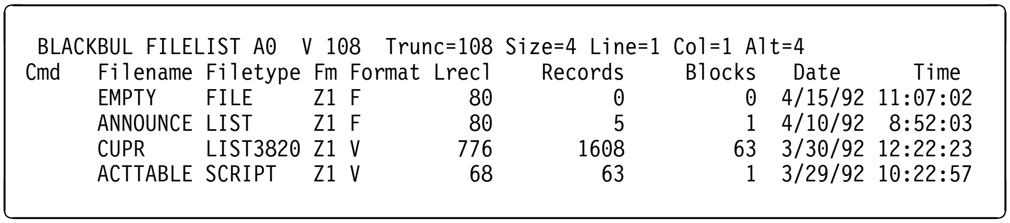
```
 CUPR FILELIST A0  V 108  Trunc=108 Size=4 Line=1 Col=1 Alt=4
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date      Time
Erase EMPTY    FILE     Z1 F         80          0          0  4/15/92 11:07:02
      ANNOUNCE LIST     Z1 F         80          5          1  4/10/92  8:52:03
      CUPR     LIST3820 Z1 V        776       1608         63  3/30/92 12:22:23
      ACTTABLE SCRIPT   Z1 V         68         63          1  3/29/92 10:22:57
```

- For empty files, you can make them nonempty by using commands such as COPYFILE or XEDIT to add records to the files. For example, using COPYFILE, do the following:



```
 CUPR FILELIST A0  V 108  Trunc=108 Size=4 Line=1 Col=1 Alt=4
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date      Time
      EMPTY    FILE     Z1 F         80          0          0  4/15/92 11:07:02
copyfile/ empty file z (append       80          5          1  4/10/92  8:52:03
      CUPR     LIST3820 Z1 V        776       1608         63  3/30/92 12:22:23
      ACTTABLE SCRIPT   Z1 V         68         63          1  3/29/92 10:22:57
```

How to tell if a file is empty: Files with the value 0 in the 'Records' column are empty files. For example, EMPTY FILE in the following FILELIST screen is an empty file.

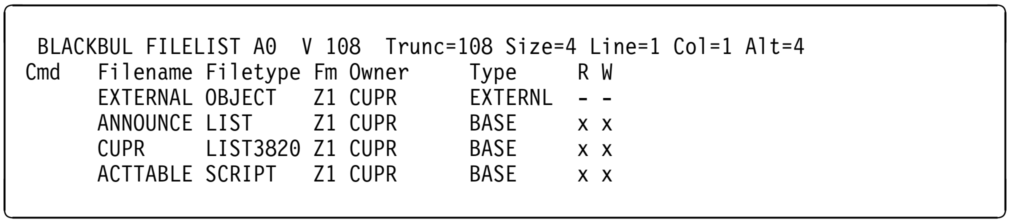


```
 BLACKBUL FILELIST A0  V 108  Trunc=108 Size=4 Line=1 Col=1 Alt=4
Cmd   Filename Filetype Fm Format Lrecl    Records     Blocks   Date      Time
      EMPTY    FILE     Z1 F         80          0          0  4/15/92 11:07:02
      ANNOUNCE LIST     Z1 F         80          5          1  4/10/92  8:52:03
      CUPR     LIST3820 Z1 V        776       1608         63  3/30/92 12:22:23
      ACTTABLE SCRIPT   Z1 V         68         63          1  3/29/92 10:22:57
```

How to tell if a file is an external object: External objects have '-' in the 'Records' column of FILELIST. SFS directories and revoked or erased aliases also have '-' in the 'Records' column. So, to determine if a file is an external object, you can use FILELIST with the SHARE option. For example:

```
filelist * * z (share
```

The file called EXTERNAL OBJECT in the resulting screen is an external object. Its 'Type' column says 'EXTERNL'.



```
 BLACKBUL FILELIST A0  V 108  Trunc=108 Size=4 Line=1 Col=1 Alt=4
Cmd   Filename Filetype Fm Owner     Type     R W
      EXTERNAL OBJECT   Z1 CUPR      EXTERNL  - -
      ANNOUNCE LIST     Z1 CUPR      BASE     x x
      CUPR     LIST3820 Z1 CUPR      BASE     x x
      ACTTABLE SCRIPT   Z1 CUPR      BASE     x x
```

DFSMS/VM [1.1.5]: If you have exploited DFSMS/VM management of SFS file pools, you must remove its effects:

1. Remove any empty files, as described in “Empty Files or External Objects [1.1.5]” on page 244.

2. Locate all migrated files. Recall or erase them. The section about recalling all files to primary storage in the *VM/ESA: DFSMS/VM Function Level 221 Installation and Customization* book shows how to do this.

3. Remove the DFSMS start-up parameter from the DMSPARMS file.

Note: Even if you have not exploited DFSMS/VM, you have to remove the NODFSMS start-up parameter if it is in your DMSPARMS file.

Expanded Character Set for VM/ESA 2.4.0 Directory Names [1.1.5]: VM/ESA 2.4.0 allows you to use the following additional characters to specify directory names: $, _, *, and #. Also, you can start a directory name with a number. In VM/ESA 1.1.5 370 Feature these were not allowed.

You need to search for all directory names that have exploited the expanded character set and change the directory name or remove the directory. For example, you could:

1. Use DIRLIST to look for all directories that use the expanded character set. For example, looking at the CUPR user ID's directory list:

```
 CUPR DIRLIST  A0  V 139  Trunc=319 Size=11 Line=1 Col=1 Alt=1
      J  SERVER3:CUPR.
      -  SERVER3:CUPR.CUSTOMERS
      -  SERVER3:CUPR.4Q91
      -  SERVER3:CUPR.4Q91.ACTION_ITEMS
      -  SERVER3:CUPR.4Q91.DOCUMENTATION
      -  SERVER3:CUPR.4Q91.SERVICE
      -  SERVER3:CUPR.INTRODUCTION
      -  SERVER3:CUPR.1Q92
      -  SERVER3:CUPR.1Q92.ACTION_ITEMS
      -  SERVER3:CUPR.1Q92.DOCUMENTATION
      -  SERVER3:CUPR.1Q92.OVERALL
```

2. You could then scan the directory list, and erase or rename those directories with names that use the expanded character set. For example:

```
 CUPR DIRLIST  A0  V 139  Trunc=319 Size=11 Line=1 Col=1 Alt=1
      J  SERVER3:CUPR.
      -  SERVER3:CUPR.CUSTOMERS
rename/ cupr.fourq91R.4Q91
rename/ cupr.fourq91.actionitemsN_ITEMS
discard  SERVER3:CUPR.4Q91.DOCUMENTATION
discard  SERVER3:CUPR.4Q91.SERVICE
      -  SERVER3:CUPR.INTRODUCTION
rename/ cupr.oneq92PR.1Q92
rename/ cupr.oneq92.actionitemsON_ITEMS
discard  SERVER3:CUPR.1Q92.DOCUMENTATION
discard  SERVER3:CUPR.1Q92.OVERALL
```

Consider modifying the ALLEO EXEC in Appendix D, “Sample Exec for Finding Empty Files and External Objects in SFS Directories” on page 643 to find these directories for you.

Chapter 11. Service Changes and Conversion Considerations

See the *VM/ESA: VMSES/E Introduction and Reference* and the *VM/ESA: Service Guide* for details about these changes.

This chapter may contain information for any or all of the conversions documented in this book. For definitions of the conversion identifiers used in this chapter, see “How to Identify the Topics That Apply to Your Conversion” on page 3.

This chapter discusses the following major topics:

- “Changes in the Product Parameter Files ($PPF) [ALL]”
- “Build Changes [1.1.5, 1.2.0, 1.2.1]” on page 250
- “Summary of Service Exec Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]” on page 250

Changes in the Product Parameter Files ($PPF) [ALL]

This section discusses the changes made to the product parameter file in VM/ESA 2.4.0.

File Names for $PPF Files [ALL]

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

Component	File name and file type
CP	2VMVMB40 $PPF
CMS	2VMVMA40 $PPF
REXX	2VMVMF40 $PPF
VMSES/E	2VMVMK40 $PPF
GCS	2VMVML40 $PPF
Dump Viewing Facility	2VMVMI40 $PPF
TSAF	2VMVMH40 $PPF
AVS	2VMVMD40 $PPF

Saved Segment $PPF File [1.1.5]

A SEGBLD $PPF is provided. It contains control information for mapping and building saved segments.

Data Block Changes [1.1.5, 1.2.0, 1.2.1]

The following table lists the data blocks that have been added, deleted, or modified in the product parameter file.

Table 22. Data Block Changes

Tag / Data Block	Type of Change	Contents
:CNTRLOP. Control Options	Modified	The following tag has been changed: **Tag** — **Definition** :CKGEN. — Supports new options: • **[1.2.0]** LOGMOD • **[1.1.5, 1.2.0, 1.2.1]** NOVVT

Build Changes [1.1.5, 1.2.0, 1.2.1]

The following major changes have been made to the build process and to the VMFBLD EXEC. For details, see the *VM/ESA: VMSES/E Introduction and Reference*.

Enhancements:

- **[1.1.5, 1.2.0, 1.2.1]** VMFBLD processes multiple select data files ($SELECT) to determine build requirements.
- **[1.1.5, 1.2.0, 1.2.1]** VMFBLD processes all objects specified with the LIST operand.
- **[1.2.0]** VMFBLD supports new build list options to build callable services libraries, to specify a segment name for building DOS libraries, and to build generated objects.
- **[1.2.0, 1.2.1]** VMFBLD supports a new option: PRIVATE.

Changes:

- **[1.1.5, 1.2.0, 1.2.1]** VMFBLD now accepts mixed case entries in select data files.

Deletions:

- **[1.1.5, 1.2.0, 1.2.1]** VMFBLD now only compares new and old build lists to determine objects to be deleted.

Summary of Service Exec Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]

The following table summarizes the changes in the service execs. For full details, see the *VM/ESA: VMSES/E Introduction and Reference*.

The following terms are used in the table:

Term	Meaning
Change	Changes have been made to the exec
Deletion	Function has been removed from the exec
Enhancement	New function has been added to the exec
New	The exec is new

Table 23 (Page 1 of 3). Changes to the Service Execs

Exec	Type of Change	Explanation
GENCPBLS	New	**[1.1.5, 1.2.0, 1.2.1]** Updates the CP load list build list.
ITNVTSTR	Enhancement	**[2.2.0]** Supports new operands: PROD, KEY
VMFAPPLY	Enhancement	**[1.1.5, 1.2.0, 1.2.1]** Now accepts mixed case entries in select data files, apply lists, and exclude lists.

Table 23 (Page 2 of 3). Changes to the Service Execs

Exec	Type of Change	Explanation
VMFASM	Enhancement	Supports new options: • **[1.2.0]** ASM HL • **[1.2.0]** HLASM *hlasm_options* EHLASM • **[1.2.0]** LOGMOD • **[1.2.0]** FILETYPE *ft* • **[1.2.0]** NOPRINT • **[1.1.5, 1.2.0, 1.2.1]** OUTMODE • **[1.1.5, 1.2.0, 1.2.1]** $SELECT • **[1.1.5, 1.2.0, 1.2.1]** NO$SELECT • **[1.1.5, 1.2.0, 1.2.1]** NOVVT
VMFBLD		**[1.1.5, 1.2.0, 1.2.1]** See “Build Changes [1.1.5, 1.2.0, 1.2.1].”
VMFDEL	Deletion	**[1.2.0]** This exec is no longer supported. Use VMFINS DELETE instead.
VMFDEP	Deletion	**[1.2.0]** This exec is no longer supported. Use VMFSIM SYSDEP instead.
VMFEXUPD	New	**[1.1.5, 1.2.0, 1.2.1]** Calls the EXECUPDT command to apply updates to a $Source program.
VMFHASM	Enhancement	Supports new options: • **[1.2.0]** LOGMOD • **[1.2.0]** FILETYPE *ft* • **[1.1.5, 1.2.0, 1.2.1]** OUTMODE • **[1.1.5, 1.2.0, 1.2.1]** $SELECT • **[1.1.5, 1.2.0, 1.2.1]** NO$SELECT • **[1.1.5, 1.2.0, 1.2.1]** NOVVT
VMFHLASM	New	**[1.2.0]** Calls the VMFASM exec with the new ASM HL option to apply updates to a source file and then assembles the file using the IBM High Level Assembler (HLASM).
	Enhancement	**[1.1.5, 1.2.1]** Supports new options: OUTMODE, $SELECT, NO$SELECT, NOVVT.
VMFINS	Enhancement	**[1.1.5, 1.2.0, 1.2.1]** The OVERRIDE and FILEPOOL options were added to VMFINS INSTALL and VMFINS MIGRATE. **[1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]** The DISABLE operand was added to VMFINS DELETE. **[1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]** The DISABLE, ENABLE, SETUP and NOSETUP operands were added to VMFINS INSTALL. **[1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]** The DISABLE and ENABLE operands were added to VMFINS MIGRATE.
	Deletions	**[1.2.0]** The DEVICE option has been removed from VMFINS INSTALL and VMFINS MIGRATE. Make sure a tape is mounted and attached at 181 (this was the default if the DEVICE option was not specified). The INFO and LIST options have been removed from VMFINS DELETE and VMFINS BUILD. Use the PPF operand instead.
	New	**[1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]** The VMFINS DISABLE and VMFINS ENABLE execs were added.
VMFINSBD	Deletion	**[1.2.0]** This exec is no longer supported. Use VMFINS BUILD instead.
VMFINST	Deletion	**[1.2.0]** This exec is no longer supported. Use VMFINS INSTALL instead.
VMFMHR	Deletion	**[1.2.0]** This exec is no longer supported. Use VMFINS MIGRATE instead.
VMFMSG	Deletion	**[1.1.5, 1.2.0, 1.2.1]** This exec is no longer supported as an external.

Table 23 (Page 3 of 3). Changes to the Service Execs

Exec	Type of Change	Explanation
VMFNLS	Enhancement	Supports new options: • **[1.2.0]** LOGMOD • **[1.2.0]** FILETYPE *ft* • **[1.1.5, 1.2.0, 1.2.1]** OUTMODE • **[1.1.5, 1.2.0, 1.2.1]** $SELECT • **[1.1.5, 1.2.0, 1.2.1]** NO$SELECT • **[1.1.5, 1.2.0, 1.2.1]** NOVVT
VMFOVER	Enhancement	**[1.1.5, 1.2.0, 1.2.1]** Supports new option: LOG.
VMFPSU	New	**[1.1.5, 1.2.0, 1.2.1]** Helps you plan for the installation of a Product Service Upgrade (PSU).
VMFREC	Enhancement	**[1.2.0]** Supports new option: INSLOG.
VMFREM	New	**[1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]** Removes PTFs received by the VMFREC exec and applied by the VMFAPPLY exec.
VMFREPL	New	**[1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]** Supports the local modification of replacement maintained parts.
VMFREQC	Deletion	**[1.2.0]** This exec is no longer supported as an external. Use VMFSIM SYSREQ instead.
VMFRMT	Deletion	**[1.2.0]** This exec is no longer supported as an external. There is no replacement.
VMFSETUP	Enhancement	**[1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]** Supports new operands: PROMPT, NOPROMPT, NOCONS
VMFSGMAP	New	**[1.1.5]** Displays a segment map and allows you to add, change, or delete saved segment definitions.
VMFSIM	Enhancement	**[1.2.0]** VMFSIM CHKLVL supports a new option: LOGMOD.
VMFSUFIN	New	**[1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]** Installs service from RSU service envelope files, COR service envelope files, or both.
	Enhancement	**[2.3.0]** Supports new ALL operand and RSUTAPE and CORTAPE options.
VMFSUFTB	New	**[1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]** 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.
VMFTSPCE	Deletion	**[1.2.0]** This exec is no longer supported as an external.
VMFVIEW	Enhancement	**[1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]** Supports new operands (all have the same function): REMOVE, VMFREM, $VMFREM.

Chapter 12. Administration Changes and Conversion Considerations

Now that you have installed the system, you are faced with the task of administration.

This chapter may contain information for any or all of the conversions documented in this book. For definitions of the conversion identifiers used in this chapter, see “How to Identify the Topics That Apply to Your Conversion” on page 3.

1.1.5

In some cases, you can perform administrative functions in the same manner that you did in VM/ESA 1.1.5 370 Feature; in other cases, you must completely alter your procedure.

This chapter discusses the following major topics:

- “CP Configurability Changes [1.1.5]”
- “ISO Date Used in Default CMS IPL Heading [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]” on page 254
- “Saved Segments and Named Saved Systems Differences [1.1.5]” on page 254
- “HELPINST Saved Segment Replaced by HELPSEG and INSTSEG [1.2.0, 1.2.1, 1.2.2, 2.1.0]” on page 266
- “Restructuring User Classes [1.1.5]” on page 266
- “Managing System Performance [1.1.5]” on page 270
- “Allocation of Real Storage for Segment Tables Has Changed [1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]” on page 275
- “Setting Up the Programmable Operator [1.1.5]” on page 276
- “CMSGEND Exec No Longer Generates PROPLIB LOADLIB [1.1.5]” on page 276
- “Setting Password Suppression [1.1.5]” on page 276
- “Setting Up Virtual Machines for Accounting and Error Recording [1.1.5]” on page 277
- “CP Changes for TCP/IP Awareness [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]” on page 278
- “Changes to Full Pack Minidisks [1.1.5]” on page 279
- “National Languages Supported [ALL]” on page 280

CP Configurability Changes [1.1.5]

The CP configurability enhancements for VM/ESA 2.4.0 cause extensive changes to the way the system is defined. The best way to learn about these changes is by referring to the *VM/ESA: Planning and Administration* book.

ISO Date Used in Default CMS IPL Heading [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

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/ESA 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).

Saved Segments and Named Saved Systems Differences [1.1.5]

Although the basic concept is the same, the saved segment support in VM/ESA 2.4.0 differs from the saved segment support in VM/ESA 1.1.5 370 Feature. Therefore, you must completely remap your installation's saved segments. See “Planning Your System's Saved Segment Layout [S370]” on page 81 for information on how to do this. You cannot simply convert your system name table (DMKSNT or SNT OVERRIDE) to be used in VM/ESA 2.4.0. “Converting DMKSNT ASSEMBLE Macros or SNT OVERRIDE Tags to VM/ESA Commands [S370]” on page 84 shows how to convert DMKSNT macros or SNT OVERRIDE tags into DEFSEG commands.

For more details on planning and administrating saved segments, see the *VM/ESA: Planning and Administration* book, which includes information about:

- Planning for saved segments based on your virtual machine size
- Planning for saved segments based on your type of virtual machine
- How to display information about existing saved segments
- How to install applications in saved segments
- How to use VMFSGMAP to plan your saved segments
- How to use VMFBLD to build saved segments you define using the new segment mapping tool.

Saved Segment Differences Due to Hardware Differences

ESA/370- and ESA/390-architected segments are 16 times larger than System/370-architected segments. A System/370 segment is 64KB (16 pages) long. An ESA/370 or ESA/390 segment is 1MB (256 pages) long.

This size difference can cause problems when you convert from a System/370 environment. One of these problems is that only 16 architected segments are available in VM/ESA 2.4.0 to a 370 virtual machine, whereas 256 architected segments are available in VM/ESA 1.1.5 370 Feature. Therefore, there are fewer segments available for your System/370 applications in VM/ESA 2.4.0.

Saved Segment Differences Due to Software Differences

In VM/ESA 2.4.0, saved segment differences include:

- Using commands rather than macros or tags to define saved segments
- Terminology and types of saved segments

How Saved Segments Work in VM/ESA 2.4.0

In VM/ESA 2.4.0, you define a saved segment with the DEFSEG command and save it with the SAVESEG command.

Classes of Saved Segments: Each saved segment you define is maintained in a system data file that has a specific ID and class. The class associated with this file can be:

A indicating an **active** saved segment

R indicating that the saved segment is active and **restricted**

S indicating that the saved segment is a **skeleton** and is not active

P indicating that the saved segment is in a **pending purge** state, meaning that the saved segment will be purged when the last user is done with it

When you define a saved segment with the DEFSEG command, a class S (skeleton) system data file is created. The file becomes an active file (class A or R) only when a corresponding SAVESEG command is issued (either by you or by some other command, such as the CMS SEGGEN command). The SAVESEG command copies the code or data included within the page ranges specified on the DEFSEG command to the associated system data file. Those page ranges must be addressable by the SAVESEG issuer. The file then is considered active. Only active saved segments can be loaded by a virtual machine.

Types of Saved Segments in CP: The types of saved segments in CP are:

discontiguous saved segment (DCSS)
is a saved segment that occupies one or more architected segments. It begins and ends on a megabyte boundary and is accessed by name.

segment space
is a saved segment composed of up to 64 member saved segments. It is referred to by a single name. It occupies one or more architected segments and begins and ends on a megabyte boundary. A user with access to a segment space has access to all of its members.

member saved segment
is a saved segment that begins and ends on a page boundary. It belongs to up to 64 segment spaces and is accessed either by its own name or by a segment space name. When a virtual machine loads any member, the virtual machine has access to all members of the space. However, the virtual machine should load the other members before trying to use them.

Member and Segment Space Directories: The system data file associated with a member saved segment contains a segment space directory identifying the segment spaces to which the member belongs. And, the segment space's system data file contains a directory of the member saved segments that belong to it.

Types of Saved Segments in CMS: The types of saved segments in CMS are:

physical saved segment
is a saved segment that must be defined as a DCSS or as a member saved segment. It contains one or more logical saved segments. A physical segment definition (PSEG) file indicates what logical saved segments are contained within the physical saved segment.

logical saved segment
is a saved segment that can contain one or more program objects (such as a module, a text file, an exec, a callable services library, language information, or user-defined objects) or file directory information for a minidisk. You may, for example, package your entire application in the logical saved segments of one physical saved segment. A logical segment definition (LSEG) file indicates the contents of the logical saved segment.

Saved Segment Terminology and Command Differences

The following table shows a summary of the differences in saved segments terminology and commands.

Table 24 (Page 1 of 2). Saved Segment Differences

Terminology or Command	Where used	Changed in ESA?	Explanation
DEFSEG command	ESA	—	Used instead of DMKSNT ASSEMBLE and SNT OVERRIDE.
DEFSEG tag	370F	Yes	Was used in the VM/ESA 1.1.5 370 Feature SNT OVERRIDE file. It is replaced with the DEFSEG command.
DEFSYS command	ESA	—	Used instead of DMKSNT ASSEMBLE and SNT OVERRIDE.
DEFSYS tag	370F	Yes	Was used in the VM/ESA 1.1.5 370 Feature SNT OVERRIDE file. It is replaced with the DEFSYS command.
DEFVM tag	370F	Yes	No exact equivalent in VM/ESA 2.4.0. Use DEFSYS or DEFSEG command instead.
Discontiguous saved segment (DCSS)	370F ESA	Yes	In VM/ESA 2.4.0, a DCSS does not have to be defined outside of the virtual machine's storage. It can be defined within the virtual machine's storage. In VM/ESA 1.1.5 370 Feature, it had to be defined outside of the virtual machine's storage. In VM/ESA 2.4.0, a DCSS is defined using the DEFSEG command (without the SPACE operand). In VM/ESA 1.1.5 370 Feature, it was defined using DMKSNT ASSEMBLE's NAMESYS macro or SNT OVERRIDE's DEFSEG tag. In VM/ESA 2.4.0, a DCSS's size is 1MB or multiples of 1MB. When less than a whole 1MB is defined, the saved segment still affects 1MB of a user's storage. In VM/ESA 1.1.5 370 Feature, a DCSS's size could be 64KB or multiples of 64KB. DCSSs can be defined as physical saved segments by using the file name of the physical segment definition (PSEG) file as the name of the saved segment on the DEFSEG command.
DMKSNT	370F	Yes	No longer used at all in VM/ESA 2.4.0. The DEFSEG and DEFSYS commands replace it.
Logical saved segment	370F ESA	No	No change

Table 24 (Page 2 of 2). Saved Segment Differences

Terminology or Command	Where used	Changed in ESA?	Explanation
Member saved segment	ESA	—	Member saved segments are what a segment space is made of. Member saved segments can be defined as physical saved segments by using the file name of the physical segment definition (PSEG) file as the name of the saved segment on the DEFSEG command.
Named saved system	370F ESA	Yes	Defined using the DEFSYS command. In VM/ESA 1.1.5 370 Feature, it was defined using DMKSNT's NAMESYS macro or the DEFSYS tag in the SNT OVERRIDE file. During the installation procedure for VM/ESA 2.4.0, the CMS named saved system is defined using the SAMPNSS command (which issues the DEFSYS command) and saved using the SAVESYS command. In the VM/ESA 1.1.5 370 Feature installation procedure, the CMS named saved system was defined and saved by using the ITASK BUILD CMS command.
Physical saved segment	370F ESA	Yes	In some cases in VM/ESA 1.1.5 370 Feature, “segment space” was used synonymously with physical saved segment. This is no longer true in VM/ESA 2.4.0. No longer defined using DMKSNT's NAMESYS macro or SNT OVERRIDE's DEFSEG tag. Physical saved segments are now defined when you create a member saved segment or a DCSS by using the file name of the PSEG file as the name of the saved segment on the DEFSEG command.
Saved system			See named saved system.
SAVESEG command	ESA	—	Used to save saved segments. If you are saving a physical saved segment, continue to use SEGGEN; SEGGEN issues SAVESEG for you.
SAVESYS command	370F ESA	No	No change except that the named saved system that you are saving is defined differently. In VM/ESA 2.4.0, it is defined using the DEFSYS command. In VM/ESA 1.1.5 370 Feature, it was defined using the DMKSNT ASSEMBLE or SNT OVERRIDE file.
SEGGEN command	370F ESA	No	Saves physical saved segments and their associated logical saved segments. No change.
SEGMENT command	370F ESA	No	Lets a virtual machine use a saved segment. No change.
SEGMENT macro	370F ESA	No	Lets an application use a saved segment. No change.
Segment space	370F ESA	Yes	In some cases in VM/ESA 1.1.5 370 Feature, “segment space” was used synonymously with physical saved segment. This is no longer true in VM/ESA 2.4.0. A CP segment space is made up of member saved segments, which are new in VM/ESA 2.4.0. Its size is 1MB or multiples of 1MB. It is not explicitly defined. Instead, when you define its member saved segments using the DEFSEG command, you use the SPACE operand to indicate what segment space the member saved segment is used in.
SNT OVERRIDE file	370F	Yes	No longer used at all in VM/ESA 2.4.0. The DEFSEG and DEFSYS commands replace it.

Differences Defining and Saving Saved Segments

Defining saved segments can have two parts:

- Defining saved segments in CP (segment spaces, member saved segments, DCSSs)
- Defining saved segments in CMS (physical saved segments, logical saved segments).

Defining saved segments in CP has changed. The SAVESEG command, for saving saved segments, still exists. Defining saved segments in CMS remains unchanged. Using the CMS SEGGEN command to save physical saved segments and their associated logical saved segments remains unchanged. Also, as mentioned in “Saved Segment Differences Due to Hardware Differences” on page 254, the size of saved segments has changed.

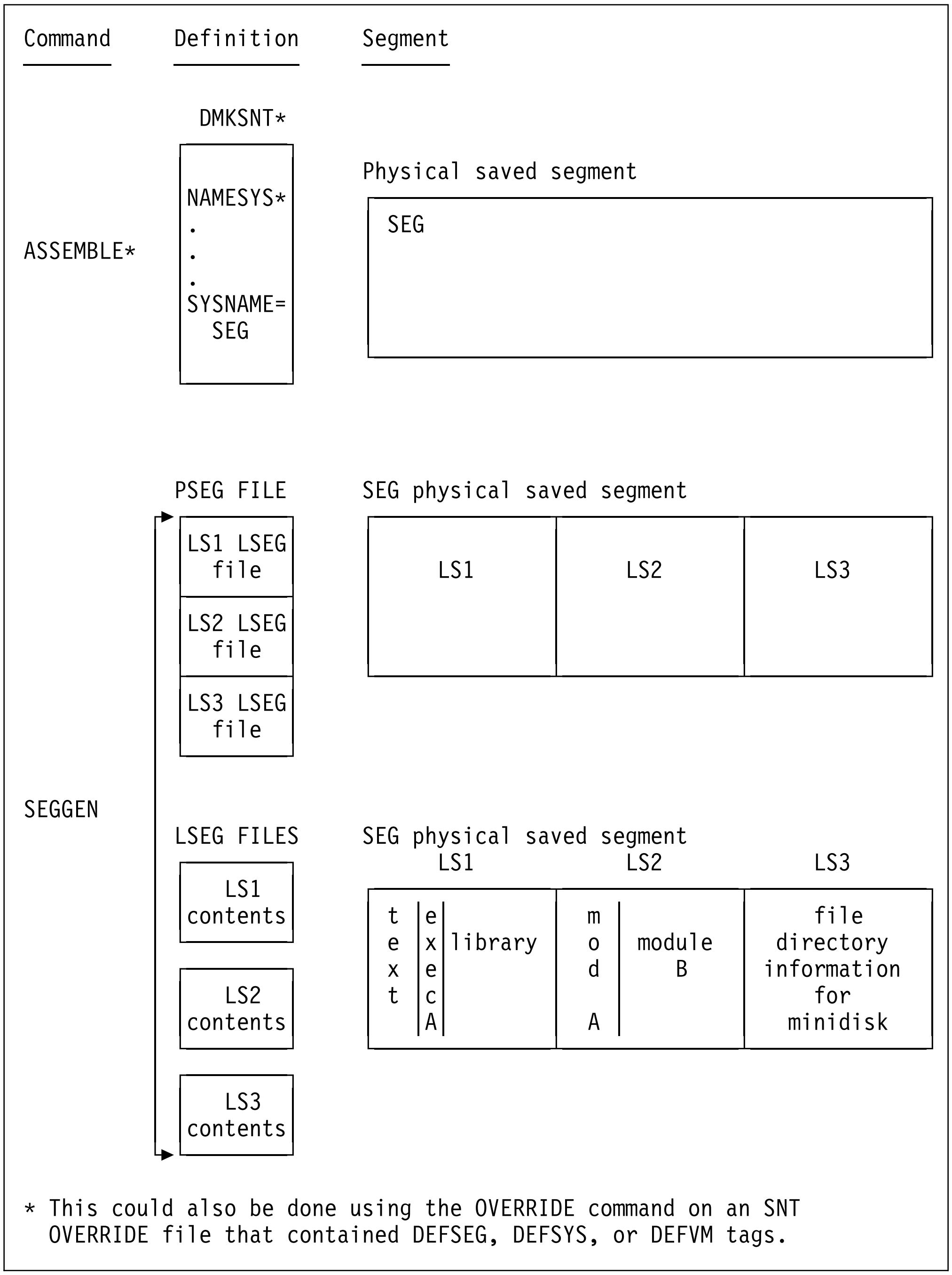
Defining Segments in VM/ESA 1.1.5 370 Feature: In VM/ESA 1.1.5 370 Feature, the terms “physical saved segment” and “segment space” were sometimes used interchangeably when talking about a physical saved segment. A discontiguous saved segment (DCSS) was a segment that had to be located entirely outside of the virtual machine (thus, the name discontiguous saved segment).

The size of saved segments could be 64KB or multiples of 64KB.

Figure 30 on page 259 shows how saved segments were defined and saved in VM/ESA 1.1.5 370 Feature.

Defining Saved Segments in VM/ESA 1.1.5 370 Feature CP: In VM/ESA 1.1.5 370 Feature, you defined a physical saved segment or a DCSS in CP using the NAMESYS macro in DMKSNT ASSEMBLE or the DEFSEG tag in SNT OVERRIDE.

Defining Saved Segments in VM/ESA 1.1.5 370 Feature CMS: Once DMKSNT was assembled, or the OVERRIDE command was issued for SNT OVERRIDE, you could define in CMS the logical saved segments that would be contained in the physical saved segment. You did this by issuing the CMS SEGGEN command. This loaded and saved your physical saved segments and their associated logical saved segments. Before issuing the SEGGEN command, you created a physical segment definition (PSEG) file that contained records that pointed to logical segment definition (LSEG) files. An LSEG file contained records that described the objects that were in a logical saved segment.



* This could also be done using the OVERRIDE command on an SNT OVERRIDE file that contained DEFSEG, DEFSYS, or DEFVM tags.

Figure 30. Defining and Saving Saved Segments in VM/ESA 1.1.5 370 Feature. The process consisted of assembling DMKSNT then issuing SEGGEN.

Defining Segments in VM/ESA 2.4.0.: In VM/ESA 2.4.0, the terms “physical saved segment” and “segment space” are used differently. The physical saved segment concept remains the same. When talking about a physical saved segment, only the term physical saved segment is used. In VM/ESA 2.4.0, a segment space is a CP concept. It is made up of member saved segments. A member saved segment never gets loaded alone into storage. The whole segment space gets loaded. A discontiguous saved segment (DCSS) can be located outside or inside of the virtual machine (in other words, the name discontiguous saved segment is now a misnomer). A physical saved segment must get defined as a member saved segment or a DCSS.

The size of segment spaces and DCSSs is 1MB or multiples of 1MB.

Figure 31 shows the differences between member saved segments and DCSSs.

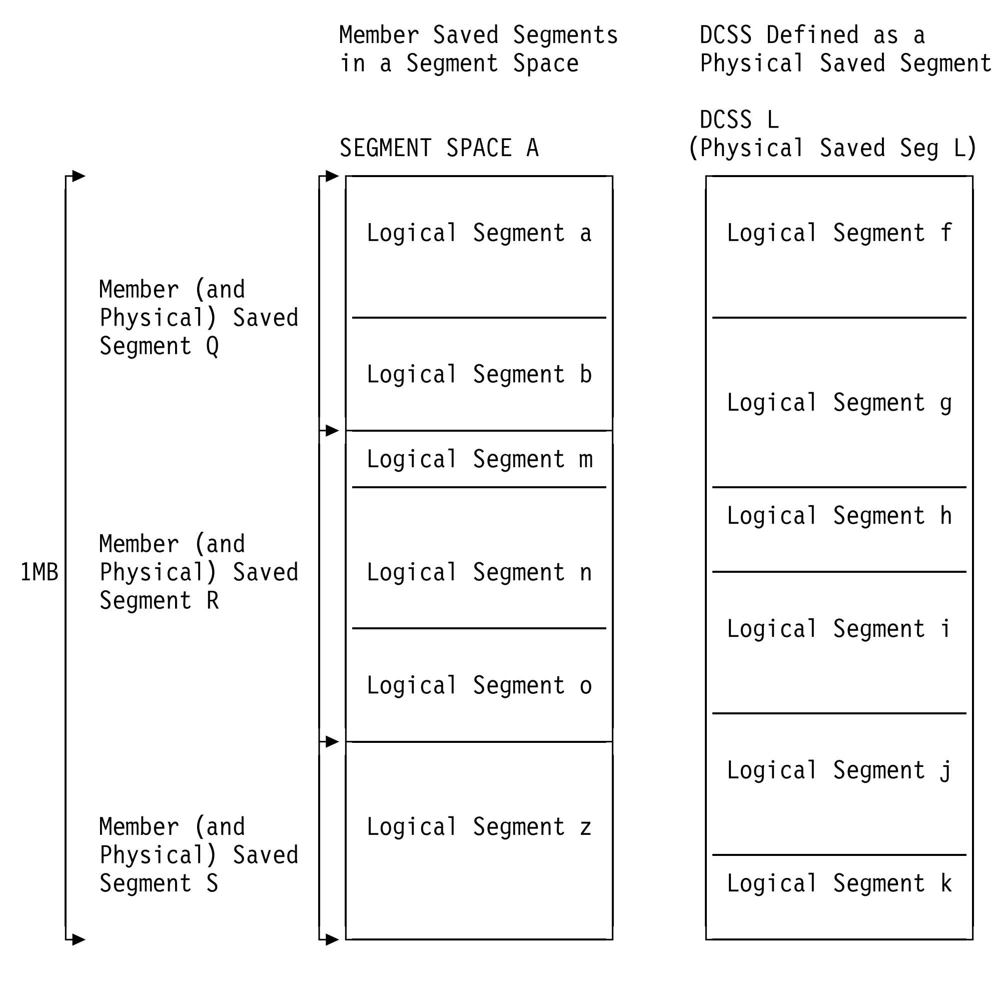


Figure 31. Member Saved Segments Versus DCSS

Defining Saved Segments in VM/ESA 2.4.0 CP: In VM/ESA 2.4.0, you build and save segment spaces, member saved segments, and DCSSs using the DEFSEG and SAVESEG commands. You build and save saved systems using the DEFSYS and SAVESYS commands. The basic DEFSEG command syntax is:

```
defseg dcssname operands [space spacename]
```

If the SPACE operand is used, *dcssname* is the name of a member saved segment you are defining in the segment space called *spacename*. If the SPACE operand is not used, you define a DCSS.

The basic syntax of the DEFSYS command is:

```
defsys name operands
```

The syntax of the SAVESEG and SAVESYS commands is:

```
saveseg dcssname
savesys name
```

Also, to alleviate the problem of having fewer architected segments, VM/ESA 2.4.0 allows segment packing. Segment packing allows you to conserve storage by placing more than one application (a licensed program or other code or data) into an architected segment.

Without segment packing, a great deal of storage would be wasted.

Figure 32 shows how applications in VM/ESA 1.1.5 370 Feature saved segments could be placed in VM/ESA 2.4.0 segments without segment packing and with segment packing.

For more information on segment packing, see *VM/ESA: Planning and Administration*.

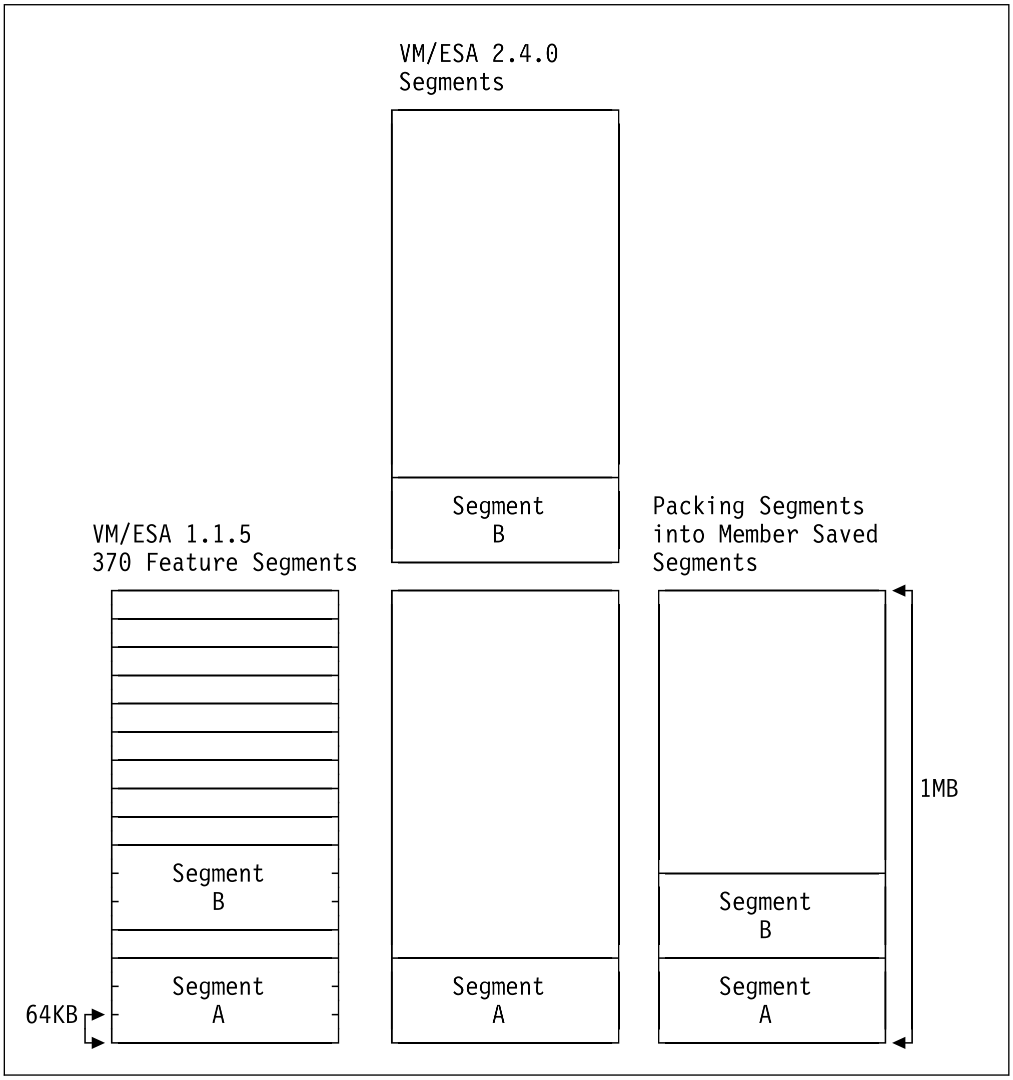


Figure 32. Packing VM/ESA 1.1.5 370 Feature DCSSs and Physical Saved Segments into a VM/ESA 2.4.0 Segment Space. Note that segments packed into the same 1MB area must be the same type (either all exclusive or all shared).

Converting DCSSs

DCSSs in VM/ESA 2.4.0 must be 1MB or multiples of 1MB in size. In VM/ESA 1.1.5 370 Feature, they were 64KB or multiples of 64KB. Consider packing your DCSSs in a segment space as member saved segments. Although segment spaces must be 1MB or multiples of 1MB, member saved segments are defined on page boundaries.

Figure 33 on page 263 shows how saved segments are defined and saved in VM/ESA 2.4.0.

Defining Saved Segments in VM/ESA 2.4.0 CMS: In VM/ESA 2.4.0, you build and save physical saved segments and their logical saved segments the same as you did in VM/ESA 1.1.5 370 Feature. To create and use a physical saved segment, however, you have to define a member saved segment or a DCSS for it using the DEFSEG command. Your PSEG file's file name is the name of the member saved segment or the DCSS.

- Issue the DEFSEG command for the member saved segment or the DCSS.
- Create the PSEG file with the same file name as the name of the member saved segment or the DCSS. The PSEG file contains the names of the LSEG files that define the logical segments making up the physical saved segment.
- Create the LSEG files defining the contents of the logical saved segments.

Once you have completed the previous steps, you can issue the SEGGEN command using the file name of the PSEG file (which is the name of the member saved segment or the DCSS):

You do not have to use SAVESEG to save the saved segments because SEGGEN issues SAVESEG for you.

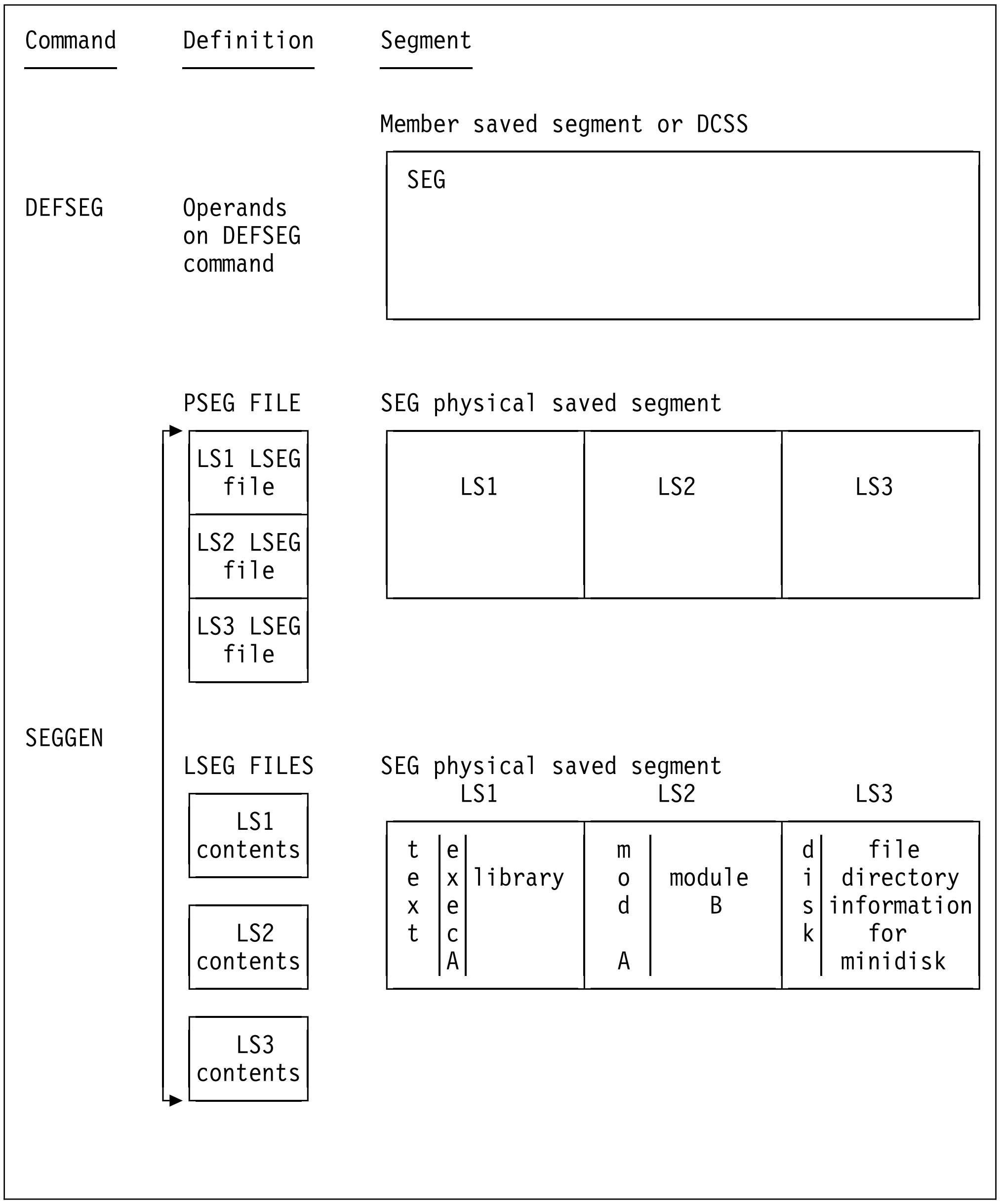


Figure 33. Defining and Saving Saved Segments in VM/ESA 2.4.0. This is done by using the DEFSEG and SEGGEN commands.

Mapping and Defining Saved Segments Simplified

You will probably find it necessary to remap your saved segments because of the fact that the size of segment spaces and DCSSs in VM/ESA 2.4.0 is 1MB or multiples of 1MB. You may want to build several of your physical saved segments and DCSSs as member saved segments in a single segment space.

VM/ESA 2.4.0 provides a saved segment mapping tool, VMFSGMAP EXEC, that allows you to display and customize your saved segment layout. Once you have defined your saved segments using this exec, you then use VMFBLD to build your saved segments. The *VM/ESA: Planning and Administration* book describes VMFSGMAP and how to use it and VMFBLD. “Planning Your System's Saved Segment Layout [S370]” on page 81 shows:

- How to plan your saved segment layout
- How to convert your VM/ESA 1.1.5 370 Feature saved segment definitions into a DEFSEG command

Conversion Inhibitors

The factors that may cause problems when you convert your saved segments to ESA/370 or ESA/390 architecture are:

- Applications that require exclusive write segments
- Applications with installation procedures that use DCSSGEN or SAVEFD

Applications Requiring Exclusive Write Segments

Some applications that run in a saved segment have to be defined in an exclusive write segment. Because each 1MB architected segment must be defined entirely as shared or entirely as exclusive, an application that requires only one page of exclusive write storage must reside in a separate segment from any applications requiring shared storage.

This problem was not as severe in System/370 architecture because the segment sizes were smaller.

Applications with Installation Procedures That Use DCSSGEN or SAVEFD

The installation procedures for some applications require the attachment of a saved segment. The installation procedures may, for example, use the DCSSGEN or SAVEFD commands to do this. DCSSGEN installs a saved segment that contains execs and XEDIT macros. SAVEFD installs a saved segment that contains minidisk file directory information. Because DCSSGEN and SAVEFD use certain VM/ESA 1.1.5 370 Feature interfaces, VM/ESA 2.4.0 requires both a skeleton and an active saved segment before using these commands. VM/ESA 1.1.5 370 Feature required only a skeleton saved segment before using these commands.

To make sure you have both a skeleton and an active saved segment, do the following before using DCSSGEN or SAVEFD:

- If you are installing the saved segment for the first time, or if you are changing the page ranges associated with the saved segment:
  - Enter the DEFSEG and SAVESEG commands for the saved segment to create a “dummy” version. This gives you the active saved segment you need for DCSSGEN and SAVEFD to work in VM/ESA 2.4.0.
  - Enter the DEFSEG command again. This gives you the skeleton file always required by DCSSGEN and SAVESEG.
  - Then enter the DCSSGEN or SAVEFD command.

  This process is shown in Figure 34 on page 265.

```
VM/ESA 1.1.5 370 Feature:                   VM/ESA 2.4.0:
DEFSEG (skeleton created)   DEFSEG  (skeleton created)
   |                           |
   V                           V
DCSSGEN                     SAVESEG (dummy active created)
   or  (active created)        |
SAVEFD                         V
                            DEFSEG  (skeleton created)
                               |
                               V
                            DCSSGEN
                               or   (active segment created)
                            SAVEFD
```

Figure 34. Using DCSSGEN or SAVEFD to create a new saved segment

- If you are not changing the page ranges associated with the saved segment, just enter the DEFSEG command to create a skeleton file. Because the saved segment exists, you already have an active saved segment.

Note: In VM/ESA 1.1.5 370 Feature, DCSSGEN and SAVEFD were used to install the CMSINST and HELP saved segments. This is no longer true in VM/ESA 2.4.0. Instead, CMSINST is defined as a logical saved segment in the INSTSEG physical saved segment, which is defined as a DCSS. HELP is defined as a logical saved segment in the HELPSEG physical saved segment, which is also defined as a DCSS.

If you are installing a saved segment using either of these commands, see the *VM/ESA: CMS Command Reference* for more information. For information about installing the HELP and CMSINST saved segments, see the *VM/ESA: Installation Guide*. For information about changing the definitions of installed saved segments, see the *VM/ESA: Planning and Administration* book.

Consideration for Backing Out

When converting to VM/ESA 2.4.0, you should maintain the DMKSNT file or SNT OVERRIDE file on the VM/ESA 1.1.5 370 Feature SYSRES in case you need to back out to VM/ESA 1.1.5 370 Feature. The VM/ESA 2.4.0 saved segments and saved systems are kept in system data files and cannot be transferred to VM/ESA 1.1.5 370 Feature using SPTAPE. (You can move files from one VM/ESA 2.4.0 system to another VM/ESA 2.4.0 system using SPTAPE, but not from a VM/ESA 1.1.5 370 Feature system to a VM/ESA 2.4.0 system or from a VM/ESA 2.4.0 system to a VM/ESA 1.1.5 370 Feature system.)

Summary of How CP Saved Segment Support Differs in VM/ESA 2.4.0

Table 25 on page 266 provides a summary of how CP saved segment support differs between VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0.

Table 25. Summary of How CP Saved Segment Support Differs in VM/ESA 2.4.0

VM/ESA 1.1.5 370 Feature Saved Segment Support	VM/ESA 2.4.0 Saved Segment Support
You define saved segments during system generation using a system names table (DMKSNT) or while the system is running using an SNT OVERRIDE file.	You define saved segments while the system is running using CP commands.
Storage is divided into 64KB segments, each containing 16 pages of 4KB storage.	Storage is divided into 1MB segments, each containing 256 pages of 4KB storage.
You can store only one program in each 64KB segment. The program must be stored on a 64KB boundary.	You can store several programs in one or more 1MB segments as long as you do not store both shared code and nonshared (exclusive) code in the same 1MB segment. Each program must be stored on a page boundary.
You define the ranges of segments using decimal numbers. Numbers in messages and responses are in decimal.	You define the ranges of segments using hexadecimal numbers. Numbers in messages and responses are in hexadecimal.
You can resave the contents of an existing saved segment without redefining it as long as you are not changing the page ranges.	You must redefine a saved segment before you resave information into it.

HELPINST Saved Segment Replaced by HELPSEG and INSTSEG [1.2.0, 1.2.1, 1.2.2, 2.1.0]

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.

Restructuring User Classes [1.1.5]

In VM/ESA 2.4.0, as in VM/ESA 1.1.5 370 Feature, if you want to give a CP command, DIAGNOSE code, or system function a class definition that is different than the IBM-defined class, you can create a class override file.

The format for the class override file (CMS source file) is different in VM/ESA 2.4.0. Also, the internal user class restructure file is kept in a system data file in VM/ESA

2.4.0 rather than, as in VM/ESA 1.1.5 370 Feature, in the OVRD space allocated on the same volume as the directory. System data files reside in spool space, but unlike reader, printer, and punch files that also reside in spool space, they are not destroyed by a cold start.

In addition, there are differences among the individual commands with respect to function, invocation, response, and privilege class. In VM/ESA 2.4.0, the user class restructure feature (UCR) is expanded to allow an installation to change the privilege classes assigned to each operand of the SET and QUERY commands. Before you convert your VM/ESA 1.1.5 370 Feature class override file to the VM/ESA 2.4.0 class override file, you should review “CP Commands” on page 315 for each command to verify that the command still exists, that the privilege class is the same, and that you still want to override the command in the same manner in VM/ESA 2.4.0 as you did in VM/ESA 1.1.5 370 Feature.

Note that in VM/ESA 2.4.0, use of Diagnostic CCW commands requires the MAINTCCW parameter on the OPTION user directory control statement in the user directory.

Class Override File for VM/ESA 1.1.5 370 Feature

The class override file consists of OVERRIDE control statements for each command or DIAGNOSE code whose IBM-defined privilege class is to be overridden.

VM/ESA 1.1.5 370 Feature commands, DIAGNOSE codes, and system functions have an IBM-defined type or multiple types. This is what you indicate on the TYPE parameter. Commands, DIAGNOSE codes, and system functions also have a class, which you can modify. You assign this using the CLASS parameter.

This is an example of a VM/ESA 1.1.5 370 Feature class override file:

```
/*                */
:COMMAND.ACNT     TYPE=O   CLASS=D
:COMMAND.ADSTOP   TYPE=G   CLASS=IJK
:COMMAND.ATTN     TYPE=G   CLASS=IJK
:COMMAND.AUTOLOG  TYPE=O   CLASS=FGI
:COMMAND.CHANGE   TYPE=S   CLASS=FG
:COMMAND.CHANGE   TYPE=G   CLASS=IJKL
```

In the example, notice that the CHANGE command has a TYPE=S version and a TYPE=G version. The TYPE attribute is used to distinguish between the two versions.

Class Override File for VM/ESA 2.4.0

The VM/ESA 2.4.0 class override file consists of an OVERRIDE control statement for each command, DIAGNOSE code, or system function whose IBM-defined privilege class is to be overridden.

Unlike VM/ESA 1.1.5 370 Feature, VM/ESA 2.4.0 commands, DIAGNOSE codes, and system functions are not associated with a user's group function. As a result, there is no TYPE attribute. VM/ESA 2.4.0 commands, DIAGNOSE codes, and system functions do have an IBM-defined class. You redefine this class using the NEWCLASS attribute. Refer to *VM/ESA: Planning and Administration* for examples.

Some VM/ESA 2.4.0 commands and system functions have more than one IBM-defined user class. To distinguish between the different classes, you must use the IBMCLASS attribute. For commands that have only one IBM-defined user class, you may use the IBMCLASS attribute, but you are not required to use it.

Here is an example of a VM/ESA 2.4.0 class override file:

```
*
*
*
*
ACNT     NEWCLASS=D
ATTN     NEWCLASS=HI  IBMCLASS=G
AUTOLOG  NEWCLASS=P   IBMCLASS=A
CHANGE   NEWCLASS=EF  IBMCLASS=D
CHANGE   NEWCLASS=HI  IBMCLASS=G
```

In the example, note that the ADSTOP command was deleted because it is not supported in VM/ESA 2.4.0. The ACNT command has only one IBM-defined class; thus, you can code the IBMCLASS attribute, but it is not required. The CHANGE command, on the other hand, has multiple versions; thus, you must code the IBMCLASS attribute to distinguish between the versions.

Comparison of VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0 Override File Parameters

Table 26 shows the differences between the VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0 override file parameters.

Table 26 (Page 1 of 2). Comparison of VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0 Override File Parameters. All VM/ESA 1.1.5 370 Feature and all VM/ESA 2.4.0 parameters are listed.

VM/ESA 1.1.5 370 Feature Override Parameter	VM/ESA 2.4.0 Counterpart	Differences
DEST	———	No equivalent parameter.
CLASS	NEWCLASS	No difference other than the parameter name.

Table 26 (Page 2 of 2). Comparison of VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0 Override File Parameters. All VM/ESA 1.1.5 370 Feature and all VM/ESA 2.4.0 parameters are listed.

VM/ESA 1.1.5 370 Feature Override Parameter	VM/ESA 2.4.0 Counterpart	Differences
TYPE	IBMCLASS	IBMCLASS means the IBM-defined user class. It is used somewhat differently from TYPE. In VM/ESA 1.1.5 370 Feature, you use TYPE when the command you are overriding has multiple privilege classes *and* when the command functions differently between those classes. If the command functions the same way between all the classes, you do not need to code the TYPE operand. In VM/ESA 2.4.0, you *must* use IBMCLASS when the command has multiple privilege classes whether or not the command functions differently between those classes. You can code only one of these classes on each IBMCLASS operand. If the command has only one privilege class, you may use IBMCLASS, but you are not required to use it. In VM/ESA 2.4.0, you use IBMCLASS only to define what is changing from the IBM default. In VM/ESA 1.1.5 370 Feature, you use TYPE to indicate both the default and what is changing. In VM/ESA 1.1.5 370 Feature, TYPE identifies a user's group function: O, R, P, S, A, C, or G. In VM/ESA 2.4.0, IBMCLASS identifies the IBM-defined command class: A, B, C, D, E, F, or G.

Converting Your VM/ESA 1.1.5 370 Feature Class Override File

To restructure user classes in VM/ESA 2.4.0, you must:

1. Create the class override file as shown in “Class Override File for VM/ESA 2.4.0” on page 267. You may decide to modify your VM/ESA 1.1.5 370 Feature class override file for your VM/ESA 2.4.0 system. Refer to “CP Commands” on page 315 to help you decide.

   Once you convert your VM/ESA 1.1.5 370 Feature override file, determine whether you want to override any new VM/ESA 2.4.0 commands. If so, add these to the file.

2. Invoke the OVERRIDE utility with the VALIDATE option to verify the syntax of the class override file.

3. Invoke the OVERRIDE utility with one of these options:

   - IMMEDIATE causes your user-defined class structure to take effect immediately. For example,

     ```
     override class override (immediate
     ```

   - DEFER causes your user-defined class structure to take effect at the next system IPL. This is the default option for the OVERRIDE utility. For example,

     ```
     override class override (defer
     ```

4. If you change your directory to use the new or changed classes, issue DIRECTXA to bring your updated directory online. For example,

   ```
   directxa user direct
   ```

To reset your class structure to the IBM-defined classes, enter the OVERRIDE utility with the CLEAR option. The resetting takes effect at the next system IPL.

Querying or Purging User Class Restructure Files

Because VM/ESA 2.4.0 user class restructure files are kept in system data files rather than on CP-owned DASDs, you must handle them differently.

VM/ESA 2.4.0 provides two commands for you to query and purge user class restructure files: QUERY UCR and PURGE UCR.

Refer to *VM/ESA: Planning and Administration* for a complete explanation of the user class restructure function in VM/ESA 2.4.0.

Managing System Performance [1.1.5]

The sections below address the following items that you might want to consider when managing system performance:

- The scheduler
- Tuning commands
- Paging and swapping differences
- The monitor

You may also want to see *VM/ESA: Performance* for more information on VM/ESA performance.

VM/ESA 2.4.0 Scheduler

Although the VM/ESA 2.4.0 scheduler incorporates some of the same concepts as the VM/ESA 1.1.5 370 Feature scheduler, it has a completely different design. The following list highlights a few of the differences and similarities.

- As in VM/ESA 1.1.5 370 Feature, the VM/ESA 2.4.0 scheduler places users in lists:

  – The dormant list (does not exist in VM/ESA 1.1.5 370 Feature)
  – The eligible list
  – The dispatch list (called the run list in VM/ESA 1.1.5 370 Feature)

- As in VM/ESA 1.1.5 370 Feature, the VM/ESA 2.4.0 scheduler classifies virtual machines in transaction classes. However, the VM/ESA 2.4.0 scheduler moves users from one transaction class to another based on real elapsed time rather than processor time. As in VM/ESA 1.1.5 370 Feature, the following transaction classes are used:

  E1, Q1 are users with short-running transactions.

  E2, Q2 are users with medium-running transactions.

  E3, Q3 are users with long-running transactions.

  In addition, the VM/ESA 2.4.0 scheduler classifies some users as **E0, Q0**. These users do not wait in the eligible list but go directly into the dispatch list.

- As in VM/ESA 1.1.5 370 Feature, VM/ESA 2.4.0 uses the concept of interactive bias. In VM/ESA 2.4.0, these concepts extend not only to storage but also to paging devices and to the multiprogramming level in general.

VM/ESA 2.4.0 Tuning Commands

The different scheduler design significantly changes the external interfaces for the CP commands that control the way the system allocates resources.

The VM/ESA 2.4.0 SET SRM command has the following operands:

- IABIAS specifies the new interactive bias to take effect. The interactive bias causes users doing trivial transactions to receive better service than their scheduling share would otherwise entitle them to receive.
- DSPBUF specifies the dispatch buffer to take effect. The dispatch buffer sets a limit on the number of users in each transaction class who are allowed in the dispatch list.
- DSPSLICE is the new size, in milliseconds, of the dispatching minor time slice. This is the amount of processor time a virtual machine may consume before reexamination by the scheduler. The default is computed at CP initialization; it is proportional to processor speed.
- LDUBUF partitions the commitment of the system's paging resources when choosing users to move from the eligible list to the dispatch list. The result is a mechanism that controls paging capacity based on the transaction class of a user.
- STORBUF partitions the scheduler's view of main storage when choosing users to move from the eligible to dispatch lists. The result is a mechanism that controls use of main storage based on the transaction class of a user.
- MAXWSS specifies the maximum working set that a virtual machine is allowed to have. It is specified as a percentage of the system's pageable storage. It is effective only when an eligible list already exists.
- XSTORE lets you specify the percentage of expanded storage you want included in the calculation of available storage.

Because the VM/ESA 2.4.0 scheduler differs from the VM/ESA 1.1.5 370 Feature scheduler, there is no one-to-one correlation between the tuning commands of the two systems. In some cases, however, the intent or use of commands does correlate. Table 27 presents VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0 commands that serve a similar purpose.

Table 27 (Page 1 of 2). Comparative Uses of VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0 Tuning Commands

If You Use This VM/ESA 1.1.5 370 Feature Command:	**You Might Consider Using This VM/ESA 2.4.0 Command:**
SET PRIORITY	SET SHARE *userid* RELATIVE (or SHARE RELATIVE user directory control statement) Although the implementation is different, SHARE RELATIVE performs a similar function to SET PRIORITY in that it controls the amount of service the user receives in comparison to other users. Note that a SET PRIORITY value of 1 is highest priority (best possible) while a SHARE RELATIVE value of 1 is lowest priority (worst possible). **Note:** You can still specify the *pri* option on the USER directory control statement but it has no function in VM/ESA 2.4.0.

Table 27 (Page 2 of 2). Comparative Uses of VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0 Tuning Commands

If You Use This VM/ESA 1.1.5 370 Feature Command:	You Might Consider Using This VM/ESA 2.4.0 Command:
SET FAVOR %	SET SHARE *userid* ABSOLUTE (or SHARE ABSOLUTE user directory control statement) Although the implementation is different, SHARE ABSOLUTE performs a similar function to SET FAVOR % in that it controls the amount of resources virtual machines receive. As is recommended with the SET FAVOR % command in VM/ESA 1.1.5 370 Feature, you should use SHARE ABSOLUTE with discretion to avoid having the total shares exceed 100%. This would cause the scheduler to attempt to deliver more resources than actually exist.
SET FAVOR	SET QUICKDSP (or OPTION QUICKDSP user directory control statement) SET QUICKDSP is like SET FAVOR in that it allows the specified virtual machine to bypass the eligible list and go directly into the dispatch list. The SET QUICKDSP user is classified Q0. Like the SET FAVOR user in VM/ESA 1.1.5 370 Feature, the dispatch priority is not improved. In VM/ESA 1.1.5 370 Feature, there is no way to reserve storage for all the SET FAVOR users. As a result, storage can be over-allocated if you use SET FAVOR too often. In VM/ESA 2.4.0, you can use the SET SRM STORBUF command to reserve storage for the SET QUICKDSP users.
SET SRM IB	SET SRM IABIAS Although the implementation is completely different, the intent of the VM/ESA 2.4.0 interactive bias is the same as the VM/ESA 1.1.5 370 Feature interactive shift bias: to give good response time to trivial transactions that only require a small fraction of a second of processing to complete.
SET SRM DSPSLICE	SET SRM DSPSLICE As in VM/ESA 1.1.5 370 Feature, the DSPSLICE is the size, in milliseconds, of the dispatching minor timeslice.
SET SRM MAXWSS	SET SRM MAXWSS As in VM/ESA 1.1.5 370 Feature, MAXWSS specifies the maximum working set that a virtual machine is allowed to have. However, in VM/ESA 2.4.0 the value (number) you specify is interpreted as a percentage of storage. In VM/ESA 1.1.5 370 Feature, it is interpreted as a number of pages.
SET RESERVE	SET RESERVED As in VM/ESA 1.1.5 370 Feature, the VM/ESA 2.4.0 SET RESERVED command keeps a specified number of pages resident in real storage at all times for a particular user. Also, it is more effective than the VM/ESA 1.1.5 370 Feature SET RESERVE.
SPMODE	DEDICATE If you have a multiprocessor configuration but are running in uniprocessor mode, the VM/ESA 1.1.5 370 Feature SPMODE command allows you to grant the sole use of the other processor to the MVS or OS/390 V=R virtual machine. The VM/ESA 2.4.0 DEDICATE command is similar in that it allows you to grant the sole use of a real processor to a virtual machine. Note that one V=R virtual processor is dedicated when the V=R virtual machine logs on.

The following VM/ESA 2.4.0 commands have no equivalent function in VM/ESA 1.1.5 370 Feature:

SET SRM DSPBUF
SET SRM LDUBUF

Because of the differences in the VM/ESA 2.4.0 scheduler and paging algorithms, the following VM/ESA 1.1.5 370 Feature commands have no equivalent function in VM/ESA 2.4.0:

```
SET PAGING
SET QDROP OFF
SET MAXDRUM
SET SRM MINNUMSS
SET SRM PB
SET SRM PCI
```

RETAIN XSTORE MDC is a new command in VM/ESA 2.4.0 that affects the partitioning of expanded storage between minidisk cache and paging.

Tuning the VM/ESA 2.4.0 System

Table 28 presents different problems you may face at your installation and the appropriate tuning controls (CP commands) that apply to VM/ESA 2.4.0. For details on tuning the VM/ESA 2.4.0 system, refer to the *VM/ESA: Performance* book. For the syntax of the VM/ESA 2.4.0 commands, refer to the *VM/ESA: CP Command and Utility Reference*.

The controls for each environment are listed in approximate order of importance, although in some environments, the order of importance is difficult to predict.

Table 28 (Page 1 of 2). Summary of Tuning Controls for VM/ESA 2.4.0

Situation	Appropriate Tuning Controls for VM/ESA 2.4.0
Processor-constrained environment	1. UNDEDICATE 2. SET SRM DSPBUF 3. SET SRM DSPSLICE 4. SET SRM IABIAS
Storage-constrained environment	1. SET SRM STORBUF - you may have to adjust the default setting for better performance, especially in a guest environment. 2. SET RESERVED
Paging-constrained environment	1. SET SRM LDUBUF - you may have to adjust the default setting for better performance, especially in a guest environment. 2. SET RESERVED
I/O-constrained environment	1. SET SRM DSPBUF 2. SET SRM DSPSLICE 3. SET SRM IABIAS
Need improved performance for a production guest	1. SET SHARE 2. V=R or V=F 3. DEDICATE 4. SET IOASSIST 5. SET CCWTRAN 6. SET RESERVED

Table 28 (Page 2 of 2). Summary of Tuning Controls for VM/ESA 2.4.0

Situation	Appropriate Tuning Controls for VM/ESA 2.4.0
Need improved performance for a service virtual machine	1. SET QUICKDSP 2. SET SHARE 3. SET RESERVED

You can combine spooling and minidisk I/O on the same control unit.

VM/ESA 2.4.0 Monitor

The SYSMON macro, used in VM/ESA 1.1.5 370 Feature to specify monitor collection parameters, does not exist in VM/ESA 2.4.0. In VM/ESA 2.4.0, all collection parameters are controlled by the MONITOR command. This eliminates the need for updating HCPSYS and reassembling and re-IPLing the system whenever you want to change monitoring parameters.

The MONITOR command has been expanded in VM/ESA 2.4.0 to allow an increased selectivity in monitoring; that is, users or devices can be monitored on an individual basis. The data categories for monitoring are called domains rather than classes.

The data domains are: APPLDATA, MONITOR, I/O, PROCESSOR, USER, SCHEDULER, SEEKS, STORAGE, and SYSTEM. The SYSTEM domain has been designed to include enough overall system performance data such that other domains do not have to be enabled unless problems are suspected.

Another difference between the VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0 MONITOR commands is in the way that the enable commands work. In VM/ESA 1.1.5 370 Feature, each successive monitor enable command overrides the previous one. In VM/ESA 2.4.0, on the other hand, each successive monitor enable command builds on the previous command. This means that the enabling of data is cumulative in VM/ESA 2.4.0.

The VM/ESA 1.1.5 370 Feature MONITOR DISPLAY command is replaced by the VM/ESA 2.4.0 QUERY MONITOR command. In addition, VM/ESA 2.4.0 provides the SET MONDATA command to control the collection of terminal input/output data for security reasons.

Unlike the VM/ESA 1.1.5 370 Feature monitor, which places monitor data in a spool file, the VM/ESA 2.4.0 monitor places data in a saved segment. You must define and save the saved segment before you can start collecting monitor data. Once you save the saved segment, a user can load it and enter an IUCV CONNECT to the *MONITOR CP system service. The *MONITOR system service will notify the virtual machine when there is data to be processed.

The VM/ESA 1.1.5 370 Feature monitor can send monitor data directly to tape. The VM/ESA 2.4.0 monitor can also send data to tape, but not directly; you have to use the MONWRITE utility to copy the data from the monitor saved segment to tape. (You can also use MONWRITE to copy the data to disk.)

Up to 65535 virtual machines can establish connections to *MONITOR. The MONITOR command can be entered at any time by any privileged user.

To establish a MONITOR saved segment and start generation of MONITOR data:

1. Create a segment for MONITOR using the DEFSEG command. The saved segment must be at least 11 pages.
2. Save the segment for MONITOR using the SAVESEG command.
3. Access the MONITOR saved segment from a virtual machine by entering the CMS SEGMENT command, SEGMENT macro, or DIAGNOSE code X'64'.
4. Establish IUCV communication with *MONITOR through IUCV CONNECT.
5. Enter MONITOR commands to set up and start collecting system monitor data.

VM/ESA 2.4.0 provides a utility called MONWRITE that loads the saved segment, connects to *MONITOR, and writes the raw monitor data to disk or tape.

For more information about the VM/ESA 2.4.0 monitor, see the *VM/ESA: Performance* book.

For details on the layout of the monitor data in VM/ESA 2.4.0 see the file, MONITOR LIST1403. This file contains the most current list of monitor records for your system. MONITOR LIST1403 was loaded onto your base CP object disk (194) at the time VM/ESA was installed on your system. It is already formatted for printing and can be printed directly or read online. You can find instructions on how to do this in the *VM/ESA: Performance* book.

Performance Guidelines

The *VM/ESA: Performance* book includes a set of performance tuning guidelines to help you set up your VM/ESA 2.4.0 system.

Allocation of Real Storage for Segment Tables Has Changed [1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

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.

In VM/ESA 2.4.0, 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.

Setting Up the Programmable Operator [1.1.5]

You should update the programmable operator routing table and action routines so they are sensitive to the changes in the CP interfaces:

- There are command changes. Refer to “CP Commands” on page 315.
- There are message changes. Some messages that use only one line in VM/ESA 1.1.5 370 Feature use more than one line in VM/ESA 2.4.0. In some cases, multiple VM/ESA 1.1.5 370 Feature messages are replaced by one VM/ESA 2.4.0 message. Refer to “CP Messages” on page 383 for details.

The programmable operator works the same way in VM/ESA 2.4.0 as it did in VM/ESA 1.1.5 370 Feature. However, there are several changes to programmable operator commands and routing table statements. See “Programmable Operator Facility Commands and Routing Table Statements” on page 453 for information about the changes.

See *VM/ESA: Planning and Administration* for a complete description of the programmable operator facility.

CMSGEND Exec No Longer Generates PROPLIB LOADLIB [1.1.5]

The CMSGEND exec is no longer used to generate the PROPLIB LOADLIB from updated text files.

PROPLIB LOADLIB is built by the VMSES/E VMFBLD exec using the DMSBLPRP build list. If you need to make any changes to the supplied action routines, you must create local modifications to those routines. If you want to add your own routines, you must provide a local modification to the DMSBLPRP build list. Refer to the discussion on installing local service in the *VM/ESA: Service Guide* for information on creating local modifications.

Setting Password Suppression [1.1.5]

In VM/ESA 1.1.5 370 Feature and in VM/ESA 2.4.0, you can set and change password suppression by using the SYSJRL macro during system generation. In VM/ESA 2.4.0, but not in VM/ESA 1.1.5 370 Feature, you can also set and change password suppression by using the SET PASSWORD command when the system is running, and you can query the setting by using the QUERY PASSWORD command.

In VM/ESA 1.1.5 370 Feature, you set and change password suppression for the LOGON, LINK, and AUTOLOG commands as a group. In VM/ESA 2.4.0, you enter SET PASSWORD and QUERY PASSWORD for the LOGON, LINK, and AUTOLOG commands individually.

Setting Up Virtual Machines for Accounting and Error Recording [1.1.5]

In VM/ESA 1.1.5 370 Feature, accounting data is collected in spool files and error recording (EREP) data is collected on a volume reserved for that purpose in DMKSYS. In VM/ESA 2.4.0, accounting data and EREP data (the latter now called LOGREC data) are collected by service virtual machines. Remember to process data on the minidisks from time to time.

To define the virtual machines that you want to use to collect accounting and EREP data, use the SYSTEM_USERIDS statement in your SYSTEM CONFIG file.

Table 29 contains changed accounting records.

Table 29 (Page 1 of 2). Changes to Accounting Records

Accounting Records for:	Release	Columns	Contents
Virtual Machine Resource Usage (Record type 01)	370F	49-52	Number of completed SIO instructions
		65-78	Reserved
	2.4.0	49-52	Number of requested SIO instructions
		65-68	Milliseconds of total Vector time
		69-72	Milliseconds of virtual Vector time
		73-78	Reserved
Dedicated Devices (Record type 02)	370F	33-36	Device class, type, model, and feature information
		37-64	Unused
		65-72	Terminal identification
		73-78	Unused
	2.4.0	33-36	Device class, type, model, and feature values in these columns may have changed.
		37-78	Reserved
Temporary Disk Space (Record type 03)	370F	33-36	Device class, type, mode, and feature information
		41-78	Unused
	2.4.0	33-36	Device class, type, model, and feature values in these columns may have changed.
		41-44	The number of 4KB pages used.
		45-78	Reserved
Accounting Record for Journaling (Record type 04)	370F	29-32	String *DEAF* if the terminal is an SNA terminal
	2.4.0	29-32	String *SNA* if the terminal is an SNA terminal

Table 29 (Page 2 of 2). Changes to Accounting Records

Accounting Records for:	Release	Columns	Contents
Accounting Records for Journaling (Record type 05)	370F	29-32	String *DEAF* if the terminal is an SNA terminal
		49-51	Minidisk address being linked to in character format
		52-70	Reserved
	2.4.0	29-32	String *SNA* if the terminal is an SNA terminal
		49-52	Minidisk address being linked to in hexadecimal format
		53	Type of minidisk link
		54-55	Reserved
		56-57	Blank
		58-70	Reserved
Accounting Records for Journaling (Record type 06)	370F	29-32	String *DEAF* if the terminal is an SNA terminal
		49-51	Minidisk address being linked to in character format
		56-70	Reserved
	2.4.0	29-32	String *SNA* if the terminal is an SNA terminal
		49-51	Reserved
		56	Blank
		57-60	Minidisk address being linked to in hexadecimal format
		61-70	Reserved

See *VM/ESA: Planning and Administration* for a complete description of how to set up service virtual machines.

CP Changes for TCP/IP Awareness [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

There have been changes to CP accounting records and to the access control interface for TCP/IP awareness.

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 30 on page 279.

Table 30. Changes to Accounting Records for TCP/IP Awareness

Accounting Record	Release	Columns	Contents
Accounting Record for Journaling (Record type 04)	Old	57-70	Reserved
		71-78	LUNAME for SNA terminal
	2.4.0	57-62	Reserved
		63-70	Network qualifier for SNA terminal or host virtual machine name for TCP/IP terminal
		71-78	LUNAME for SNA terminal or IP address for TCP/IP terminal
Accounting Record for Journaling (Record type 05)	Old	58-70	Reserved
		71-78	LUNAME for SNA terminal
	2.4.0	58-62	Reserved
		63-70	Network qualifier for SNA terminal or host virtual machine name for TCP/IP terminal
		71-78	LUNAME for SNA terminal or IP address for TCP/IP terminal
Accounting Record for Journaling (Record type 06)	Old	61-70	Reserved
		71-78	LUNAME for SNA terminal
	2.4.0	61-62	Reserved
		63-70	Network qualifier for SNA terminal or host virtual machine name for TCP/IP terminal
		71-78	LUNAME for SNA terminal or IP address for TCP/IP terminal
Accounting Record for Journaling (Record type 08)	Old	29-64	Reserved
	2.4.0	29-48	Reserved
		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.

Changes to Full Pack Minidisks [1.1.5]

In VM/ESA 1.1.5 370 Feature, the documentation warned not to define overlapping minidisks. However, CP did not check for them. If you defined an overlapping minidisk, it was ignored. In VM/ESA 2.4.0, if you overlap minidisks, you may be:

- Prevented from linking to those minidisks and receive messages HCP104, HCP105, and HCP106, or

- Linked read-only to those minidisks and receive messages HCP101, HCP102, and HCP103

National Languages Supported [ALL]

The following table shows the national languages supported in VM/ESA 2.4.0:

Language	**Supported in Release?**							
	1.1.5	**1.2.0**	**1.2.1**	**1.2.2**	**2.1.0**	**2.2.0**	**2.3.0**	**2.4.0**
Mixed case American English	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Uppercase American English	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Brazilian Portuguese	Yes	No	No	No	No	No	No	No
Chinese Simplified (Hanzi)	Yes	Yes	Yes	Yes	Yes	Yes	No	No
French	Yes	Yes	Yes	No	*	*	*	*
Canadian French	Yes	Yes	Yes	No	*	*	*	*
German	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Japanese (including Kanji)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Spanish	Yes	Yes	Yes	Yes	No	No	No	No
Argentinian Spanish	Yes	No	No	No	No	No	No	No

***:** CP messages and CMS messages only.

Chapter 13. System Operation Changes and Conversion Considerations

System operation tasks include bringing up the system, starting and stopping the system, running the system (this consists of controlling processors, controlling I/O, and managing data files), shutting down the system, and running utility programs.

This chapter may contain information for any or all of the conversions documented in this book. For definitions of the conversion identifiers used in this chapter, see “How to Identify the Topics That Apply to Your Conversion” on page 3.

1.1.5

Some system operation tasks are performed differently in VM/ESA 2.4.0 than in VM/ESA 1.1.5 370 Feature. This chapter discusses some of them. There are also differences in commands and messages. See the compatibility tables for details.

This chapter discusses the following major topics:

- “Differences in Starting the System [1.1.5]”
- “Warm Start Changes [1.1.5]” on page 284
- “Clean Start Added [1.2.0, 1.2.1]” on page 284
- “Change for IPLing from a Saved Segment [1.1.5]” on page 284
- “Miscellaneous Operational Differences [1.1.5]” on page 284
- “System Message Change for Disconnect [1.2.0]” on page 285
- “Differences Logging Off or Disconnecting [1.1.5, 1.2.0]” on page 285

Differences in Starting the System [1.1.5]

VM/ESA 1.1.5 370 Feature provided four kinds of starts: warm, checkpoint, force, and cold. These starts could be modified by specifying DRAIN or SHUTDOWN. The VM/ESA 1.1.5 370 Feature prompt was:

```
START ((WARM|CKPT|FORCE|COLD) (DRAIN))|(SHUTDOWN)
```

VM/ESA 2.4.0 provides four kinds of starts: warm, force, cold, and clean. These starts can be modified by specifying DRAIN, DISABLE, NODIRECT, NOAUTOLOG, or SHUTDOWN. The VM/ESA 2.4.0 prompt is:

```
Start ((Warm|Force|COLD|CLEAN) (DRain) (DIsable) (NODIRect) (NOAUTOlog))
      or (SHUTDOWN)
```

If you IPLed VM/ESA 1.1.5 370 Feature with a warm start and an error occurred, the VM/ESA 1.1.5 370 Feature sent a message to the operator before going into a wait state. VM/ESA 2.4.0 notifies the operator how many spool files will be lost and gives the operator the option to continue or stop.

In VM/ESA 1.1.5 370 Feature, you used a checkpoint start when there were I/O errors or invalid data in the warm-start area. You used a force start when there were I/O errors or invalid data in the checkpoint area. In VM/ESA 2.4.0, you use a force start in both cases.

Cold starts are basically the same in VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0. However, VM/ESA 2.4.0 notifies the operator about how many spool files will be lost if the system is cold started and then gives the operator the option to continue or stop. In VM/ESA 1.1.5 370 Feature you did not receive the number of spool files that would be lost.

A clean start in VM/ESA 2.4.0 IPLs the system without attempting to recover spool files and system data files that existed prior to system shutdown.

The following sample shows the kind of messages you may receive during a cold start of VM/ESA 2.4.0.

```
10:04:20 VM/ENTERPRISE SYSTEMS ARCHITECTURE  V2 R4.0  SERVICE LEVEL  0000;
10:04:21 SYSTEM NUCLEUS CREATED ON 1999-01-25 AT 10:19:15, LOADED FROM 240RES
10:04:21
10:04:21 ****************************************************************
10:04:21 * LICENSED MATERIALS - PROPERTY OF IBM*                        *
10:04:21 *                                                              *
10:04:21 * 5654-030 (C) COPYRIGHT IBM CORP. 1983, 1999. ALL RIGHTS      *
10:04:21 * RESERVED. US GOVERNMENT USERS RESTRICTED RIGHTS - USE,       *
10:04:21 * DUPLICATION OR DISCLOSURE RESTRICTED BY GSA ADP SCHEDULE     *
10:04:21 * CONTRACT WITH IBM CORP.                                      *
10:04:21 *                                                              *
10:04:21 * * TRADEMARK OF INTERNATIONAL BUSINESS MACHINES.              *
10:04:21 ****************************************************************
10:04:21
10:04:21 HCPZCO6718I Using parm disk 1 on volume 240RES (device 154F).
10:04:21 HCPZCO6718I Parm disk resides on cylinders 2071 through 2170.
10:04:21
10:04:21 HCPISU951I CP VOLID ESASRV NOT MOUNTED
10:04:21 HCPISU951I CP VOLID ESAP01 NOT MOUNTED
10:04:21 HCPISU951I CP VOLID DUMPPK NOT MOUNTED
10:04:21 Start ((Warm|Force|COLD|CLEAN) (DRain) (DIsable)  (NODIRect)
10:04:21 (NOAUTOlog)) or (SHUTDOWN)
10:04:28 COLD
10:04:28 NOW 10:04:28 EST TUESDAY 1999-09-26
10:04:28 Change TOD clock (Yes|No)
10:04:42 NO
10:04:42 The directory on volume 240RES at address 154F has been brought online.
10:04:43 HCPWRS2513I
10:04:43 HCPWRS2513I Spool files available       25
10:04:43 HCPWRS2513I
10:04:43 HCPWRS2513I Spool files on offline volumes          NONE
10:04:43 HCPWRS2513I Spool files with I/O errors             NONE
10:04:43 HCPWRS2513I Spool files with control errors         NONE
10:04:43 HCPWRS2513I Spool files to be discarded                3
10:04:43 HCPWRS2513I                                      _______
10:04:43 HCPWRS2513I Total files to be deleted                  3
10:04:43 HCPWRS2511A
10:04:43 HCPWRS2511A Spool files will be deleted because of COLD start.
10:04:43 HCPWRS2511A No files have been deleted yet.
10:04:43 HCPWRS2511A To continue COLD start and delete files, enter GO.
10:04:43 HCPWRS2511A To stop COLD start without deleting files, enter STOP.
10:04:47 GO
10:04:48 HCPWRS2512I Spooling initialization is complete.
10:04:50 DASD 154F dump unit CP IPL pages 1740
10:04:50 LOGMSG - 13:13:51 EST Friday 09/25/98
10:04:50 *************************************************************
```

```
10:04:50 *                                                           *
10:04:50 *                                                           *
10:04:50 *************************************************************
10:04:50 *                                                           *
10:04:50 *  This System Is to Be Used For IBM Business Purposes Only *
10:04:50 *                                                           *
10:04:50 *************************************************************
10:04:50 FILES:   NO RDR,   NO PRT,   NO PUN
10:04:50 LOGON AT 10:04:50 EST TUESDAY 01/26/99
10:04:50 GRAF  0020 LOGON  AS  OPERATOR USERS = 1
10:04:50 HCPIOP951I USER VOLID 19ESP4 NOT MOUNTED
10:04:50 HCPIOP951I USER VOLID SMPRES NOT MOUNTED
10:04:50 HCPIOP952I 0024M system storage
10:04:50 FILES: 0000002 RDR, 0000001 PRT,      NO PUN
10:04:50 PRT 000E STARTED     SYSTEM   CLASS A
10:04:50 PRT 000E FORM STANDARD MANUAL SEP   NO3800  LIMIT NONE
10:04:50 PRT 000E DEST OFF      DEST - NONE - DEST - NONE - DEST - NONE -
10:04:50 PRT 000E AFP
10:04:50 PUN 000D STARTED     SYSTEM   CLASS A
10:04:50 PUN 000D FORM STANDARD MANUAL NOSEP LIMIT NONE
10:04:50 PUN 000D DEST OFF      DEST - NONE - DEST - NONE - DEST - NONE -
10:04:50 RDR 000C STARTED     SYSTEM
10:04:50 XAUTOLOG EREP
10:04:50 HCPRSR431E Reader 000C id card missing or invalid
10:04:50 Command accepted
10:04:50 XAUTOLOG DISKACNT
10:04:50 Command accepted
10:04:50 XAUTOLOG AUTOLOG1
10:04:50 AUTO LOGON  ***       EREP     USERS = 2     BY OPERATOR
10:04:50 AUTO LOGON  ***       DISKACNT USERS = 3     BY OPERATOR
10:04:50 Command accepted
10:04:50 XAUTOLOG OPERSYMP
10:04:50 Command accepted
10:04:50 AUTO LOGON  ***       AUTOLOG1 USERS = 4     BY OPERATOR
10:04:50 HCPCLS6056I XAUTOLOG information for EREP: The IPL command is verified
by the IPL command processor.
10:04:50 AUTO LOGON  ***       OPERSYMP USERS = 5     BY OPERATOR
10:04:50 HCPCLS6056I XAUTOLOG information for DISKACNT: The IPL command is verif
ied by the IPL command processor.
10:04:51 HCPCLS6056I XAUTOLOG information for AUTOLOG1: The IPL command is verif
ied by the IPL command processor.
10:04:51 HCPCLS6056I XAUTOLOG information for OPERSYMP: The IPL command is verif
ied by the IPL command processor.
VM/ESA V2.4.0 1999-01-26 13:04
10:04:58 AUTO LOGON  ***       VMSERVR  USERS = 6     BY AUTOLOG1
10:04:58 AUTO LOGON  ***       VMSERVS  USERS = 7     BY AUTOLOG1
10:04:58 AUTO LOGON  ***       VMSERVU  USERS = 8     BY AUTOLOG1
10:04:58 USER DSC   LOGOFF AS  AUTOLOG1 USERS = 7
10:04:58 HCPCRC8082I Accounting records are accumulating for userid DISKACNT.
10:05:04  * MSG FROM EREP    : 1 RECORDING FILE(S), 0 RECORDS, A DISK 1 4 % FUL
 L
10:05:05 HCPCRC8064I Recording data retrieval has been started; recording *LOGRE
C for userid EREP.
10:05:08 PRT 000E STARTED     SYSTEM   CLASS P
10:05:08 PRT 000E FORM STDN     AUTO   SEP   NO3800  LIMIT NONE
10:05:08 PRT 000E DEST OFF      DEST - NONE - DEST - NONE - DEST - NONE -
10:05:08 PRT 000E AFP
Ready; T=0.10/0.13 10:05:08
```

As in VM/ESA 1.1.5 370 Feature, you use DRAIN to suspend spooling activity to unit record devices and SHUTDOWN to cancel the process of bringing up the system. In addition, in VM/ESA 2.4.0 you can use DISABLE to keep displays from being enabled and NODIRECT to bring up the system without a user directory. The NODIRECT option is provided in VM/ESA 2.4.0 because there is no stand-alone directory program. In this case, CP starts the system without trying to access the user directory. Once the system is up, you can recover the user directory. The NOAUTOLOG option lets you indicate that you do not want the service machines to be autologged. See the *VM/ESA: System Operation* for information on recovering your user directory.

In VM/ESA 1.1.5 370 Feature, printer spool files marked to be purged were recovered during a warm or force start. In VM/ESA 2.4.0, they are purged.

In VM/ESA 2.4.0, spool files and system data files are lost if the spooling volume on which they reside is not mounted. Also, spool files and system data files may be lost if the SYSCPVOL list is altered in any way other than appending to it.

Warm Start Changes [1.1.5]

The FOLD/NOFOLD status of an impact printer is maintained over a system shutdown and warm start re-IPL.

Clean Start Added [1.2.0, 1.2.1]

VM/ESA now provides a fourth type of start: CLEAN. A clean start IPLs the system without attempting to recover spool files and system data files that existed prior to system shutdown. Clean start restores the least amount of the system's environment.

IPLing with the NODIRECT Option [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

When you IPL your VM/ESA 2.4.0 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. (On VM/ESA 2.2.0 systems, this change may have already been made through an APAR.)

Change for IPLing from a Saved Segment [1.1.5]

If you are IPLing from a saved segment, a virtual device, or your reader when the S-disk or the Y-disk is not linked, warning message DMS2015W is issued. In VM/ESA 1.1.5 370 Feature, no warning message was issued.

Miscellaneous Operational Differences [1.1.5]

The processor alarm sounds more often with VM/ESA 2.4.0 than with VM/ESA 1.1.5 370 Feature.

VM/ESA 2.4.0 records soft and hard CP abend dumps. A soft abend dump is always written to a system data file. You define the recipient user ID of the system data file at system generation. A hard abend dump is written to printer, tape, or DASD. You define the dump device using the SET DUMP command.

A new installation-wide exit assemble module, HCPMSU, allows customization of the message command functions, MESSAGE, MSGNOH, SMSG, and WARNING. Local CP modifications based on earlier message-processing code no longer work. For information on the exit, see *VM/ESA: Planning and Administration*.

Also see "Difference in Transferring a Print File [1.1.5]" on page 299, which describes the differences between VM/ESA 2.4.0 and VM/ESA 1.1.5 370 Feature when transferring a converted print file.

System Message Change for Disconnect [1.2.0]

The message issued to the operator when the system disconnects a terminal because of an I/O error has changed. The message could look like this:

```
11:19:20 GRAF 0AB3 DISCONNECT ETC18566 USERS=59 FORCED BY SYSTEM
```

In your old release, the message might have looked like this:

```
11:19:20 GRAF 0AB3 DISCONN AS ETC18566 USERS=59 FORCED BY SYSTEM
```

Differences Logging Off or Disconnecting [1.1.5, 1.2.0]

For LOGON *userid* HERE or FORCE *userid*, the *userid*'s screen is cleared before the logoff or disconnected messages appear. In the old release, this did not occur.

On local terminals, logical terminals, and, in most cases, VTAM terminals, the messages issued after entering LOGOFF or DISCONNECT are displayed in the system default color. In the old release, if the SCREEN CPOUT *extcolor* command had been used, the color of these messages was *extcolor*.

Chapter 14. Virtual Machine Operation Changes and Conversion Considerations

Virtual machine operation tasks include those tasks that allow you to run operating systems in virtual machines. This section describes tasks that have changed. There are also differences in commands and messages; see the compatibility tables for details.

This chapter may contain information for any or all of the conversions documented in this book. For definitions of the conversion identifiers used in this chapter, see "How to Identify the Topics That Apply to Your Conversion" on page 3.

Break Out of APPC/VM Wait State [1.1.5]

Virtual machine users can use the IPL, LOGOFF, SYSTEM CLEAR, or SYSTEM RESET command to immediately break the system out of an APPC/VM wait state or a wait state caused by a DIAGNOSE code `X'A4'` or `X'A8'` request.

For additional information, see the *VM/ESA: Virtual Machine Operation* book and the *VM/ESA: CP Command and Utility Reference*.

Year 2000 Considerations [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

Range of Dates Supported for Files

Minidisk files and SFS files on VM/ESA 2.4.0 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. VM/ESA 2.4.0 will interpret all dates associated with a file on a back level server as 19*yy*, 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 VM/ESA 2.4.0, all the 4-digit years for existing files will appear to be 19*yy*, even if they have been created after 1999.

If you copy a file from a remote system to a minidisk or SFS directory on VM/ESA 2.4.0 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 VM/ESA 2.4.0 to create a file on a minidisk on a remote system that is not Year 2000 ready, VM/ESA 2.4.0 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 profile from the system disk, followed by your customized changes. For more information, see Appendix A of the *VM/ESA: CMS Command Reference*.

Newer Processors Do Not Support 370-Mode Virtual Machines [1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

Certain new processors, such as the IBM S/390 Parallel Enterprise Server - Generation 4 and later, do not support 370-mode virtual machines. If you install VM/ESA 2.4.0 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.

(You may have already applied this change to your old system through an APAR.)

Chapter 15. Application Programming Changes and Conversion Considerations

This chapter describes CMS application program conversion differences. There are also differences in commands, DIAGNOSE codes, macros, and messages. See the compatibility tables for details.

This chapter may contain information for any or all of the conversions documented in this book. For definitions of the conversion identifiers used in this chapter, see “How to Identify the Topics That Apply to Your Conversion” on page 3.

This chapter discusses the following major topics:

- “CMS Virtual Machine Environments [1.1.5, 1.2.0, 1.2.1, 1.2.2]” on page 290
- “Non-Relocatable Modules Loaded between 16-20MB [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]” on page 290
- “Converting Your REXX and EXEC 2 Application Programs [ALL]” on page 290
- “Converting Your 370-only CMS Application Programs [1.1.5, 1.2.0, 1.2.1, 1.2.2]” on page 291
- “CMS Migration Utility Feature No Longer Available with VM/ESA [2.1.0, 2.2.0]” on page 292
- “REXX/VM Changes [1.1.5, 1.2.0]” on page 293
- “Storage Key Settings Are Different [1.1.5]” on page 293
- “CMS OS Simulation Enhancements and Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]” on page 294
- “Compatibility Considerations for OS VSAM Applications on CMS [1.1.5]” on page 296
- “CPI Communications Changes [1.1.5, 1.2.0]” on page 297
- “CP, GCS, and REXX Macro Library Restructure [1.1.5]” on page 297
- “DBCS Changes [1.1.5]” on page 298
- “Difference in Transferring a Print File [1.1.5]” on page 299
- “Device Class Field in Accounting Record 3 Changed [1.1.5]” on page 299
- “SFS External Security Manager (ESM) Support [1.1.5]” on page 299
- “CSL Routine Changes [1.1.5]” on page 299
- “Changes to SFS Space Processing [1.1.5, 1.2.0]” on page 300
- “Querying the CMS Level [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]” on page 300
- “Pseudo Timer Extended [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]” on page 300
- “Calling CSL Routines That Have Dates as Output [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]” on page 301
- “CMS Pipelines Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]” on page 301

CMS Virtual Machine Environments [1.1.5, 1.2.0, 1.2.1, 1.2.2]

The CMS level for VM/ESA 2.4.0, CMS Level 15, runs in 370-XA (XA virtual machine) or ESA/XC (XC virtual machine) architecture. Only CMS levels prior to CMS Level 12 can run in a 370 virtual machine.

Most CMS applications once restricted to running in a 370 virtual machine will now run in an XA or XC virtual machine if you issue the CP SET 370ACCOM ON command or the CMS SET CMS370AC ON command. In addition, modules generated with the 370 option of the GENMOD command can be run in an XA or XC virtual machine by issuing the CMS SET GEN370 OFF command.

See the *VM/ESA: CP Command and Utility Reference* for information on the SET 370ACCOM command. See *VM/ESA: CP Programming Services* for more information on how to run your 370-only CMS applications in an XA or XC virtual machine. See the *VM/ESA: CMS Command Reference* for information on the SET CMS370AC and SET GEN370 commands.

Non-Relocatable Modules Loaded between 16-20MB [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

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 *VM/ESA: CMS Command Reference* for information on generating modules using the LOAD and GENMOD commands.

Converting Your REXX and EXEC 2 Application Programs [ALL]

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.

Converting Your 370-only CMS Application Programs [1.1.5, 1.2.0, 1.2.1, 1.2.2]

In VM/ESA 2.4.0, CMS cannot run in a 370 virtual machine. Therefore, you need to convert your existing 370-only CMS application programs to run on CMS in XA or XC virtual machines. Although the conversion effort varies from installation to installation, there are four basic stages of conversion to VM/ESA 2.4.0 CMS, as shown in Table 31. Your effort depends on the stage that your application programs are currently in and what stage you plan to convert to.

Table 31. Stages of CMS Application Program Conversion

Conversion Stage	Virtual Machine Mode	Addressing Mode	Conversion Effort
System/370 compatibility	XA or XC with 370ACCOM	24-bit	Least effort. Your application programs run as they did on a 370 mode virtual machine. Your applications cannot work above the 16MB line. See “Running 370-Only CMS Applications in an XA or XC Virtual Machine [1.1.5, 1.2.0, 1.2.1, 1.2.2]” on page 292.
370-XA architecture toleration	XA or XC	24-bit	More effort. At this level your applications work in XA or XC virtual machines without taking advantage of 31-bit addressing or ESA/XC features. Because of basic architectural differences between System/370 on the one hand, and 370-XA and ESA/XC on the other (such as PSWs, assembler instructions, and I/O), 370-XA architecture toleration can require more conversion effort than System/370 compatibility.
370-XA addressing exploitation	XA or XC	31-bit	More effort. Like 370-XA toleration, 370-XA exploitation requires you to adapt your programs to 370-XA architecture. In addition, your programs can support 31-bit addressing; they can run and access data either above or below the 16MB line. Once programs are at this point, they can also execute in an XC virtual machine without any changes.
ESA/XC exploitation	XC	31-bit	Most effort. First, your programs must be able to run in an XC virtual machine. Then, to take full advantage of the data space support offered by ESA/XC architecture, your programs require some additional adaptation. The application must be changed to manage access registers and to understand data spaces concepts.

IBM recommends that you use the following strategy when you convert your CMS application programs to run on VM/ESA 2.4.0:

- If possible, you should convert your CMS application programs to the 370-XA exploitation stage. This lets you move your CMS application programs above the 16MB line and optimize your use of storage below the 16MB line.
- You should use the bimodal CMS preferred programming interface to shield your CMS application programs from the underlying architecture.
- If you cannot convert your CMS application programs to the 370-XA exploitation stage, you should convert them to the 370-XA toleration stage. This will facilitate program calling and the exchange of data addresses within the same CMS virtual machine.

- You should make use of shared segments whenever possible to allow CMS users to share code and data, thus relieving real and virtual storage constraints. You can put XA and XC programs that use 31-bit addressing in saved segments above the 16MB line, freeing space below the line for private code.
- Convert your CMS application programs to exploit VM data spaces as appropriate. Use VM data spaces to enhance the performance of your applications, particularly if they operate in a service virtual machine environment.
- **[1.1.5, 1.2.0]** If possible, you should convert all programs currently using Virtual Machine Communications Facility (VMCF) for communication to APPC/VM (using the high level language interface, CPI Communications or using the assembler interface) or IUCV. VMCF is now maintained in VM/ESA only for compatibility, so any new programs should use APPC/VM for communication.

  Note: VMCF no longer uses the VMDVMCFA bit in VMDBK.

  See Appendix F, “Migrating Programs from VMCF to APPC/VM” on page 653.

You may also have to make changes to your programs because of changes to CMS commands, macros, or messages. Refer to the CMS compatibility tables for your release. See Part 3, “Compatibility Tables” on page 313.

Running 370-Only CMS Applications in an XA or XC Virtual Machine [1.1.5, 1.2.0, 1.2.1, 1.2.2]

You can run most 370-only CMS applications in an XA or XC virtual machine. To do this, enter the CP SET 370ACCOM ON command in your XA or XC virtual machine. (You may also need to enter the CMS SET CMS370AC ON command.) If your program was generated using the 370 option of the GENMOD command, you must also enter the new CMS SET GEN370 OFF command.

The *VM/ESA: CP Programming Services* book contains details about how CP does this.

CMS Migration Utility Feature No Longer Available with VM/ESA [2.1.0, 2.2.0]

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/ESA. 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.

REXX/VM Changes [1.1.5, 1.2.0]

REXX/VM has changed as follows:

- **[1.1.5]** REXX/VM can run in 370, XA, or XC virtual machines and now exploits 31-bit addressing when in XA or XC virtual machines.
- **[1.1.5, 1.2.0]** REXX/VM includes the full REXX Level 2 non-I/O support and stream I/O support that is available on other SAA platforms:
  - Non-I/O support includes support of binary literal strings and binary conversion functions.
  - Stream I/O support includes the following new input and output functions, which replace the external user functions with the same names: CHARIN, CHAROUT, CHARS, LINEIN, LINEOUT, LINES, and STREAM.
- **[1.1.5]** Enhancements to the VALUE function include:
  - Access to CMS global variables
  - Improved variable support in the DROP and PROCEDURE instructions
  - Enhancements in parsing templates.
- **[1.1.5]** Arithmetic operations have been improved so that they are now more accurate.

Storage Key Settings Are Different [1.1.5]

CMS storage management no longer sets storage keys for all virtual storage at IPL. Storage keys are set at IPL only for:

- CMS nucleus inside the virtual machine
- Pages for the page allocation table and the storage management work area
- Pages starting at X'20000'
- Transient area
- DMSNUC

In VM/ESA 1.1.5 370 Feature, storage keys were set for all virtual storage at IPL. CMS storage management no longer sets the storage key to X'E' for fully unallocated pages of virtual storage. When pages are returned to storage management, the storage keys remain in the setting requested upon allocation.

CMS OS Simulation Enhancements and Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

The following topics are discussed in this section:

- “OS Simulation Documentation [1.1.5, 1.2.0, 1.2.1]”
- “DMSTVI Tape Exit (TVISECT)[1.1.5]”
- “MVS CVT (CMSCVT) Expansion [1.1.5]”
- “STAE/ESTAE Macros [1.1.5, 1.2.0, 1.2.1, 1.2.2]”
- “OS Simulation Support for Tape Library Dataservers [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]”

OS Simulation Documentation [1.1.5, 1.2.0, 1.2.1]

Documentation about OS simulation has been clarified, corrected, and enhanced. For example, OS simulation documentation now clearly indicates which MVS macros are not supported and which parameters and options are supported on supported MVS macros. See the *VM/ESA: CMS Application Development Guide for Assembler* and the *VM/ESA: CMS Application Development Reference for Assembler*. Also, information about the FILEDEF and GLOBAL commands has been improved in the *VM/ESA: CMS Command Reference*.

DMSTVI Tape Exit (TVISECT)[1.1.5]

The DMSTVI tape exit has upwardly compatible changes. The TVISECT macro supports four new fields:

TVIFLAGS	If on, it indicates that the new fields are available.
TVIRECFM	Indicates RECFM values.
TVILRECL	Indicates LRECL values.
TVIBLKSI	Indicates BLKSIZE values.

MVS CVT (CMSCVT) Expansion [1.1.5]

The simulated MVS CVT table, CMSCVT, is expanded to its full MVS size to accommodate the CVTOSLVL fields that are at the end of CVT. The CVTDCB field is set with an indicator that shows whether the CVTOSLVL fields are available to the application. Field CVTTZ in CVTSECT is updated to reflect current timezone offset when X'2004' interrupt is processed by CMS.

STAE/ESTAE Macros [1.1.5, 1.2.0, 1.2.1, 1.2.2]

As a part of retry support, the value of register 0 upon entry to a STAE/ESTAE routine was always 4. In VM/ESA 2.4.0, the value of register 0 upon entry to a STAE/ESTAE routine is always 8. This may cause some applications, which determine whether or not to attempt retry based on the contents of Register 0, to execute a different code path and produce different results.

OS Simulation Support for Tape Library Dataservers [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

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.

For more information about using Tape Library Dataservers under OS simulation, see the *VM/ESA: CMS Application Development Guide for Assembler*.

Additional OS Simulation Enhancements [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]

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:

- *VM/ESA: CMS Application Development Guide for Assembler*

- *VM/ESA: CMS Command Reference*

Compatibility Considerations for OS VSAM Applications on CMS [1.1.5]

In CMS, VSAM support is based on VSE/VSAM. For OS VSAM applications, all VSAM requests are translated by CMS into VSE/VSAM requests and passed to VSE/VSAM. In VM/ESA 1.1.5 370 Feature, the CMS mechanism for performing this translation included converting the application's own control blocks (ACB, RPL, EXLST) from OS VSAM control blocks to VSE/VSAM control blocks.

In VM/ESA 2.4.0, CMS performs the control block conversion by making a separate VSE copy of the ACB, RPL, and EXLST control blocks. The application's OS VSAM copies are left intact.

OS VSAM applications that retrieve or alter data within their own ACB, RPL, or EXLST blocks directly may break because of this change. These are now OS control blocks rather than VSE/VSAM control blocks. OS control blocks and VSE/VSAM control blocks are generally incompatible because fields representing the same data are at different offsets.

Applications that use only the Control Block Manipulation Macros (CBMMs), such as GENCB, MODCB, SHOWCB, and TESTCB, are immune to this change. CBMMs are provided by OS VSAM to manipulate OS VSAM ACB, RPL, or EXLST control blocks.

GETIME (CMS DOS), used by AMSERV, will reflect the current time zone.

An Example of an Application that Broke

One application that broke was using an MVC instruction to update the DDNAME within an ACB. The offset of the VSE ACB's DDNAME field corresponds to the password field in an OS ACB. So, the password field in the application's OS ACB was overlaid with the DDNAME. When the application tried to OPEN the ACB, CMS issued an error message stating that the password field was above the 16MB line.

To fix this problem, the application changed to use the MODCB macro to update the DDNAME field of the ACB. This allowed the application to run on the new release as well as past releases. Note that had the application merely been changed to do an MVC of the DDNAME into the correct offset for an OS ACB, the application would have failed on past releases. Therefore, it is recommended that applications use the CBMMs whenever possible to manipulate VSAM ACBs, RPLs, and EXLSTs.

Steps to Avoid This Problem in Your Applications

Before moving to VM/ESA 2.4.0, check applications for direct references to VSAM ACBs, RPLs, and EXLSTs:

1. Search for any IKQACB, IKQRPL, or IKQEXLST mapping macros, which are mapping macros for VSE/VSAM control blocks.
2. Search for any hard-coded offsets for specific fields:
   a. Find any references to fields in ACBs, RPLs, and EXLSTs
   b. See if any of these are direct references.

3. Change these references to use CBMMs.

CPI Communications Changes [1.1.5, 1.2.0]

The following section summarizes the changes in CPI Communications. For information on changes to CPI Communications routines since VM/ESA 1.1.5 370 Feature, see "Common Programming Interface (CPI) Communications Routines" on page 425.

TXTLIBs Required for Module Creation [1.1.5, 1.2.0]

For all languages except REXX, when the CPI Communications application is bound into a module, the disks or directories containing VMLIB TXTLIB and CMSSAA TXTLIB must be accessed. Both VMLIB and CMSSAA must be specified on a GLOBAL TXTLIB command.

Event Management [1.1.5]

IBM recommends you use CMS Event Management Services and the VMCPIC system event rather than the Wait_on_Event (XCWOE) routine to monitor and respond to CPI Communications events. Wait_on_Event continues to be supported, but it blocks the entire virtual machine. The CMS EventWait call, however, blocks only the thread on which it is issued. This allows a multitasking application to continue performing work on other threads.

The VMCPIC event supports the following communications events:

- Allocation requests
- Information input from partner
- Resource revoked notification

User events can be defined and monitored using CMS Event Management Services, and the VMCON1ECB system event is provided for handling console input.

The section about understanding CPI Communications in the *VM/ESA: CMS Application Development Guide* has example scenarios showing the use of the VMCPIC event. The *VM/ESA: CMS Application Multitasking* book describes CMS Event Management Services and the EventWait function.

CP, GCS, and REXX Macro Library Restructure [1.1.5]

Table 32 on page 298 shows how the macro library structures for CP, GCS, and REXX have been changed.

Table 32. MACLIB Restructure by Component

Component	Old Name	New Name	MACLIB Description
CP	DMKGPI	HCPGPI	General-use programming interface macros
	DMKPSI	HCPPSI	Other programming interface macros
	DMKOM	HCPOM1	For IBM use only
		HCPOM2	For IBM use only
	CPLIB	HCPGPI	General-use programming interface macros
		HCPPSI	Other programming interface macros
		HCPOM1	For IBM use only
		HCPOM2	For IBM use only
	DMKOMA	—	Removed (was AP)
	DMKOMAT	—	Removed (was AP Fret Trap)
	DMKOMM	—	Removed (was MP)
	DMKOMMT	—	Removed (was MP Fret Trap)
GCS	CSIGPI	GCTGPI	General-use programming interface macros
	CSIOM	GCTOM	For IBM use only
REXX	IXXOM	IXXOM	For IBM use only
IPCS	DMMOM	—	Removed (was for IBM use only)

Note: HCPGPI and HCPPSI are now located on the system disk. The maclibs that existed in the 370 Feature, DMKGPI and DMKPSI, were located on the MAINT-owned 194 minidisk.

DBCS Changes [1.1.5]

In VM/ESA 2.4.0, your application can issue a line-mode write operation with a virtual start (SIO, SIOF, and SSCH) to write double-byte character set (DBCS) data to the console. In VM/ESA 1.1.5 370 Feature, this could be done only with a DIAGNOSE code X'58'.

Also, in VM/ESA 2.4.0, single-byte character set (SBCS) and DBCS data was formatted for display based on the physical size of the screen. VM/ESA 2.4.0 displays SBCS and DBCS data based on the smaller of:

- The physical size of the screen
- The logical line size as set with the TERMINAL LINESIZE command.

If you have issued TERMINAL LINESIZE OFF, or have never issued the TERMINAL LINESIZE command, then your VM/ESA 2.4.0 system is compatible with your VM/ESA 1.1.5 370 Feature system in this aspect.

VM/ESA now has input support for DBCS. A terminal user can input mixed DBCS data on the input line of the terminal screen. Some CP commands have been enhanced to accepted mixed DBCS data. Existing CMS commands that accepted mixed DBCS data in fullscreen CMS now can be entered in line mode.

Difference in Transferring a Print File [1.1.5]

In VM/ESA 1.1.5 370 Feature, when you transfer a converted print file to a reader or punch queue, the file is left in its converted state. Before printing the file, you must enter the CHANGE command with the UNCONV operand to cancel any preprocessing that the print server may have done on the file.

In VM/ESA 2.4.0, the transferred file is “unconverted” automatically, and CHANGE UNCONV is not required before printing; this is a usability benefit.

Device Class Field in Accounting Record 3 Changed [1.1.5]

Applications that depend on the device class field in accounting record 3 may have to change. In VM/ESA 2.4.0, the device class field is always X'04', which means 'DASD'. In VM/ESA 1.1.5 370 Feature, the device class field contained X'40' for non-FBA DASD and X'01' for FBA DASD.

In VM/ESA 2.4.0, to determine if the DASD is FBA or not, use the device type field instead:

- X'02' for 3370
- X'21' for 9332
- X'22' for 9335
- X'24' for 9336

Another scenario that you might have used in your old application was the following:

1. Look at the device class or the device type field
2. If it indicates CKD, get the number of cylinders from columns 37-38
3. If it indicates FBA, get the number of blocks from columns 37-40.

To avoid this dependency on the architecture of your DASD device in your old applications, or in any new applications that you write, you can now use the number of 4KB pages instead of the number of cylinders or blocks. A new field in columns 41-44 of the accounting record contains the number of 4KB pages used.

SFS External Security Manager (ESM) Support [1.1.5]

SFS support in VM/ESA 2.4.0 provides an interface to an ESM for external objects. ESMs that support SFS should be aware that:

- They must accept, reject, or defer external objects
- The LOGSTR string used on RACROUTE macros for SFS contains additional information for external objects.

CSL Routine Changes [1.1.5]

In many CSL routines, you can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.

Changes to SFS Space Processing [1.1.5, 1.2.0]

When using any of the following to write to an SFS directory in a VM/ESA 2.4.0 file pool, a disk full error is reported when the file space limit is exceeded:

- FS macros
- EXECIO command
- OS/MVS PUT or WRITE macros

In the old release:

- For FSWRITE and EXECIO, no disk full condition was reported, and you could continue writing.
- For OS/MVS PUT or WRITE, the disk full condition was returned when CMS reported that the file space threshold was exceeded.

If you want to continue writing beyond your file space limit, you have to use CSL routines (such as, DMSWRITE, DMSWRBLK, or DMSWRDBK).

Also, applications that want to return a file to its original state after FSWRITE has failed because the SFS file space limit was exceeded must use the DMSROLLB CSL routine. In the old release, the FINIS command would result in a rollback of all changes because the file space limit was exceeded. In VM/ESA 2.4.0, however, writes are not allowed when the file space limit is exceeded. Any earlier successful writes are committed.

If your application must support VM/ESA 1.2.0 or earlier file pools, you can use the FSCBTHEX (X'80'), file space threshold exceeded, indicator bit in the FSCBFLG byte of the FSCB block to better anticipate if you have enough blocks available to continue writing to a file space.

Querying the CMS Level [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

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 Version 2 Release 1.0). The CMSLEVEL assembler macro will not map CMS levels beyond CMS level 12. To determine the CMS level on VM/ESA Version 2 Release 1.0 or later, use the DMSQEFL CSL routine or the DMSQEFL assembler macro.

Pseudo Timer Extended [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

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 [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

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:

Keyword	**Format**
SHORTDATE	*yy/mm/dd*
FULLDATE	*yyyy/mm/dd*
ISODATE	*yyyy-mm-dd*

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.

CMS Pipelines Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

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.
- 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.

  Note: User-written applications that are sensitive to these changes may require alterations. Published externals (such as PIPGFTXT) have **not** been changed.
- 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 *VM/ESA: CMS Pipelines Reference* can be found in the *CMS/TSO Pipelines: Author's Edition*, which is now included with the VM/ESA 2.4.0 library.

For more information, see:

- *VM/ESA: CMS Pipelines User's Guide*
- *VM/ESA: CMS Pipelines Reference*
- *CMS/TSO Pipelines: Author's Edition*

For cross-references between the old (DMS) and new (FPL) Pipelines message numbers, see Appendix H, "CMS Pipelines Message Cross-Reference [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]" on page 669.

Chapter 16. Diagnosis Changes and Conversion Considerations

Diagnosis tasks include isolating, describing, reporting, and correcting failures in VM/ESA 2.4.0 and guest operating systems.

Some diagnosis tasks perform differently in VM/ESA 2.4.0 than in your old release. This chapter discusses them and the tools used to perform them. There are also differences in commands and messages. See the compatibility tables for more details.

This chapter may contain information for any or all of the conversions documented in this book. For definitions of the conversion identifiers used in this chapter, see “How to Identify the Topics That Apply to Your Conversion” on page 3.

This chapter discusses the following major topics:

- “Error Recording Differences [1.1.5]”
- “Dump Viewing Facility Replaces IPCS [1.1.5]”
- “Dump Processing in VM/ESA 2.4.0 [1.1.5, 1.2.0, 1.2.1, 1.2.2]” on page 307
- “Minidisk Corruption Detection [1.1.5]” on page 309
- “VM/ESA 2.4.0 CP Trace Table Differences [1.1.5]” on page 309
- “No CPTRAP Command [1.1.5]” on page 309
- “Monitor Record Changes [ALL]” on page 309
- “FST and ADT Macro Changes to Support 4-digit Years [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]” on page 311

Error Recording Differences [1.1.5]

In VM/ESA 1.1.5 370 Feature, CP creates EREP records when I/O and machine errors occur. In VM/ESA 2.4.0, the same happens, but the records are called LOGREC records.

In VM/ESA 1.1.5 370 Feature, EREP records are kept on error recording cylinders specified during system generation, and you use the CMS CPEREP command to format and print the records. In VM/ESA 2.4.0, LOGREC records are kept in a CMS file, and you use the CPEREPXA utility to format and print them.

In VM/ESA 2.4.0, you can use the RECORDING command to change processing parameters that control the recording of LOGREC records.

An operating system running in a virtual machine can either use its own error recording facilities or have CP handle error recording. Use the VM/ESA 2.4.0 SET SVC76 command to specify which one is to be used.

See *VM/ESA: System Operation* for more information about error recording.

Dump Viewing Facility Replaces IPCS [1.1.5]

VM/ESA 2.4.0 uses the Dump Viewing Facility rather than IPCS for viewing dumps. The Dump Viewing Facility is a component of VM/ESA 2.4.0 that runs under CMS. Using the Dump Viewing Facility, you can:

- Interactively display dump data from CP abend dumps, CP stand-alone dumps, and virtual machine dumps (including AVS, CMS, GCS, RSCS, SFS, TSAF, and PVM virtual machine dumps).
- Format and print summary information from CP abend dumps, CP stand-alone dumps, and virtual machine dumps.
- Process CP trace table data stored on tape, or process CP trace table data, data trace data, I/O trace data, and guest trace data stored in system trace files. Specifically, you can:
  - Load CP trace table data from tape or DASD
  - Load data trace, I/O trace, and guest trace data from DASD
  - View and format CP trace table data
  - View and format data trace, I/O trace, and guest trace data
  - Merge CP trace table, data trace, I/O trace, and guest trace data
  - Print CP trace table, data trace, I/O trace, or guest trace data.

The Dump Viewing Facility does not support the problem reporting and management function found in IPCS.

Comparison of IPCS and Dump Viewing Facility Commands

The following table compares IPCS and Dump Viewing Facility commands. See *VM/ESA: Dump Viewing Facility* for a complete description of the Dump Viewing Facility.

Table 33 (Page 1 of 2). Comparison of IPCS and Dump Viewing Facility Commands

IPCS Command	Dump Viewing Facility Command	Notes
APAR	———	The Dump Viewing Facility does not support problem reporting and management.
IPCS	DUMPSCAN	The syntax is different, the Dump Viewing Facility uses the XEDIT interface for display management, and some of the subcommands have functional differences. See Table 34 on page 305 for subcommand differences.
IPCSDUMP	———	The Dump Viewing Facility does not support problem reporting and management. However, the VM/ESA 2.4.0 DUMPLOAD utility has some of the functions of IPCSDUMP.
MAP	Same	The syntax is different.
PRB	———	The Dump Viewing Facility does not support problem reporting and management.
PROB	———	The Dump Viewing Facility does not support problem reporting and management.
PRTDUMP	Same	The syntax is different.
STAT	———	The Dump Viewing Facility does not support problem reporting and management.

Table 33 (Page 2 of 2). Comparison of IPCS and Dump Viewing Facility Commands

IPCS Command	Dump Viewing Facility Command	Notes
———	ADDMAP	Use this command to append the compressed module map to the CMS file containing the dump processed by the VM/ESA 2.4.0 DUMPLOAD utility. In VM/ESA 1.1.5 370 Feature, the IPCSDUMP command does this.
———	VIEWSYM	This command selects and views symptom records in CMS files created by the RETRIEVE SYMPTOM utility of VM/ESA 2.4.0.
———	TRACERED	This Dump Viewing Facility command is an equivalent of the VM/ESA 1.1.5 370 Feature TRAPRED facility. It also contains functions from the DSJVMCMS program.

Comparison of DUMPSCAN to IPCSSCAN

The following table compares IPCS IPCSSCAN and Dump Viewing Facility DUMPSCAN.

Table 34 (Page 1 of 3). Comparison of IPCS IPCSSCAN and Dump Viewing Facility DUMPSCAN Subcommands

IPCS IPCSSCAN Subcommand	Dump Viewing Facility DUMPSCAN Subcommand	Notes
Reuse	(Null line)	
?	Same	The Dump Viewing Facility ? subcommand is not limited to displaying the last DUMPSCAN subcommand run, but displays the last command or subcommand run of any type, DUMPSCAN or not.
+ and −	Same	
&name	Same	
AREGS	AREGS, REGS	The IPCS AREGS subcommand displays general, control, and floating-point registers; clocks; PSWs; the CSW; the CAW; and timers. The Dump Viewing Facility AREGS subcommand displays only access registers; use REGS for the other functions.
ARIOBLOK	———	Not supported by the Dump Viewing Facility.
C	CREGS	The Dump Viewing Facility supports CP, stand-alone, and virtual machine dumps, and lets you specify a processor address.
CHAIN	Same	The syntax is different and the Dump Viewing Facility has more functions.
CMS	Same	The Dump Viewing Facility can run CMS commands from the DUMPSCAN command line.
CMSPOINT	Same	
CORTABLE	FRAMETBL	The Dump Viewing Facility has enhanced capabilities for invocation and displaying by frame table address.
DISPLAY	Same	The Dump Viewing Facility has enhanced offset and indirect addressing displays, but does not display previous lines.
DOSPOINT	Same	
DUMPID	Same	
END	Same	
FDISPLAY	Same	

Table 34 (Page 2 of 3). Comparison of IPCS IPCSSCAN and Dump Viewing Facility DUMPSCAN Subcommands

IPCS IPCSSCAN Subcommand	Dump Viewing Facility DUMPSCAN Subcommand	Notes
FORMAT	FORMAT	In IPCS, FORMAT displays trace entries in their long format, usually multiple lines per trace entry. In the Dump Viewing Facility, FORMAT changes or queries the type of virtual machine dump being viewed.
G	GREGS	The Dump Viewing Facility lets you specify a processor address.
GDISPLAY	Same	
HELP	Same	The Dump Viewing Facility does not give help for messages.
HX	Same	
IPCSMAP	———	Not supported by the Dump Viewing Facility.
IUCV	Same	
LOCATE (UP)	Same	
MAPA	FINDMOD	FINDMOD combines the functions of MAPA and MAPN.
MAPN	FINDMOD	FINDMOD combines the functions of MAPA and MAPN.
MREGS	REGS	
MRIOBLOK	———	Not supported by the Dump Viewing Facility.
OSPOINT	Same	
PRINT	Same	
QUIT	Same	
REGS	Same	The IPCS REGS subcommand displays only general registers for virtual machine dumps, in addition to other information. The Dump Viewing Facility REGS subcommand displays all types of registers for virtual machine dumps, in addition to other information. The IPCS REGS subcommand does not let you specify a processor address; the Dump Viewing Facility REGS subcommand does.
RIOBLOK	Same	The Dump Viewing Facility RIOBLOK subcommand may be invoked with logical device or real device addresses in addition to device numbers.
SCROLL	FORWARD	FORWARD does not have operands and can be used with FINDMOD, LOCATE, DISPLAY, and TRACE.
SCROLL U	BACKWARD	BACKWARD does not have operands and can be used with FINDMOD, LOCATE, DISPLAY, and TRACE.
SYMP	SYMPTOM	The Dump Viewing Facility does not support symptom string extraction for virtual machine dumps.
TACTIVE	Same	
TLOADL	Same	
TRACE	Same	The Dump Viewing Facility TRACE subcommand merges trace tables from multiple processors and does not support the SCROLL and SCROLLU operands for CP dumps.
TSAB	Same	
USERMAP	———	Not supported by the Dump Viewing Facility.
VIOBLOK	Same	The Dump Viewing Facility VIOBLOK subcommand may be invoked with virtual subchannel or virtual device addresses in addition to virtual device numbers.

Table 34 (Page 3 of 3). Comparison of IPCS IPCSSCAN and Dump Viewing Facility DUMPSCAN Subcommands

IPCS IPCSSCAN Subcommand	Dump Viewing Facility DUMPSCAN Subcommand	Notes
VMBLOK	VMDBK	The Dump Viewing Facility does not have the SYSSPOOL operand, and VMDBK may be invoked with storage addresses or individual VMDBK fields.
VMLOADL	Same	
———	BLOCK	This subcommand formats control blocks within a dump.
———	CPU	This subcommand displays the processor address and the prefix register value for each processor in the dump.
———	FINDUSER	This subcommand determines the CP module flow for a hung virtual machine.
———	SCAN	This subcommand processes a PF key assignment or a command string from REXX.
———	SELECT	This subcommand selects trace table entries for display.
———	SNAPLIST	This subcommand displays a summary snaplist in a soft abend dump.
———	SYMPTOM	This subcommand displays formatted symptom record information.
———	VREG	This subcommand displays the contents of a specific vector register or a designated number of vector elements if that information is available in a virtual machine dump.
———	VPAIR	This subcommand displays the contents of a specific even/odd pair of vector registers or a designated number of vector elements.
———	VSTAT	This subcommand displays status information about the Vector Facility for any processor in a virtual machine dump.
———	XEDIT	This subcommand causes the Dump Viewing Facility to pass the command line to XEDIT for execution.

Dump Processing in VM/ESA 2.4.0 [1.1.5, 1.2.0, 1.2.1, 1.2.2]

This section discusses the following topics:

- “Soft and Hard CP Abend Dumps [1.1.5]”
- “Requirements for Hard CP Abend Dump Space [1.1.5]”
- “Virtual Machine Dump Difference [1.1.5]” on page 308
- “Dump Format Change [1.1.5, 1.2.0, 1.2.1]” on page 308
- “Changes to Dumps Caused by Severe File System Errors [1.1.5, 1.2.0]” on page 308
- “Stand-Alone Dump Utility Differences [1.1.5]” on page 309

Soft and Hard CP Abend Dumps [1.1.5]

VM/ESA 2.4.0 records soft and hard CP abend dumps:

- A soft CP abend dump is always written to a soft abend dump file on the reader queue. You define the recipient user ID of the soft abend dump file during system generation.
- A hard CP abend dump is written to printer, tape, or DASD. You define the dump device using the SET DUMP command.

In VM/ESA 2.4.0, dumps include a control block identifier.

In VM/ESA 2.4.0, you can use the DUMPLOAD utility to load a dump into a CMS dump file for use by the Dump Viewing Facility, and to print a dump from a CMS dump file in the same format as a system abend dump.

Requirements for Hard CP Abend Dump Space [1.1.5]

The advantages of writing hard CP abend dumps to DASD rather than to a printer are that the dumps are transmitted faster and they are machine-readable. However, there must be enough DASD space to hold a complete storage dump; if not, it is sent to a printer. In VM/ESA 2.4.0, hard CP abend dumps are more likely to be written to DASD because they are written to discontiguous cylinders or extents in an area of spool space reserved for dumps. In VM/ESA 1.1.5 370 Feature, the dumps require contiguous cylinders in their own space (dump space). Even though there might be enough overall dump space, if there are not enough contiguous cylinders to hold a full storage dump, the dump is sent to a printer.

In VM/ESA 1.1.5 370 Feature, dump space is allocated during system initialization by the SYSPAG macro of DMKSYS. In VM/ESA 2.4.0, the SYSPAG macro does not exist. Instead, CP automatically reserves enough space in spool space to hold a hard CP abend dump. (You can use the CP_OWNED statement in the SYSTEM CONFIG file to define space on a particular volume.) Therefore, when you define overall spool space, be sure to include space that you specified as dump space in VM/ESA 1.1.5 370 Feature, plus additional space to allow for the larger storage sizes in VM/ESA 2.4.0.

Virtual Machine Dump Difference [1.1.5]

Virtual machine dumps created on VM/ESA 1.1.5 370 Feature cannot be processed by the DUMPLOAD utility on VM/ESA 2.4.0.

Dump Format Change [1.1.5, 1.2.0, 1.2.1]

CP dumps and stand-alone dumps now use dump file formats that match those created by a virtual machine dump.

Changes to Dumps Caused by Severe File System Errors [1.1.5, 1.2.0]

For some severe irrecoverable file system errors detected by CMS, an entire virtual machine dump is generated unless SET AUTODUMP OFF was specified. In your old release, your AUTODUMP setting was always respected.

Stand-Alone Dump Utility Differences [1.1.5]

Because VM/ESA 2.4.0 stores dumps in system data files, you cannot transport dumps from your VM/ESA 2.4.0 system to your VM/ESA 1.1.5 370 Feature system. If you have a problem and want to back out to VM/ESA 1.1.5 370 Feature, you will need the stand-alone dump utility to examine the problem.

See “CP Utilities” on page 357 for other differences between the VM/ESA 1.1.5 370 Feature stand-alone dump utility and the VM/ESA 2.4.0 stand-alone dump utility.

Minidisk Corruption Detection [1.1.5]

In VM/ESA 2.4.0, the CMS file system has been enhanced to detect some forms of minidisk corruption. Two new error messages, DMS1305T and DMS1306T, help diagnose these file system errors.

It is possible that minidisks that you move to the new system may have already-existing corruption that CMS now detects upon the first output operation to an accessed R/W disk. In this case, you receive message DMS1305T, which indicates the problem, and you are placed in a disabled wait state.

VM/ESA 2.4.0 CP Trace Table Differences [1.1.5]

The number and type of trace table entries differs in VM/ESA 2.4.0.

For soft abend dumps, the dump contains the trace tables for the failing processor. For hard abend dumps, the dump contains the CP trace tables of all processors.

VM/ESA 2.4.0 CP trace events are time-stamped to allow comparison of events happening currently on more than one processor. TRSAVE sends data to a system data file or to tape rather than to a CPTRAP spool file as in VM/ESA 1.1.5 370 Feature.

No CPTRAP Command [1.1.5]

The CPTRAP command is not available in VM/ESA 2.4.0. Instead, use the SET CPTRACE, TRSOURCE, and TRSAVE commands. SET CPTRACE traces CP events, TRSOURCE traces virtual machine events, and TRSAVE records CP and virtual machine events. See *VM/ESA: CP Command and Utility Reference* for further information.

Monitor Record Changes [ALL]

The layouts of the monitor records are contained in a file named MONITOR LIST1403, which was loaded onto your base CP object disk (194) at the time VM/ESA was installed on your system. This file, which is already formatted for printing, can be printed directly or it can be read online. You can find instructions on how to do this in the *VM/ESA: Performance* book.

New SFS and CRR Server Monitor Record Counter ID Information [1.1.5, 1.2.0, 1.2.1, 1.2.2]

The following table shows new counter IDs for SFS and CRR servers' monitor records.

- **[1.2.0]** Counters 149-205 are new.
- **[1.1.5, 1.2.1, 1.2.2]** Counters 150-205 are new.

Counter ID	Counter Name	New	Description
149	OPENCMF	New	Open File CreateMig Requests
150	BFACCESS	New	Byte File Check File Accessibility Requests
151	BFCHMOD	New	Byte File Change Mode Requests
152	BFCHOWN	New	Byte File Change Owner Requests
153	BFCLOSE	New	Byte File Close File Requests
154	BFCLOSEDIR	New	Byte File Close Directory Requests
155	BFZAPCAT	New	Byte File ZAPCAT Requests
156	BFLINK	New	Byte File Create Link Requests
157	BFLOCKBY	New	Byte File Lock Byte Requests
158	BFLOOKUP	New	Byte File Lookup Requests
159	BFMKCAT	New	Byte File Makecat Requests
160	BFMKREGFILE	New	Byte File Create Regular file Requests
161	BFMKDIR	New	Byte File Create Directory Requests
162	BFMKSYMLINK	New	Byte File Create Symbolic Link Requests
163	BFMKEXTLINK	New	Byte File Create External Link Requests
164	BFMKFIFO	New	Byte File Create Named Pipe (FIFO) Requests
165	BFMKCHARSPEC	New	Byte File Create Character Special File Requests
166	BFMKBLKSPEC	New	Byte File Create Block Special File Requests
167	BFOPENNEWR	New	Byte File Open File New With Intent Read Requests
168	BFOPENNEWW	New	Byte File Open File New With Intent Write Requests
169	BFOPENREAD	New	Byte File Open File Read Requests
170	BFOPENWRIT	New	Byte File Open File Write Requests
171	BFOPENDIR	New	Byte File Open Directory Requests
171	BFREAD	New	Byte File Read File Requests
173	BFREADDIR	New	Byte File Read Directory Entry Requests
174	BFREADLINK	New	Byte File Read Link Contents Requests
175	BFRENAME	New	Byte File Rename Requests
176	BFRMDIR	New	Byte File Remove Directory Requests
177	BFSREGFILECLEANUP	New	Byte File Unlinked File Cleanup Requests
178	BFTOKRETRN	New	Byte File Token Return Requests
179	BFTSLOCKBY	New	Byte File Test Locked Bytes Requests
180	BFUNLINK	New	Byte File Unlink Requests
181	BFUNLOCKBY	New	Byte File Unlock Byte Requests

Counter ID	Counter Name	New	Description
182	BFUTIME	New	Byte File Change Access/Modification Time Requests
183	BFWRITE	New	Byte File Write File Requests
184	TOKCONFLCAUSCBS	New	Byte File Token Conflicts Causing Callbacks
185	GLOBCBWTIME	New	Byte File Callback Wait Time
186	TOKCBTORETRIES	New	Byte File Token Callback Timeout Retries
187	TOKCBREQRETRIES	New	Byte File Token Callback Requestor Retries
188	DOIDCLLOCK	New	Byte File Directory Creation/Deletion Logical Lock Conflicts
189	OIDBLLOCK	New	Byte File Token Manager Logical Lock Conflicts
190	NAMTIDLLOCK	New	Byte File NAMECAT Unallocated
191	OIDGSLLOCK	New	Byte File Global Storage Logical Lock Conflicts
192	OIDLLLOCK	New	Byte File File Logical Lock Conflicts
193	BFSLOCKRETRIES	New	Byte File Logical Lock Retries
194	BFSLOCKEXCCEDED	New	Byte File Logical Lock Retries Exceeded
195	BFSBRLockWaits	New	Byte File Byte Range Lock Waits
196	BFPIPEOPENREAD	New	Byte File Pipe Open For Read Requests
197	BFPIPEOPENWRITE	New	Byte File Pipe Open For Write Requests
198	BFPIPEREAD	New	Byte File Pipe Read Requests
199	BFPIPEWRITE	New	Byte File Pipe Write Requests
200	BFPIPECLOSE	New	Byte File Pipe Close Requests
201	BFPIPEACCESS	New	Byte File Pipe Access Requests
202	BFPIPEUTIME	New	Byte File Pipe Utime Requests
203	BFPIPESTAT	New	Byte File Pipe Stat Requests
204	BFCANCEL	New	Byte File Cancel Requests
205	BFCHAUDIT	New	Byte File Change Audit Requests

FST and ADT Macro Changes to Support 4-digit Years [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0]

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 *VM/ESA: CMS Application Development Reference for Assembler*.

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 *VM/ESA: CMS Application Development Reference*. For information about FSSTATE, see the *VM/ESA: CMS Application Development Reference for Assembler*.

Table 35. 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 (*mm yy*) last written
1F	31	1	B	E	FSTFLAGS	FST flag byte
		Bit 4	B	E	FSTCNTRY	X'08' Century last written (0 - 19*nn*, 1 - 20*nn*)
26	38	2	B		FSTYEARW	Year (*yy*) last written
36	54	6		E	FSTADATI	Alternate date/time (*yy mm dd hh mm ss*)
4E	78	1		E	FSTFB3	FST flag byte 3
		Bit 4		E	FSTCDOLR	X'08' Century date of last reference (0 - 19*nn*, 1 - 20*nn*)
54	84	3		E	FSTDOLR	Date of last reference

Note: 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.

Part 3. Compatibility Tables

This part of the book contains a chapter for each type of conversion:

- Chapter 17, “Compatibility Tables for Converting from VM/ESA 1.1.5 370 Feature” on page 315
- Chapter 18, “Compatibility Tables for Converting from VM/ESA 1.2.0” on page 455
- Chapter 19, “Compatibility Tables for Converting from VM/ESA 1.2.1” on page 503
- Chapter 20, “Compatibility Tables for Converting from VM/ESA 1.2.2” on page 545
- Chapter 21, “Compatibility Tables for Converting from VM/ESA 2.1.0” on page 575
- Chapter 22, “Compatibility Tables for Converting from VM/ESA 2.2.0” on page 599
- Chapter 23, “Compatibility Tables for Converting from VM/ESA 2.3.0” on page 611

Each chapter contains a series of tables listing the VM/ESA externals (commands, messages, and so on) that have changed between that VM/ESA release and VM/ESA 2.4.0.

Chapter 17. Compatibility Tables for Converting from VM/ESA 1.1.5 370 Feature

This chapter identifies the VM/ESA externals that have changed between VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0. It contains the following major sections:

- "CP Changes"
- "CMS Changes" on page 384
- "REXX/VM Changes" on page 448
- "GCS Changes" on page 450
- "AVS Changes" on page 452
- "VMSES/E Changes" on page 452
- "Programmable Operator Facility Changes" on page 453

Use the information provided in this chapter to determine if you need to make any changes in the way you use these functions. See "Terminology and Symbols" on page 3 for the meaning of compatibility terms used in the tables.

CP Changes

This section identifies the CP externals that have changed since VM/ESA 1.1.5 370 Feature. It contains the following subsections:

- "CP Commands"
- "Converting DMKRIO and DMKSYS Macros to SYSTEM CONFIG File Statements" on page 348
- "User Directory Control Statements" on page 352
- "CP Utilities" on page 357
- "CP DIAGNOSE Codes" on page 360
- "CP Macros" on page 380
- "CP Messages" on page 383

CP Commands

Table 36 lists the CP commands that have changed since VM/ESA 1.1.5 370 Feature. Refer to the *VM/ESA: CP Command and Utility Reference* for complete descriptions of CP commands.

Note: The prefix for CP messages is now HCP rather than DMK. Also, the three character module identifiers have been removed from the messages listed in the *VM/ESA: CP Command and Utility Reference*. For example, a message that used to be listed as DMKUSO003E is now listed as HCP003E.

Table 36 (Page 1 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
ACNT	**Incompatible changes:** • Accepts but ignores the CLOSE operand if specified. • Accounting records are kept in storage and sent to your A-disk when requested. In VM/ESA 1.1.5 370 Feature, accounting records were kept in spool files. • Does not send a response that accounting has been done. **Upwardly compatible changes:** • New operand: SYSTEM.

Table 36 (Page 2 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
ADSTOP	**VM/ESA 1.1.5 370 Feature only.** TRACE performs a similar function.
ATTACH *raddr*	**Incompatible changes:** • Operands no longer supported: CHANNEL, PROC, 3330V. • Does not accept A instead of AS. • Clears temporary disk space during attachment to the system. **Upwardly compatible changes:** • Adds new operands: – [WITH] NOASSIGN – [WITH] SYSCTL – [WITH] DEVCTL – [WITH] NOCTL – XSTORE (or XSTORAGE) [TO] {*userid*|*} [*nnnnnnnnM*|ALL] – NOIOASSIST. • Limits the range of devices attached to 256; in VM/ESA 1.1.5 370 Feature, the limit was 48. • Supports access mode for a tape device. • Supports the new dynamic switching devices and dual copy function of the 3990-3. • The R/O operand implies NOIOASSIST. • The response may contain new values or fields for new support.
AUTOLOG	**Incompatible changes:** • For 370 virtual machines and XA or XC virtual machines, CP presents *console-input-data* to CMS at different times. (System/370 worked like 370 virtual machines work in the new release.) The result is that, in 370 virtual machines, *console-input-data* is presented to CMS when the first console read occurs. The console read is normally done by SYSPROF EXEC when it issues a PARSE EXTERNAL instruction, unless SYSPROF EXEC is bypassed or the AUTOCR parameter was used at IPL time. In XA or XC virtual machines, *console-input-data* is presented to CMS before the console read occurs. CMS places the *console-input-data* in the terminal input buffer in anticipation of some subsequent request to read from the console. The subsequent request to read from the console is usually the console read that occurs because of the PARSE EXTERNAL instruction in the SYSPROF EXEC. Note that, when in XA or XC virtual machines, the EXTERNAL() VM REXX built-in function can be used to test for the existence of *console-input-data* before the PARSE EXTERNAL is issued. • Users defined with the LBYONLY operand in the password field of the USER directory control statement cannot be logged on by the AUTOLOG command. **Upwardly compatible changes:** • Supports the System Console.
BACKSPAC	**Incompatible changes:** • Does not support logical printers. • Provides a one-line response instead of a two-line response, and the response is different. **Upwardly compatible changes:** • BACKWARD is a synonym.
BEGIN	**Upwardly compatible changes:** • Supports new operands: * and CPU {*cpuaddr(s)*|ALL}.

Table 36 (Page 3 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
CHANGE	**Incompatible changes:** • The default for the FLASH operand is ALL. In VM/ESA 1.1.5 370 Feature, the ALL operand did not exist, and the default was 0. • Truncates the *dsname* of the NAME operand to 16 characters; in VM/ESA 1.1.5 370 Feature, the *name* could be 24 characters long. • The response has changed, including: – The number of files changed may be up to seven digits; in System/370, it was only four digits. – Includes the number of records and KEEP setting in the response. – Includes the new user ID in the response if the file is changed to another user. **Upwardly compatible changes:** • Supports new operands: ALTID, KEEP, NOKEEP, MSG, NOMSG, NONAME, SECLABEL *seclabel*, and TO {*userid2*|*|ALTID} [**READER**|**PRINTER**|**PUNCH**]. • The following operands also support the new OFF operand: DIST, FORM, FLASH, MODIFY, FCB, and CHARS. • The DIST operand also supports *. • The DEST operand supports a new ANY operand. • Message HCP007E or HCP356E may now be issued if an external security manager is installed and you are not authorized to issue this command.
CLOSE	**Incompatible changes:** • Truncates the *dsname* of the NAME operand to 16 characters; in VM/ESA 1.1.5 370 Feature, the *name* could be 24 characters long. • The response has many changes because of the added function. **Upwardly compatible changes:** • New operands: – KEEP, NOKEEP – MSG, NOMSG – EOF – PURGE – CLASS *c* – COPY [*]*nnn* – [TO] {*userid*|*} [READER|PRINTER|PUNCH] – FLASH {*name* [ALL|*nnn*] | OFF} – MODIFY {*name* [*n*] | OFF} – FCB {*name*|6|8|10|12|OFF} – CHARS {*name(s)*|OFF} – NONAME – SYNCHRONOUS, ASYNCHRONOUS. • The DIST operand supports new OFF and * operands. • The FORM operand supports a new OFF operand. • The DEST operand supports a new ANY operand. • Message HCP007E may now be issued if an external security manager is installed and you are not authorized to issue this command.
COMMANDS	**Incompatible changes:** • The response does not include * in the list of authorized commands. • The responses for the QUERY and SET operands display individual operands of the QUERY and SET commands. In VM/ESA 1.1.5 370 Feature, the responses did not include individual QUERY and SET operands. **Upwardly compatible changes:** • You can specify a particular command to check the authorization of that command. **Note:** The new QUERY COMMANDS command will do the same thing as COMMANDS.

Table 36 (Page 4 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
COUPLE	**Incompatible changes:** • The responses have changed. The words COUPLE and DROP in the System/370 response are spelled COUPLED and DROPPED in the VM/ESA 2.4.0 response. **Upwardly compatible changes:** • Message HCP644E may now be issued if an external security manager is installed and you are not authorized to issue this command.
CPTRAP	**VM/ESA 1.1.5 370 Feature only.** The SET CPTRACE (A, C), TRSOURCE (A, C, E), and TRSAVE (A, C, E) commands provide the same functions, plus more. SET CPTRACE traces CP events, TRSOURCE traces virtual machine events, and TRSAVE records CP and virtual machine events. In VM/ESA 1.1.5 370 Feature, CPTRAP traces and records CP events and virtual machine events. In addition to the functions in CPTRAP, the VM/ESA 2.4.0 commands provide: • Two new trace types, I/O and data • The ability to collect data in multiple trace files, not just a single trace file • The ability to record CP trace data to tape, not just to DASD • The ability to define, enable, disable, and drop traces.
DCP	**VM/ESA 1.1.5 370 Feature only.** DISPLAY H replaces DCP. Note that DISPLAY H defaults to prefixing and DCP does not.
DEFINE (in general)	**Incompatible changes:** • The response contains four digits for virtual device numbers; in VM/ESA 1.1.5 370 Feature, it contained three digits. • The AS operand cannot be truncated to A; it could in VM/ESA 1.1.5 370 Feature. • VM/ESA 2.4.0 provides stricter syntax checking and therefore extra or erroneous parameters are rejected on DEFINE. In VM/ESA 1.1.5 370 Feature, extra or erroneous parameters were ignored. • See DEFINE commands below. **Upwardly compatible changes:** • Supports new operands: CPU, CRYPTO, MDISK, MSGPROC, VAFP, VECTOR, VFB-512. • The *vdev* can be four digits. In VM/ESA 1.1.5 370 Feature, it could only be three digits. In VM/ESA 2.4.0, device numbers are valid up to FFFF in XA, ESA, and XC virtual machines. In a 370 virtual machine, you can define virtual device numbers through FFFF, however, your virtual machine can use only the virtual devices numbered 0 through 1FFF. In VM/ESA 1.1.5 370 Feature, in basic control mode (ECMODE OFF), virtual address numbers could be no higher than 5FF. In extended control mode (ECMODE ON), all addresses through FFF were valid. • Additional messages: HCP045E, HCP260E, HCP1014E, HCP2800E, HCP2801E, HCP2802E, HCP2803E, HCP2804I, HCP2806E, HCP2811E.
DEFINE CHANNELS	**VM/ESA 1.1.5 370 Feature only.**
DEFINE Channel To Channel Adaptor	**Upwardly compatible changes:** • Supports new operands: USER *userid*, USER *.
DEFINE CONSOLE	**Upwardly compatible changes:** • Supports new operands: 3215 or 3270 for a virtual console device type; 3215 is the default.

Table 36 (Page 5 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
DEFINE GRAF	**Incompatible change:** • 3270 is the only device type operand supported. In VM/ESA 1.1.5 370 Feature, 3036, 3066, 3138, 3148, and 3158 were also supported.

DEFINE PRINTER	**Incompatible changes:**
	• The following are not supported following the PRINTER operand: 1403, 1443, 3203, 3211, 3262, 3289E, 3800, 3800-1, 3800-3, 4245, 4248. 1403 is assumed. This refers to the following format that was valid in VM/ESA 1.1.5 370 Feature:
	`DEFINE PRINTER AS vdev prt_type`
	The *prt_type* operand is not supported in VM/ESA 2.4.0.
	The following format is still supported, although *prt_type* of 1443 and 3289E are no longer supported.
	`DEFINE prt_type AS vdev`

DEFINE STORAGE	**Incompatible changes:** • Response has changed: – Seven digits are allotted for kilobytes and four digits for megabytes; in VM/ESA 1.1.5 370 Feature, five digits were allotted for kilobytes and two digits for megabytes. – You get “Storage cleared - system reset” rather than the “CP disabled” response that you got in VM/ESA 1.1.5 370 Feature. • Sizes are multiples of 64KB; in VM/ESA 1.1.5 370 Feature, sizes were multiples of 4KB. • The minimum storage size is 64KB. In VM/ESA 1.1.5 370 Feature, the minimum was 8KB. • For V=V guests, in VM/ESA 2.4.0 if the storage size you specify is not a multiple of 1MB and is not greater than 16MB, it is rounded to the next 64KB boundary. If the storage size you specify is not a multiple of 1MB and is greater than 16MB, it is rounded up to the next 1MB boundary. For V=F or V=R guests, in VM/ESA 2.4.0 the storage size is always rounded up to the nearest multiple of the minimum storage size for preferred guests. This storage size is at least 1MB. Use the QUERY V=R command to determine the preferred guests' minimum storage size. In VM/ESA 1.1.5 370 Feature, if the storage size you specified was not a multiple of 4KB, it was rounded up to the next 4KB boundary. **Upwardly compatible changes:** • The maximum storage size is 2047MB. In VM/ESA 1.1.5 370 Feature, the maximum storage size was 16MB.
DEFINE SYSVIRT	**VM/ESA 1.1.5 370 Feature only.**

Table 36 (Page 6 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
DEFINE (temporary disk)	**Incompatible changes:** • The following operands are no longer supported: T0671, T9313. • The T3310, T2314, and T2319 operands are ignored; you receive a message indicating that the temporary disk is not defined because space is not available. • The default is for only cylinder 0 or pages 0-3 of a temporary disk to get cleared when the disk is defined. In VM/ESA 1.1.5 370 Feature, the default was for all previously written data and directory areas that were in TDISK DASD space to be cleared to binary zeros. To change the setting so that data and directory areas are cleared, specify ENABLE CLEAR_TDISK on the FEATURES statement of the SYSTEM CONFIG file (new for VM/ESA 2.4.0). If you are not using a SYSTEM CONFIG file, specify SYSCLR=YES on the SYSRES macro in the HCPSYS file. If you are a class B user, you can determine if data and directory areas on temporary disks are cleared by using the new QUERY TDISKCLR command. If the response is: – “Temporary Disk Security is active.” Then all previously written data and directory areas that are in TDISK DASD space are cleared to binary zeros. – “Temporary Disk Security is inactive.” Then only cylinder 0 or pages 0-3 of a temporary disk is cleared. **Upwardly compatible changes:** • Supports new operands: T3390, T9332, T9335, T9336. **Note:** If your installation has applied the '9370 Processors, 9332 and 9335 Direct Access Storage Devices, and 9347 Tape Drive' between-release enhancement (APAR VM27115), then your system already supports T9332 and T9335.
DEFINE TIMER	**VM/ESA 1.1.5 370 Feature only.**
DEFINE VIRTUAL	**VM/ESA 1.1.5 370 Feature only.**
DEFINE 1443 3289E	**VM/ESA 1.1.5 370 Feature only.**
DEFINE 3088	**Upwardly compatible changes:** • New operands support: USER {*userid*|*}.
DETACH (in general)	**Incompatible changes:** • See DETACH commands below. **Upwardly compatible changes:** • The DETACH command supports additional functions such as dual copy, and Cryptographic Facility function. • Supports new operands: CPU, CRYPTO, MSGPROC. • Additional messages: HCP260E, HCP2805E, HCP2807E.
DETACH CHANNEL	**VM/ESA 1.1.5 370 Feature only.**

Table 36 (Page 7 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
DETACH *(real device)*	**Incompatible changes:** • Does not support the CHANNEL operand. Also, the CHANNEL response is removed. • The minimum truncation for the FROM operand is FR; in VM/ESA 1.1.5 370 Feature, it was F. • The channel-to-channel adaptor response contains “CTCA” instead of “CTC” and “DROPPED” instead of “DROP.” **Upwardly compatible changes:** • Supports the new: – GIVE command function – Dynamic switching devices – Dual copy function of the 3990-3. • Responses may contain new fields or values for new support. Also, there are new responses for support of the new GIVE command function. • If the device is part of a CONCOPY session, the session is terminated. • There is a new response indicating termination of a CONCOPY session.
DETACH *(virtual device)*	**Incompatible changes:** • The channel-to-channel adaptor response contains “CTCA” instead of “CTC” and “DROPPED” instead of “DROP.” **Upwardly compatible changes:** • Supports a new operand: VIRTUAL. • Supports dynamic switching devices. • If the device is part of a CONCOPY session, the session is terminated. • Some responses may contain new values to support new function. • Supports new responses for new support.
DIAL	**Incompatible changes:** • The response has five digits for the number of users; in VM/ESA 1.1.5 370 Feature, the response had three or four digits.
DISABLE	**Upwardly compatible changes:** • DISABLE is now a class F command as well as A or B.
DISCONN	**Incompatible changes:** • Response has five digits for the number of users; in VM/ESA 1.1.5 370 Feature, it was three or four digits. **Upwardly compatible changes:** • Supports the System Console. • The full form of the command is DISCONNECT; in VM/ESA 1.1.5 370 Feature, it was DISCONN.

Table 36 (Page 8 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
DISPLAY (in general)	**Incompatible changes:** • The following operands are no longer supported: CAW, CSW. **Upwardly compatible changes:** • New operands are supported: – ACCLIST (access list) – CDX (crypto domain index) – (ESA/XC storage) – H (host storage) – LKS (linkage stack) – PREFIX – SCHIB (subchannel information blocks) – VR, VP, VMR, VAC, VSS (vector) • The DISPLAY command supports additional functions, such as access registers and the Cryptographic Facility.
DISPLAY Guest Storage	**Incompatible changes:** • The response may be different. **Upwardly compatible changes:** • Allows you to control the display of the contents of storage with the following new operands: – BASE*nn*, INDEX*nn*, PRI, SECO, HOME, ASTE*raddr*., STD*hexword*., ASN*asn*., AREG*areg*., ALET*hexword*., ALET*hexword*.AL*raddr*, SPACE*spaceid*., ASIT*asit*.. In VM/ESA 1.1.5 370 Feature, there was no way to display third-level virtual storage directly. You had to convert your third-level storage locations to second level storage locations before issuing DISPLAY. • The following operands have been added for displaying guest storage: R displays the contents of second-level storage S displays storage in hexadecimal translation as a string I displays storage in instruction format N displays storage without EBCDIC translation (this is the default) U displays storage in EBCDIC translation. • Responses that display third-level storage include an additional field appended at the end of the response that shows the second-level address that corresponds to the third-level address. • The L operand will display the contents of: – Second-level storage if the virtual machine is operating in DAT OFF mode – Third-level storage if the virtual machine is operating in DAT ON mode. In VM/ESA 1.1.5 370 Feature, L only displayed second-level virtual storage. • *hexloc* operands can be one to eight hexadecimal digits; in VM/ESA 1.1.5 370 Feature, they could only be one to six hexadecimal digits.
DISPLAY Program Status Words	**Upwardly compatible changes:** • Supports new operands: EXT, SVC, PROG, PRG, MCH, I/O, ALL. • Adds a response for the new operands.

Table 36 (Page 9 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
DISPLAY Registers	**Upwardly compatible changes:** • Supports new operands: AR, FPC. • Has a response for the new operand. • The response to the X*reg1* operand includes all 16 control registers; VM/ESA 1.1.5 370 Feature included only control register 0 when running with ECMODE OFF. • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New response when FPC operand is used. • New messages: HCP6153E, HCP6154E.
DMCP	**VM/ESA 1.1.5 370 Feature only.** DUMP H replaces DMCP. Note that DUMP H defaults to prefixing and DMCP does not.
DRAIN	**Incompatible changes:** • The response may be different. It shows the LIMIT value as set by CP START command or the system configuration file. **Upwardly compatible changes:** • The response provides more detail (two added lines) concerning the device being drained. • Supports a new operand: UR.
DUMP (in general)	**Incompatible changes:** • See DUMP commands below. **Upwardly compatible changes:** • New operands supported: – ACCLIST (access list) – CDX (crypto domain index) – (ESA/XC storage) – H (host storage) – LKS (linkage stack) – PREFIX – PSW (program status words) – G, X, AR, Y (registers) – SCHIB (subchannel information blocks) – VR, VP, VSR, VMR, VAC, VSS (vector -information when a Vector Facility is defined).

Table 36 (Page 10 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
DUMP Guest Storage	**Incompatible changes:** • The response may be different. **Upwardly compatible changes:** • Allows you to control the dump of the contents of storage with the following new operands: PRI, SECO, HOME, ASTE*raddr*., STD*hexword*., ASN*asn*., AREG*areg*., ALET*hexword*., ALET*hexword*.AL*raddr*, SPACE *spaceid*., ASIT*asit*.. In VM/ESA 1.1.5 370 Feature, there was no way to dump third-level virtual storage directly. You had to convert your third-level storage locations to second level storage locations before issuing DUMP. • The following operands for dumping guest storage have been added: K dumps the storage keys in hexadecimal R dumps the contents of second-level storage S dumps storage in hexadecimal translation as a string I dumps storage in instruction format N dumps storage without EBCDIC translation (this is the default) U dumps storage in EBCDIC translation BASE *nn* and INDEX*nn* dump the contents of guest storage using the specified starting address (*hexloc1*) as a displacement from the address in the specified general register. • Responses that dump third-level storage include an additional field appended at the end of the response that shows the second-level address that corresponds to the third-level address. • The L operand will dump the contents of: – Second-level storage if the virtual machine is operating in DAT OFF mode. – Third-level storage if the virtual machine is operating in DAT ON mode. In VM/ESA 1.1.5 370 Feature, L only dumped second level virtual storage. • *hexloc* operands can be one to eight hexadecimal digits; in VM/ESA 1.1.5 370 Feature, they could only be one to six hexadecimal digits.
ECHO	**Incompatible changes:** • The response includes a period at the end and is mixed case; in VM/ESA 1.1.5 370 Feature, there is no period at the end and the text is all uppercase.
ENABLE	**Incompatible changes:** • Logical devices are not supported. **Upwardly compatible changes:** • ENABLE is also a class F command, as well as A or B.
EXTERNAL	**Incompatible changes:** • The default is KEY; in VM/ESA 1.1.5 370 Feature, the default was 40. **Upwardly compatible changes:** • Supports new operands that allow you to specify the type of interrupt: KEY, INTERVAL, MALFUNCTION, EMERGENCY, CALL, *cpuaddr*, CR0 *bitno*, CMD *command*.
FLUSH	**Incompatible changes:** • The response is different. • The response is on a single line. In VM/ESA 1.1.5 370 Feature, the response was on two lines.

Table 36 (Page 11 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
FORCE	**Incompatible changes:** • Response may have changed: – The number of users still logged on the system in the response that the system operator receives may be five digits; in VM/ESA 1.1.5 370 Feature, it was three or four digits. **Upwardly compatible changes:** • If the user who is being forced from the system has an open reader file and the user's virtual reader is spooled NOHOLD and NOKEEP, the open file is retained on the user's reader queue in USER hold status. In VM/ESA 1.1.5 370 Feature, the open file was purged when the user was forced off. Note that the open file is still purged when the user logs off. • New operands have been added: LOGOFF and DISCONNECT. LOGOFF is the default. • The response may contain new information: – Whether a user ID or the SYSTEM issued the FORCE. – Indication that a DISCONNECT was forced. This would occur when FORCE was issued with the DISCONNECT operand.
FREE	**Upwardly compatible changes:** • Adds a class B operand to release a *userid* from HOLD LOGON status: LOGON.
GIVE	**Incompatible changes:** • Using a *userid* of `'FR'`, `'FRO'`, or `'FROM'` may cause unpredictable results. • More variations of parameters are supported. If you omit too many optional parameters, such as TO, FROM, or AS, you may receive a new message that indicates that GIVE could not decide how to interpret your parameters. • Responses may be different. **Upwardly compatible changes:** • Supports new operands: VIRTUAL, LEAVE, UNLOAD, R/W. • Supports real devices. For real devices, an additional *userid* operand is required to indicate the virtual machine to which the device is currently attached. • `'*'` for a *userid* indicates your own virtual machine.

HALT	**Incompatible changes:**
	• The response has changed. You will now get one of the following responses:
	`HALT INITIATED TO type rdev`
	`HALT NOT INITIATED BECAUSE type rdev IS NOT ACTIVE`
	In VM/ESA 1.1.5 370 Feature, you would get the following response:
	`DEVICE HALTED`
	Upwardly compatible changes:
	• Supports dynamic switching devices.

HOLD	**Upwardly compatible changes:** • Adds a class B operand to prevent a *userid* from accessing the host system: LOGON.
INDICATE (in general)	**Incompatible changes:** • See INDICATE commands below. **Upwardly compatible changes:** • Supports new operands: NSS, SPACES, ACTIVE.

Table 36 (Page 12 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
INDICATE FAVORED	**VM/ESA 1.1.5 370 Feature only.** Use QUERY SHARE or QUERY QUICKDSP instead. QUERY SHARE is similar to INDICATE FAVORED %. QUERY QUICKDSP is similar to INDICATE FAVORED without a percent.
INDICATE I/O	**Upwardly compatible changes:** • The I/O operand can be issued by a class B or class E user; in VM/ESA 1.1.5 370 Feature, only a class E user could issue it.
INDICATE LOAD	**Incompatible changes:** • The response is different. **Upwardly compatible changes:** • Supports new operands for CSE complexes: AT *sysname*, AT *.
INDICATE PAGING	**Incompatible changes:** • Has changes in the response: – The number of pages may be five digits. In VM/ESA 1.1.5 370 Feature, the number of pages was up to four digits. – The format of the response when there are no users in page wait has also changed: - In VM/ESA 2.4.0: `NO USERS IN PAGE WAIT` - In VM/ESA 1.1.5 370 Feature: `NO USERS IN PAGEWAIT` • The information in the response is different: – The first number contains all zeros unless expanded storage is available. If expanded storage is available, this number is the pages the user has in XSTORE blocks. In VM/ESA 1.1.5 370 Feature, this number was the pages, in hexadecimal, that CP allocated on drum storage for the user. – The second number is the decimal number of pages allocated on auxiliary storage. In VM/ESA 1.1.5 370 Feature, this number was the pages, in hexadecimal, that CP allocated on disk storage for the user.
INDICATE QUEUES	**Incompatible changes:** • The response has changed: – A field is inserted after the *userid* field. – Q0 is a new list status, which means in the dispatch list and exempt from eligible list delays. – The number of pages is decimal; in VM/ESA 1.1.5 370 Feature, it was hexadecimal. – Some of the status indicators are different and you can receive a new one. – The order of the response is different: 1. All dispatch list users (“Qn”); in VM/ESA 1.1.5 370 Feature, this was last. 2. The eligible list users (“En”); in VM/ESA 1.1.5 370 Feature, this was first. **Upwardly compatible changes:** • Supports a new operand for expanded information: EXP. • A new response displays expanded information for the new EXP operand.
INDICATE USER	**Incompatible changes:** • The response has changed and includes more information. • Total CPU usage and virtual CPU usage information in the response is different. It is reset to zero whenever an ACNT command is issued. In VM/ESA 1.1.5 370 Feature, this information was never reset. • Using a *userid* of SYSTEM gives you the system's resource information rather than the user ID SYSTEM's information. **Upwardly compatible changes:** • Supports new operands: SYSTEM, EXPANDED.

Table 36 (Page 13 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
IPL	**Incompatible changes:** • The STOP operand halts the virtual machine during the IPL procedure just after the initial PSW is loaded. In VM/ESA 1.1.5 370 Feature, it does this just **before** the initial PSW is loaded. • A new response is issued if tracing is active when a preferred guest is IPLed. Tracing must be turned off and the guest reIPLed. • In VM/ESA 1.1.5 370 Feature, when you IPL CMS with the PARM operand, CMS initialization attaches a fence of 8 bytes of `X'FF'`s to the end of the PARM data before passing it to the SYSPROF EXEC. In VM/ESA 2.4.0, 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. **Upwardly compatible changes:** • Supports integrated-adaptor-connected FBA DASD (9335 and 9336 only) as the IPL device. • Message HCP449E may now be issued if an external security manager is installed and you are not authorized to issue this command. • Terminates all CONCOPY sessions for the virtual machine.
LINK	**Incompatible changes:** • The AS operand must be entered as “AS”; in VM/ESA 1.1.5 370 Feature, “A” was also accepted. • The response has five digits for the number of users; in VM/ESA 1.1.5 370 Feature, it had three or four digits. • The *password* operand cannot be the same as the new access modes: SR, SW, SM, ER, or EW. • You receive new messages if you are linking to a disk that someone else has linked to using the new access modes (SR, SW, SM, ER, or EW). **Upwardly compatible changes:** • For multiple exposure devices, you must specify the base exposure; LINK provides access to all exposures. • Supports five new access modes: stable read-only, stable write, stable multiple, exclusive read-only, exclusive write. New operands provide the support: – SR (stable read-only) – SW (stable write) – SM (stable multiple) – ER (exclusive read-only) – EW (exclusive write). **Note:** Because these new link modes are not supported in earlier releases of VM, they will not function in a CSE complex that has one or more systems operating before VM/ESA 1.1.1. If you attempt to link with one of these new modes in a CSE complex that has a pre-VM/ESA 1.1.1 system, you receive an error message. • There are 50 levels of link indirection supported. In VM/ESA 1.1.5 370 Feature, the limit was three. • The following messages may now be issued if an external security manager is installed: – HCP298E, if you are not authorized to issue the command – HCP1156I, if you are not authorized for R/W access.

Table 36 (Page 14 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
LOADBUF	**Incompatible changes:** • New message: HCP1562E. • If you specify INDEX 0, indexing is turned off. In VM/ESA 1.1.5 370 Feature, message DMK036E ("Index missing or invalid") was issued. **Upwardly compatible changes:** • A new NOFOLD operand is supported and is the default; in VM/ESA 1.1.5 370 Feature, the lack of the FOLD operand implied NOFOLD.
LOADVFCB	**Incompatible changes:** • New message: HCP1562E. **Upwardly compatible changes:** • Supports a new operand for specifying the image library which contains the specified FCB: IMAGE *imagelib*. If you do not specify the IMAGE operand you will get the default image library for the device type.
LOCATE	**Incompatible changes:** • Because VM/ESA 2.4.0 uses different control blocks, different control blocks get displayed. **Upwardly compatible changes:** • Supports an added operand to display the address of a module identified by a symbolic label: *symbol*. • A response is added to support the new *symbol* operand.
LOCK	**Upwardly compatible changes:** • Supports a new operand, SYMBOL *symbol*, which locks any pages containing CP system symbols that match the specified *symbol*. Pattern-matching characters may be specified in *symbol*. MAP is the default if SYMBOL is used. NOMAP may be specified. • Supports new operands: USERID, ASIT, and SPACE.
LOGOFF or LOGOUT	**Incompatible changes:** • The response sent to the primary system operator has five digits for the number of users; in VM/ESA 1.1.5 370 Feature, it had three or four. **Upwardly compatible changes:** • Supports the System Console. • Can be used to cause a break-out from a wait state caused by APPC/VM, a DIAGNOSE code X'A4', or a DIAGNOSE code X'A8'.

Table 36 (Page 15 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
LOGON or LOGIN	**Incompatible changes:** • The *console-input-data* gets read during initialization of the specified *userid.* In VM/ESA 1.1.5 370 Feature, it did not get read until after initialization of the specified *userid*. • Users' passwords cannot be any of the allowable CP LOGON operands if password suppression is OFF. See the new operands listed under upwardly compatible changes to see which new words now cannot be passwords. • The LOGON response has changed: – Five digits for the number of users; was three or four in VM/ESA 1.1.5 370 Feature. – Four digits for the device address; was three in VM/ESA 1.1.5 370 Feature. – Four digits for the number of files; was three in VM/ESA 1.1.5 370 Feature. • A LOGON of a user with LBYONLY in the password field of the USER directory control statement can only be done by an authorized user using the new BY option of the LOGON command. **Upwardly compatible changes:** • Supports the System Console. • Supports new operands: IPL *vdev*, IPL *sysname*, ACCOUNT *nnnnnnnn*, NORUN, APL, HERE, TEXT, SECLABEL *seclabel*, STORAGE *nnnnnnn*K STORAGE *nnnn*M, QUIET, *#console-input-data*, MACHINE XA, MACHINE ESA, MACHINE XC, BY *byuserid*. • XC virtual machines are logged on as XA virtual machines if CP is not running on an ESA/390 processor. • New responses for the new HERE and MACHINE operands. • New message: HCP2808E.
MESSAGE	**Incompatible changes:** • The response to the person receiving the message may have changed: – The time the message was sent is included on the same line as the message text; in VM/ESA 1.1.5 370 Feature, some responses had the time stamp on a separate line from the message text. – The FROM *xxxxxxxx* field may be different. • Supports new operands for CSE complexes: AT ALL, AT *sysname*, AT *. If you use 'AT' as the first part of your message text, you receive message HCP2970E, which indicates that the CSE name is missing or invalid. In VM/ESA 1.1.5 370 Feature, you could use 'AT' as the first part of your message text. **Upwardly compatible changes:** • Added two operands for handling mixed DBCS input: ALLDBCS, ALLSBCS. DBCS text is supported for the *msgtext* parameter. • Message HCP003E may now be issued if an external security manager is installed and you are not authorized to issued this command.
MIGRATE	**VM/ESA 1.1.5 370 Feature only.**
MONITOR	**Incompatible changes:** • Has a different structure. • Produces different data records. • Has different collection mechanisms.

Table 36 (Page 16 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
MSGNOH	**Incompatible changes:** • Is a privilege class B only command. In VM/ESA 1.1.5 370 Feature, it was also a class A command. • Supports new operands for CSE complexes: AT ALL, AT *sysname*, AT *. If you use 'AT' as the first part of your message text you receive message HCP2970E, which indicates that the CSE name is missing or invalid. In VM/ESA 1.1.5 370 Feature, you could use 'AT' as the first part of your message text. **Upwardly compatible changes:** • Supports two new operands for handling mixed DBCS input: ALLDBCS, ALLSBCS. DBCS text is supported for the *msgtext* parameter. • Message HCP003E may now be issued if an external security manager is installed and you are not authorized to issue this command.
NETWORK	**VM/ESA 1.1.5 370 Feature only.** In VM/ESA 2.4.0, VM/Pass-Through Facility commands as well as the CCLOAD and CCDUMP utilities provide equivalent functions. Also, the NCPDUMP command is not available in VM/ESA 2.4.0.
ORDER	**Incompatible changes:** • The response can have seven digits; in VM/ESA 1.1.5 370 Feature, it could have four or six digits. **Upwardly compatible changes:** • The class G command supports the new *userid* operand.
PER	**Incompatible changes:** • The class G PER command is a synonym for the CP TRACE command, which has invocation, response, and functional differences. See the TRACE command description in *VM/ESA: CP Command and Utility Reference*. It provides certain equivalent functions with different operands: – SAVE is provided by GOTO and APPEND. – GET is provided by GOTO and APPEND. – END CURRENT is provided by applying CLEAR to the active trace set. – END *ident* is provided by DELETE. – END *set* is provided by CLEAR. – END *traptype* has the same result achieved by deleting each trap separately. – GUESTR is provided by NODAT. – GUESTV is provided by DAT.
PURGE	**Incompatible changes:** • ALL is not supported for device type; UR replaces it. Note: ALL is still supported to specify that all files on a particular device (RDR, PCH, PRT, UR) be purged. • The response may have seven digits for the number of files purged; in VM/ESA 1.1.5 370 Feature, it could have had four digits.
QUERY (in general)	**Incompatible changes:** • See QUERY commands below. **Upwardly compatible changes:** • See QUERY commands below.
QUERY AFFINITY	**VM/ESA 1.1.5 370 Feature only.**
QUERY ALL	**Incompatible changes:** • Response has changed.

Table 36 (Page 17 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
QUERY CHANNELS	**VM/ESA 1.1.5 370 Feature only.**
QUERY CPASSIST	**VM/ESA 1.1.5 370 Feature only.**
QUERY CPLEVEL	**Incompatible changes:** • The format of the response has changed to include the version. In addition, the service level is returned as *yynn*, where *yy* is the last two digits of the year and *nn* is the sequential number of the RSU tape for that year. In VM/ESA 1.1.5 370 Feature, the service level was returned as *nnnn*, which was a sequential number referring to the service tape. **Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • In the response, the release level value has changed. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q CPLEVEL command, the output from Q 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.
QUERY CPTRAP	**VM/ESA 1.1.5 370 Feature only.** The QUERY CPTRACE, QUERY TRSOURCE, and QUERY TRSAVE commands provide the same functions, plus more.
QUERY DASD	See also QUERY VIRTUAL DASD. **Incompatible changes:** • You can get an additional response for offline DASDs. **Upwardly compatible changes:** • The ATTACH operand supports an added *userid* operand. • The real device address of offline DASD is echoed in a new response.
QUERY DUMP	**Incompatible changes:** • Response may be different: – May return a printer as a dump unit. – Has no TEMP or DUMP fields. – May have “V=R” in the CP or ALL field. **Upwardly compatible changes:** • Has additional fields appended in the response: IPL, NOIPL, NOPRINT.
QUERY FILES	**Upwardly compatible changes:** • The class G response has four digits; in VM/ESA 1.1.5 370 Feature, it had three. • The class D response may have seven digits; in VM/ESA 1.1.5 370 Feature, it had four. • Supports new operands: SYSTEM, ALL (the default), ALTID, *spoolid*, XFER *userid2*, XFER ALL, XFER ALTID, AVAIL.
QUERY HOLD	**Incompatible changes:** • Response has changed: – Can contain seven digits in the response; in VM/ESA 1.1.5 370 Feature, could contain three or four. – Does not contain RDR in the response. **Upwardly compatible changes:** • Is also a class B command with the required LOG operand. The response displays the number of files belonging to users in logon HOLD status.

Table 36 (Page 18 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
QUERY LINKS	**Upwardly compatible changes:** • A new operand, STABLE, displays the status of links accessed with stable read-only or stable write modes.
QUERY (logical devices)	**Incompatible changes:** • Response has changed: – "LDEV" is replaced with "PRT," "GRAF," or "DEV." **Upwardly compatible changes:** • Supports specification of more than one logical device or a range of logical devices. • Supports a new operand: LDEV. • When LDEV is specified, the response includes the user ID of the virtual machine that created the logical device. • The response may include the IP address for TCP/IP.
QUERY LOGMSG	**Upwardly compatible changes:** • Issues a message if no LOGMSG exists. • The class B response includes log message line numbers. • Supports new operands: ACCOUNT, ACIGROUP, SYSTEM, USERID, CONTENTS, ORIGIN, DBCS, SBCS. • Supports DBCS log messages.
QUERY MITIME	**Incompatible changes:** • Response is different. **Upwardly compatible changes:** • A class A or B user is allowed to issue this command. In VM/ESA 1.1.5 370 Feature, only a class B could issue this command.
QUERY NAMES	**Incompatible changes:** • Response has changed: – Can contain new responses if your installation is using System Console support: `LOGNSYC   - SYSC` *`userid`*`   - SYSC` – Contains LOGN*xxxx* instead of LOGO*xxx* and contains LOGL*xxxx* instead of LOGOL*xxx*. **Upwardly compatible changes:** • Supports new operands for CSE complexes: AT ALL, AT *sysname*, AT *. • Supports the EXTended parameter to display the network qualifiers.
QUERY PAGING	**VM/ESA 1.1.5 370 Feature only.**
QUERY PATHS	**Incompatible changes:** • "QUERY PATHS *rdev*" replaces "QUERY *type* PATHS." QUERY PATHS displays all paths installed to a specified device and their current status (online or offline); in VM/ESA 1.1.5 370 Feature, QUERY *type* PATHS displays the path status for all alternate path devices and the processor address for each device in MP mode. • Response may be different.
QUERY PER	**Incompatible changes:** • Is a synonym for the QUERY TRACE command. • Operands and responses are different.
QUERY PF	**Incompatible changes:** • Response may be different.

Table 36 (Page 19 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
QUERY PRINTER	**Incompatible changes:** • See information under QUERY READER/PRINTER/PUNCH.
QUERY PRIORITY	**VM/ESA 1.1.5 370 Feature only.**
QUERY PROCESSR	**Incompatible changes:** • This is a class A, B, C, or E command. In VM/ESA 1.1.5 370 Feature, it was a class G command as well. • Response is different. **Upwardly compatible changes:** • You can spell PROCESSOR out fully, although it accepts PROCESSR; in VM/ESA 1.1.5 370 Feature, it had to be spelled PROCESSR.
QUERY PUNCH	**Incompatible changes:** • See the information for QUERY READER/PRINTER/PUNCH.
QUERY QDROP	**VM/ESA 1.1.5 370 Feature only.**
QUERY READER / PRINTER / PUNCH	**Incompatible changes:** • Changed responses: – The EXP operand, which used to be the TBL operand, response is changed. The FLASH count field is in a different position. – You may get different responses. The DATE and TIME fields may contain different information. **Upwardly compatible changes:** • Supports new operands: AVAIL, EXP, SYSTEM, SECLABEL *seclabel*, XFER *userid*, XFER ALL, XFER ALTID, PREVOWN *userid* ALL, PREVOWN *userid* ALTID. • The EXP operand replaces the TBL operand, but TBL is still accepted. • Changed responses: – The ALL operand response appends fields for KEEP and MSG settings. – The PSF operand response appends a field for PURGE status. – NORESCAN or RESCAN information is provided for QUERY VIRTUAL READER. • The response may include the security label (SECLABEL) and certain fields may contain asterisks if an external security manager is installed. (Asterisks are used to mask fields that the user is not authorized to view while logged on with the current security label.) • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses include 4-digit years for FULLDATE and ISODATE operands. • 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. • 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 Q RDR|PRT|PUN command, the output from Q 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).

Table 36 (Page 20 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
QUERY (real device)	**Incompatible changes:** • The STATUS operand must come after *rdev* instead of before it; in other words, QUERY *rdev* STATUS replaces QUERY STATUS *rdev*. • SYSVIRT and VIRTUAL operands are not supported. • “QUERY *type* PATHS” is replaced with “QUERY PATHS *rdev*.” QUERY PATHS displays all paths installed to a specified device and their current status (online or offline); in VM/ESA 1.1.5 370 Feature, QUERY *type* PATHS displayed the path status for all alternate path devices and the processor address for each device if in MP mode. • Two-line responses are now one-line responses and are changed. **Upwardly compatible changes:** • Supports new operands: – *userid* operand with ATTACH – DASD DETAILS *rdev(s)* – DASD DRAINING *rdev(s)* – BOXED. • Supports dynamic switching devices and dual copy function of the 3990-3. • New responses for new support. • Responses may contain new fields for new support.
QUERY SASSIST	**VM/ESA 1.1.5 370 Feature only.**
QUERY SCREEN	**Upwardly compatible changes:** • The minimum abbreviation for SCREEN is SCR; in VM/ESA 1.1.5 370 Feature, it was SCRE.
QUERY SECUSER	**Upwardly compatible changes:** • New operands: *, ALL, *userid*. • Displays new responses.
QUERY SET	**Upwardly compatible changes:** • Changed response: – The response appends additional information: MACHINE XA, MACHINE ESA, MACHINE XC, SVC76 CP, SVC76 VM, NOPDATA ON, NOPDATA OFF, IOASSIST ON, IOASSIST OFF, 370ACCOM ON, 370ACCOM OFF. – Some fields appear for compatibility purposes, but are not supported by the SET command as they were in VM/ESA 1.1.5 370 Feature. These settings cannot be changed: ACNT OFF, ISAM OFF, ECMODE ON, ASSIST OFF, AFFINITY NONE, VMSAVE OFF, 370E OFF, STBYPASS OFF, STMULTI OFF, and SVCACCL OFF.
QUERY SPMODE	**VM/ESA 1.1.5 370 Feature only.**
QUERY SRM	**Incompatible changes:** • The following operands are not supported: APAGES, IB, MAXDRUM, MHFULL, PB, PCI, PGMSTAT, PGMTLIM. • The response is different. **Upwardly compatible changes:** • Supports new operands: ALL, IABIAS, LDUBUF, STORBUF, DSPBUF.
QUERY STATUS	**VM/ESA 1.1.5 370 Feature only.** Use QUERY *rdev* STATUS instead.
QUERY STORAGE	**Incompatible changes:** • Response changed: – The number of kilobytes of storage may be seven digits; it was five digits in VM/ESA 1.1.5 370 Feature. – The response may be in megabytes of storage instead of kilobytes.

Table 36 (Page 21 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes

QUERY SYSTEM	**Incompatible changes:**
	• The new operand, ALL, is the default. It displays information about all system volumes.
	• Has an added response:
	`DASD rdev ATTACHED {SYSTEM│CPVOL} nnnn volid`
	Upwardly compatible changes:
	• Supports new operands: ALL, TDISK, TDSK.
	• You can display usage information about only temporary disks; in VM/ESA 1.1.5 370 Feature, information was displayed about all minidisks, including temporary disks.
	• Has an added response:
	`DASD TDSKS NOT FOUND`

QUERY S370E	**VM/ESA 1.1.5 370 Feature only.**
QUERY TAG	**Upwardly compatible changes:** • Message HCP026E may now be issued if an external security manager is installed and you are not authorized to issue this command. • Message HCP6525E may now be issued if an external security manager is installed but unavailable to verify authorization.
QUERY TDSK	**Incompatible changes:** • Response has changed. • Has an added response: `DASD rdev ATTACHED CPVOL nnnn volid` **Upwardly compatible changes:** • TDISK is a supported spelling for the TDSK operand; in VM/ESA 1.1.5 370 Feature, only TDSK was supported.
QUERY TERMINAL	**Incompatible changes:** • Changed response: – Additional values are possible for the CONMODE and BRKKEY fields in the response. **Upwardly compatible changes:** • The response has changed: – The SCRNSAVE field is always OFF. SCRNSAVE cannot be set ON in VM/ESA 2.4.0. – Has appended fields: AUTOCR, MORE, HOLD.
QUERY TIME	**Upwardly compatible changes:** • Supports a new operand: OFFSET. • Displays new responses. • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q TIME command, the output from Q TIME uses the new default date format. This causes the date to be expanded to include the 4-digit year.

Table 36 (Page 22 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
QUERY UR	**Incompatible changes:** • Now is a class B command. • Does not support the PATHS operand. Use “QUERY PATHS *rdev*” instead. QUERY PATHS displays all paths installed to a specified device and their current status (online or offline); in VM/ESA 1.1.5 370 Feature, QUERY *type* PATHS displayed the path status for all alternate path devices and the processor address for each device if in MP mode. • Does not support logical printers. **Upwardly compatible changes:** • Has added operands: SYSTEM, *userid* (with the ATTACH operand).
QUERY USERID	**Upwardly compatible changes:** • Supports new operands for a CSE complex: AT ALL, AT *sysname*, AT *. • The response can be different if you are using the System Console: `userid   - SYSC` • Supports the EXTended parameter to display the network qualifiers.
QUERY USERS	**Incompatible changes:** • Displays five digits for the number of users; VM/ESA 1.1.5 370 Feature displayed four digits. **Upwardly compatible changes:** • Supports new operands for a CSE complex: AT ALL, AT *sysname*, AT *. • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL ALL	**Incompatible changes:** • Changed response: – Does not display CHANNELS information. – Has added information. **Upwardly compatible changes:** • Supports dynamic switching devices and the Cryptographic Facility. • Supports virtual message processors.
QUERY VIRTUAL CHANNELS	**VM/ESA 1.1.5 370 Feature only.**
QUERY VIRTUAL CONSOLE	**Incompatible changes:** • The response is different: – FOR in VM/ESA 1.1.5 370 Feature response is TO in the VM/ESA 2.4.0 response. – COPY field may be CPY **nnn* or COPY *nnn*. In VM/ESA 1.1.5 370 Feature, it was only COPY *nnn*. • The response may include a new line containing TCP/IP information. **Upwardly compatible changes:** • The response has new information. • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL DASD	**Incompatible changes:** • The response has eight digits for cylinders; in VM/ESA 1.1.5 370 Feature it had four digits. **Upwardly compatible changes:** • Supports a new operand: DETAILS. • Supports dual copy function of the 3990-3. • Changed response; new information may be appended. • In the response, the *volid* field can be (VDSK) and the *rdev* field can be VDSK.

Table 36 (Page 23 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
QUERY VIRTUAL GRAF	**Incompatible changes:** • The response may be different. **Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL LINES	**Incompatible changes:** • The response may be different. **Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL STORAGE	**Incompatible changes:** • Response is changed because the maximum storage value is 2047MB. – The number of kilobytes of storage may be seven digits; it was five digits in VM/ESA 1.1.5 370 Feature. – The response may be in megabytes of storage instead of kilobytes.
QUERY VIRTUAL TAPES	**Incompatible changes:** • The response has changed. It also appends new information.
QUERY VIRTUAL UR	**Incompatible changes:** • The response is different.
QUERY VIRTUAL *vaddr*	**Upwardly compatible changes:** • New operand: DETAILS. • Supports dual copy function of the 3990-3. • Appends new information in the response. • Response indicates REAL-MPLF or SIMULATED-MPLF if enabled for the device. • Supports virtual message devices. The device type MSGD appears in the response.
QUERY VMSAVE	**VM/ESA 1.1.5 370 Feature only.**
QVM	**VM/ESA 1.1.5 370 Feature only.**
REPEAT	**Incompatible changes:** • Does not support logical printers. • Removed operand: HOLD. • Changed response: – Some of the fields are different. – The response is one line; in VM/ESA 1.1.5 370 Feature, it was two lines.
RESET	**Upwardly compatible changes:** • Response has appended fields for the type and virtual device number of the device that has been reset. • A class B user can enter new operands: RESERVE ON *rdev*.
REWIND	**Incompatible changes:** • New message, HCP1122E, is issued if the tape drive was given to another user ID.
SAVESYS	**Incompatible changes:** • Function is different because of changed saved segment support. • Responds with a message instead of a response for successful completion.

Table 36 (Page 24 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
SCREEN	**Incompatible changes:** • Local terminals and logical devices display incoming MSGs, WNGs, and MSGNOHs in the CPOUT color. In VM/ESA 1.1.5 370 Feature, they were displayed in the INREDISP color. Certain other terminal output, such as password prompts and spool file notices are similarly affected. SNA (VTAM) terminal users are unaffected, because they continue to use CPOUT for this category of output.
SEND	**Incompatible changes:** • The text that you send when you do not specify the CP operand has changed. In VM/ESA 2.4.0, if the first four characters of text you want to send are: (1) your logical escape character, (2) the receiver's logical line end character, (3) C, (4) P, then you need not specify the CP operand. In VM/ESA 1.1.5 370 Feature, if the first 3 characters of the text were '#CP' you did not need to specify the CP operand. **Upwardly compatible changes:** • Added capability for handling mixed DBCS input for the *text* operand.
SET (in general)	**Incompatible changes:** • See SET commands below. **Upwardly compatible changes:** • See SET commands below.
SET ACNT	**VM/ESA 1.1.5 370 Feature only.**
SET AFFINITY	**VM/ESA 1.1.5 370 Feature only.**
SET ASSIST	**VM/ESA 1.1.5 370 Feature only.**
SET CPASSIST	**VM/ESA 1.1.5 370 Feature only.**
SET DUMP	**Incompatible changes:** • Does not support the AUTO operand; use SET DUMP DASD or SET DUMP *rdev* instead. • Returns a different response. **Upwardly compatible changes:** • Lets you specify the order in which DASD volumes are searched for spooling space to be assigned as system abend dump space. • Supports new parameters: DASD (which replaces AUTO), V=R, IPL, NOIPL, NOPRINT.
SET ECMODE	**VM/ESA 1.1.5 370 Feature only.**
SET EMSG	**Incompatible changes:** • EMSG is set to ON when you log on. In VM/ESA 1.1.5 370 Feature, it was set to TEXT. • CMS recognizes EMSG settings for all error (E) and information (I) messages; in VM/ESA 1.1.5 370 Feature, CMS also recognized EMSG settings all warning (W) messages.
SET FAVORED	**VM/ESA 1.1.5 370 Feature only.** SET SHARE is similar to SET FAVORED with a percent. SET QUICKDSP is similar to SET FAVORED without a percent.
SET IMSG	**Incompatible changes:** • Does not control responses issued by the IPL command.
SET ISAM	**VM/ESA 1.1.5 370 Feature only.**
SET LOGMSG	**Upwardly compatible changes:** • Added capability for handling mixed DBCS input for the *text* operand.

Table 36 (Page 25 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
SET MITIME	**Upwardly compatible changes:** • Supports new operands: *rdev*, *rdev-rdev*, SWITCH, SWCH. • Also can be entered by a class A user.
SET MODE	**Incompatible changes:** • Operands removed: MAIN and *cpuid*. • Has a response. VM/ESA 1.1.5 370 Feature did not have a response. **Upwardly compatible changes:** • Is class A or F command. In VM/ESA 1.1.5 370 Feature, was only a class F command.
SET PAGING *nn*	**VM/ESA 1.1.5 370 Feature only.**
SET PF*nn*	**Upwardly compatible changes:** • New operands supported: SUBSTITU, NODISP. • Added capability for handling mixed DBCS input mixed DBCS strings for PF key data. • For IMMED and NODISP, the length limit for the input data passed to CP when you press the PF key has changed. The limit is now the greater of the following: – 240 – The length of the input area In VM/ESA 1.1.5 370 Feature, this limit was always the length of the input area. (The length of the input area is twice the terminal line size minus 21.) **Notes:** 1. You can use the LINESIZE field in the QUERY TERMINAL response to determine the terminal line size. Remember that this needs to be the terminal where you will be pressing the PF key. 2. For DELAYED, the limit of the length of data displayed when you press the PF key remains unchanged. It is the length of the input area.
SET PF*nn* RETRIEVE	**Incompatible changes:** • In VM/ESA 2.4.0, the RETRIEVE buffer can save up to seven input lines. Each input line can be up to 139 bytes long. In VM/ESA 1.1.5 370 Feature, the RETRIEVE buffer would save more than seven input lines, however, the total number of bytes in all input data could not exceed 254 bytes. Input data includes both the input lines and length bytes used by CP. • Saves the retrieve buffer after a disconnect. VM/ESA 1.1.5 370 Feature clears the retrieve buffer after a disconnect. **Upwardly compatible changes:** • Supports a new operand: CLEAR.
SET PRIORITY	**VM/ESA 1.1.5 370 Feature only.** The SET SHARE command provides similar functions but is not compatible with the VM/ESA 1.1.5 370 Feature SET PRIORITY *userid nn*.
SET QDROP	**VM/ESA 1.1.5 370 Feature only.**
SET RESERVE	**Upwardly compatible changes:** • The operand may be spelled RESERVED.
SET SASSIST	**VM/ESA 1.1.5 370 Feature only.**

Table 36 (Page 26 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
SET SRM	**Incompatible changes:** • The following operands are not supported: APAGES, IB, MAXDRUM, MHFULL, PB, PCI, PGMLIM. (Operands still supported are DSPSLICE and MAXWSS.) • The MAXWSS operand gives a different response. **Upwardly compatible changes:** • Supports new operands: IABIAS, STORBUF, LDUBUF, DSPBUF.
SET STBYPASS	**VM/ESA 1.1.5 370 Feature only.**
SET STMULTI	**VM/ESA 1.1.5 370 Feature only.**
SET SVCACCL	**VM/ESA 1.1.5 370 Feature only.**
SET S370E	**VM/ESA 1.1.5 370 Feature only.**
SET TIMER	**Incompatible changes:** • In XA, ESA and XC virtual machines CP accepts the SET TIMER OFF command but ignores it. VM/ESA 2.4.0 does not accept SET TIMER ON or SET TIMER REAL in XA, ESA, or XC virtual machines. **Note:** SET TIMER works the same in 370 virtual machines as it did in VM/ESA 1.1.5 370 Feature.
SET VMSAVE	**VM/ESA 1.1.5 370 Feature only.**
SET 370E	**VM/ESA 1.1.5 370 Feature only.**
SHUTDOWN	**Incompatible changes:** • The POWEROFF operand is no longer supported. This operand was only used for 4361 processors. **Upwardly compatible changes:** • Supports new operands: AT *addr*, CYLINDER, EXTENT, MDISK, FNAME, MODULE, OFFSET, ORIGIN, IPLPARMS. • Displays an additional response indicating that shutdown has started.
SLEEP	**Upwardly compatible changes:** • Supports new operand: ATTN.
SMSG	**Incompatible changes:** • Supports new operands for CSE support: AT *sysname*, AT *. If you use `'AT'` as the first part of your message text you receive message HCP2970E, which indicates that the CSE name is missing or invalid. In VM/ESA 1.1.5 370 Feature, you could use `'AT'` as the first part of your message text. **Upwardly compatible changes:** • Mixed DBCS data may be used as input message text. • Message HCP003E may now be issued if an external security manager is installed and you are not authorized to issue this command.
SPACE	**Incompatible changes:** • Does not support logical printers. • Has a response.
SPMODE	**VM/ESA 1.1.5 370 Feature only.**

Table 36 (Page 27 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
SPOOL	**Incompatible changes:** • In VM/ESA 2.4.0, SPOOL RDR HOLD leaves the files in the reader queue after you read it but does not put the file in user hold status. In VM/ESA 1.1.5 370 Feature, the files were left in the reader queue after you read it but, they were placed in user hold status. If you want to place the file in user hold status in VM/ESA 2.4.0 after reading, use SPOOL RDR KEEP. • FLASH has the new ALL operand as its default; in VM/ESA 1.1.5 370 Feature, 0 was the default. • The function of EOF is different: – For a punch, EOF causes the device to be closed when 50000 records are placed into the active spool file. VM/ESA 1.1.5 370 Feature does not have this limit. – For print files, if page controls are found within 100 records following the 50000th record, the file is closed at the end of the page. If no page controls are found by then, it is presumed there are none, and the file is closed after 50100 records. • The response is different. • The default settings for printer spool files are changed: – Added default values are: NOKEEP, NOMSG, NONAME, NOEOF, DEST OFF. In VM/ESA 1.1.5 370 Feature, FLASHC 001 was also a default. • CONT scans the reader queue once from beginning to end. After each spool file is read, scanning continues with the next logical file in the queue. In VM/ESA 1.1.5 370 Feature, after each spool file was read, CONT scanned starting from the beginning of the queue. If you spool the reader with the new RESCAN option, scanning starts from the beginning of the queue. **Upwardly compatible changes:** • New operands supported: KEEP, NOKEEP, MSG, NOMSG, NAME, NONAME, NORESCAN, RESCAN. • DIST supports a new * operand, which does the same as DIST OFF. • DEST supports the new ANY operand. • FLASH, MODIFY, FCB, and CHARS support the new OFF operand. • Has a minimum truncation of NOT for NOTERM; in VM/ESA 1.1.5 370 Feature, it was NOTE. • Has new default settings for reader, punch, console spool files. – For reader spool files, NOKEEP and NORESCAN are the new default values. – For punch files, new default values are: NOKEEP, NOMSG, NONAME, NOEOF, and DEST OFF. – For console files, new default values are: NOKEEP, TO * PRINTER, NOMSG, STOP, NONAME, NOEOF, DEST OFF.

Table 36 (Page 28 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
SPTAPE	**Incompatible changes:** • DUMP and LOAD no longer support operands * and SYSTEM. • LOAD and SCAN no longer support the SADUMP operand. • LOAD no longer supports the FOR *userid* operands. The new *userid* operand of LOAD indicates that only those files owned by the user ID are to be processed. In VM/ESA 1.1.5 370 Feature, the FOR *userid* operands indicated that each file loaded from the tape is assigned to the specified user ID. • The ALL operand of DUMP and LOAD indicates all files regardless of their spoolid, class, or form; in VM/ESA 1.1.5 370 Feature, ALL indicated all files regardless of class. • When an I/O error condition occurs, the tape is positioned as specified by the user; in VM/ESA 1.1.5 370 Feature, the tape was unloaded. • Multi-volume tapes are supported. While dumping, the current tape is rewound to the point marking the end of the last completely dumped file. Then a response is issued requesting that the next tape be mounted. In VM/ESA 1.1.5 370 Feature, multi-volume tapes were not supported. Processing completed after the first tape, and partial files were purged. • VM/ESA 1.1.5 370 Feature system-type files are skipped. These are files created by the CPTRAP, MONITOR, ACNT, CPDUMP, or VMDUMP commands. Also skipped are files that contain VM/ESA 1.1.5 370 Feature CP abend dumps. A response indicates that a VM/SP system-type file is being skipped. • If the owner user ID of a file being transported to a VM/XA system does not exist on the VM/XA system, the file is assigned to the SPTAPE command issuer's user ID. Also, if the file owner runs out of spoolids, the file is assigned to the SPTAPE command issuer and a response is sent to both users. • Gives a completion response. • The response has a *dest* field inserted. **Upwardly compatible changes:** • The response for SPTAPE operation complete, cancelled, or stopped may contain information for the new devices, 3424 and 9348. • Changed operands: – DUMP and LOAD support new operands: NSS, IMG, UCR, NLS, and TRFILES. – DUMP supports other new operands: COPY, MODE XF, MODE COMP, MODE NOCOMP, MODE 3490B, MODE 3490C. – The FORM operand of DUMP and LOAD supports a new operand, *. – The CLASS operand allows you to use up to eight classes or * to indicate all classes. In VM/ESA 1.1.5 370 Feature, only one to four classes were allowed. – DUMP, LOAD, and SCAN support a new *userid* operand. • A class E user can also issue SPTAPE, but not for reader, printer, or punch files. • Responses can have new values because of new operands. • The first two digits of the 4-digit year are included in hexadecimal format in the SFBLOK dumped to tape. **Note:** If you convert a VM/ESA 1.1.5 370 Feature spool file to VM/ESA 2.4.0, it loses those file attributes that do not have VM/ESA 2.4.0 equivalents. Therefore, if the file is converted back to the VM/ESA 1.1.5 370 Feature system, it will not be identical with the original file.

Table 36 (Page 29 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
START	**Incompatible changes:** • Does not support the CFILEFCB operands. • DEST defaults to the default definition specified in the RDEVICE statement in the SYSTEM CONFIG file or in the RDEVICE macro in HCPRIO. In VM/ESA 1.1.5 370 Feature, DEST defaulted to OFF. • Has a different response. • New message: HCP1562E. **Upwardly compatible changes:** • Supports new operands: – SECLABEL *seclabel* – UR – LIMIT *limit* – SEP (as the opposite of NOSEP) – HOLD (as the opposite of PURGE) – USER *userid* or USER SYSTEM (also valid for reader devices; in VM/ESA 1.1.5 370 Feature, no operands were valid for reader devices except for ALL) – AFP or NOAFP – IMAGE *imagelib* – *name* for FCB – FOLD or NOFOLD – INDEX *nn* – OFF for FLASH. • The minimum truncation for NOSEP is NO; in VM/ESA 1.1.5 370 Feature, it was NOS. • The minimum truncation for SETUP is SE; in VM/ESA 1.1.5 370 Feature, it was SET. • The CLASS operand supports up to eight classes; in VM/ESA 1.1.5 370 Feature, up to four classes were supported. • The DEST operand is not required in front of all destination values, but only required if the destination value is four or fewer hexadecimal characters or if it matches the minimum truncation of an operand on the START command. In VM/ESA 1.1.5 370 Feature, the DEST operand was required in front of all destination values. • If an external security manager is installed: – You may receive message HCP1013E if an incorrect security label is specified with this command. – If the device being started is a printer, the security label may be displayed in the response.
STCP	**VM/ESA 1.1.5 370 Feature only.** STORE H replaces STCP. Note that STORE H defaults to prefixing and STCP does not.

Table 36 (Page 30 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
STORE	**Incompatible changes:** • L*hexloc* stores third-level storage, which is storage that appears virtual to the virtual machine, if the virtual machine is operating in DAT ON mode. In VM/ESA 1.1.5 370 Feature, you could only store second level storage. • Y*reg* does not support odd numbers. *reg* must be 0, 2, 4, or 6. In VM/ESA 1.1.5 370 Feature, if an odd number was specified, it was adjusted to the preceding even number. **Upwardly compatible changes:** • Supports new operands: – R, PRI SECO HOME ASTE*raddr*., STD*hexword*., ASN*asn*., AREG*areg*., ALET*hexword*., ALET*hexword*.AL*raddr*., N, K, U, BASE*nn*, INDEX*nn* – H, which is for a class C user to store data in first-level storage – AR*reg1* – CAW, CSW – CDX (crypto domain index) – VECTOR. • STORE STATUS also stores the prefix register at decimal address 264 and access registers at decimal address 288.
SYSTEM	**Incompatible changes:** • Does more clearing than VM/ESA 1.1.5 370 Feature. • CLEAR and RESET leave the virtual machine in a stopped state. In VM/ESA 1.1.5 370 Feature, they left the virtual machine in a disabled wait state. • RESTART causes a dump. **Upwardly compatible changes:** • Supports new operands STORE STATUS, which perform the same function as the STORE STATUS command. • Can be used to cause a break-out from a wait state caused by APPC/VM, a DIAGNOSE code X'A4', or a DIAGNOSE code X'A8'.
TAG	**Incompatible changes:** • The DEV operand does not support DE as an abbreviation. • Response may be different. **Upwardly compatible changes:** • DEV operand is not required. • Message HCP003E or HCP026E may now be issued if an external security manager is installed and you are not authorized to issue this command. • Message HCP6525E may now be issued if an external security manager is installed but unavailable to verify authorization.

Table 36 (Page 31 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
TERMINAL	**Incompatible changes:** • The SCRNSAVE operand is not supported. • Reconnecting does not affect the HILIGHT setting. In VM/ESA 1.1.5 370 Feature, HILIGHT is always set to OFF when you reconnect. • If the VTAM service machine controlling the SNA/CCS terminal does not support the TERMINAL BREAKIN GUESTCTL command then the command is not executed and you receive a message. When you reconnect, BREAKIN is reset to IMMED. If the SNA/CCS terminal does support the TERMINAL BREAKIN GUESTCTL command then it executes the same way it did in VM/ESA 1.1.5 370 Feature. In VM/ESA 1.1.5 370 Feature GUESTCTL changed the BREAKIN setting and was not reset to IMMED when you reconnected. • The NONE operand for TERMINAL BRKKEY is ignored in a line mode environment; the break key places the user in CP mode. In VM/ESA 1.1.5 370 Feature, the NONE operand caused the break key to be disabled so the user could not enter CP mode. **Upwardly compatible changes:** • BRKKEY also supports PA2 and CLEAR. • Supports the changes in sending mixed double-byte character set (DBCS) data streams from a virtual machine to its console. • Added capability for handling mixed DBCS input when TERMINAL APL/TEXT is ON for VM terminals connected to VTAM through VSCS. In previous releases, when DBCS was not supported and VTAM was connected through VSCS, you received no error message and unexpected results, such as your terminal being disconnected. With this new support, DBCS output is disabled and you receive error message HCP2104I.

Table 36 (Page 32 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
TRACE	**Note:** TRACE has a lot of changes and additions. You can find details on tracing and the TRACE command in *VM/ESA: Virtual Machine Operation* and *VM/ESA: CP Command and Utility Reference*. **Incompatible changes:** • The minimum truncation of the PROGRAM operand is PROG. In VM/ESA 1.1.5 370 Feature, it was PRO. • Operands no longer supported: CSW, OFF, SNS, PRIV. • TRACE CCW traces RIO, SIO, SIOF, SSCH, and RSCH instructions that successfully initiate I/O. In Release 5, TRACE CCW traced SIO and TIO instructions to all devices and virtual and real CCWs for nonspool nonconsole device I/O operations. You can also just trace SIO, SIOF, SSCH, RIO, and RSCH instructions by including the appropriate operand. For example, TRACE SIO CCW will trace CCWs for SIO instructions. • Responses may be different. • New message: HCP1038E. **Upwardly compatible changes:** • Supports additional operands including many from the VM/ESA 1.1.5 370 Feature PER command: – APPEND, AR*s*, CLEAR, COUNT, GOTO, CALL, RETURN, G*prs*, MCH, *mnemonic*, RIO, RSCH, SIOF, SSCH, DIAGNOSE, MC, TABLE, TPI, TRAP. – New operands common to all types of traces include: PSWA, FROM, RANGE, PRI, SECO, HOME, ASTE*raddr*, STD*hexword*, STO*raddr*, ASN*asn*, AREG*areg*, ALET*hexword*, ALET*hexword*.AL*raddr*, *range*, SKIP, PASS, STOP, STEP, NOTERMINAL, NOPRINTER, SUPERVISOR, PROBLEM, DAT, NODAT, IDENTIFIER, CMD, NOCMD. – BRANCH supports many new operands. – END supports two new operands: ALL, COUNT. ALL is the default. – INSTRUCTION supports new operands: DATA *string*. – I/O supports new operands: *range*, INSTRUCTION, CCW, NOCCW, SIM, NOSIM, INTERRUPT. I/O also accepts IO as an abbreviation. – SIO supports new operands: *range*, SIM, NOSIM, CCW, NOCCW. – SVC supports new operands: *range*, SIM, NOSIM. • Minimum truncations of some operands have changed: – TERMINAL is TE; in VM/ESA 1.1.5 370 Feature, it was TERM. – BOTH is BO; in VM/ESA 1.1.5 370 Feature, it was BOTH. – INSTRUCTION is I; in VM/ESA 1.1.5 370 Feature, it was INST. – PRINTER is P; in VM/ESA 1.1.5 370 Feature, it was PRIN. – PROGRAM also supports PRG. – I/O also supports IO.
TRANSFER	**Incompatible changes:** • Allows files created on any device to be transferred to any other device. • You cannot retrieve a file if it was created by DIAGNOSE code X'94' with the NORETURN operand. You will receive a message indicating that the file was not transferred because it was created with the NORETURN operand. • Responses are different, for example: – You can get 'WAS' rather than 'AS' – 'COPY' is 'CPY'. **Upwardly compatible changes:** • Message HCP007E may now be issued if an external security manager is installed and you are not authorized to issue this command.

Table 36 (Page 33 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
UNLOCK	**Upwardly compatible changes:** • Supports new operands: *, *symbol*, RIO370, SYMBOL, USERID, ASIT, SPACE, *firstpage*, and *lastpage*.
VARY	**Incompatible changes:** • Logical printers are not supported. • The PROCESSR operand is spelled PROCESSOR; in VM/ESA 1.1.5 370 Feature, it was PROCESSR. • The FORCE, VPHY, and VLOG operands are not supported. • A new operand, PATH, enables or disables paths. In VM/ESA 1.1.5 370 Feature, the path was enabled or disabled when the device was. **Upwardly compatible changes:** • Supports a new operand: VECTOR. • A new response is displayed if the PATH is OFFLINE because of a request initiated by a control unit.
VMDUMP	**Upwardly compatible changes:** • Supports new operands: DCSS (same as DSS), XA, CLOSE, CANCEL, CONT, NOCONT, PRI, SPACE, ALET, ASIT, AREG. • Dumps more information than in VM/ESA 1.1.5 370 Feature. • Message HCP003E may now be issued if an external security manager is installed and you are not authorized to issue this command. **Note:** The dump is formatted for Dump Viewing Facility use. In VM/ESA 1.1.5 370 Feature, the dump was formatted for IPCS use.
WARNING	**Incompatible changes:** • The response includes the time and message text on the same line; in VM/ESA 1.1.5 370 Feature, the time was on a separate line. • Supports new operands for CSE complexes: AT ALL, AT *sysname*, AT *. If you use `'AT'` as the first part of your message text you receive message HCP2970E, which indicates that the CSE name is missing or invalid. In VM/ESA 1.1.5 370 Feature, you could use `'AT'` as the first part of your message text. **Upwardly compatible changes:** • Class C users can also enter this command, however, they cannot use the ALL operand. • Added two operands for handling mixed DBCS input: ALLDBCS, ALLSBCS. DBCS text is supported for the *msgtext* parameter. • Message HCP003E may now be issued if an external security manager is installed and you are not authorized to issue this command.

Table 36 (Page 34 of 34). CP Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
XAUTOLOG	**Incompatible changes:** • For 370 virtual machines and XA or XC virtual machines, CP presents *console-input-data* to CMS at different times. (VM/ESA 1.1.5 370 Feature worked like 370 virtual machines work in the new release.) The result is that, in 370 virtual machines, *console-input-data* is presented to CMS when the first console read occurs. The console read is normally done by SYSPROF EXEC when it issues a PARSE EXTERNAL instruction, unless SYSPROF EXEC is bypassed or the AUTOCR parameter was used at IPL time. In XA or XC virtual machines, *console-input-data* is presented to CMS before the console read occurs. CMS places the *console-input-data* in the terminal input buffer in anticipation of some subsequent request to read from the console. The subsequent request to read from the console is usually the console read that occurs because of the PARSE EXTERNAL instruction in the SYSPROF EXEC. Note that, when in XA or XC virtual machines, the EXTERNAL() VM REXX built-in function can be used to test for the existence of *console-input-data* before the PARSE EXTERNAL is issued. • Users defined with the LBYONLY operand in the password field of the USER directory control statement cannot be logged on by the XAUTOLOG command if the password validation is required. **Upwardly compatible changes:** • Supports the System Console.

Converting DMKRIO and DMKSYS Macros to SYSTEM CONFIG File Statements

This section contains the following subsections:

- "Converting DMKRIO Macros to SYSTEM CONFIG Statements"
- "Converting DMKSYS Macros to SYSTEM CONFIG Statements" on page 349
- "SYSTEM CONFIG Statements for Supported Devices" on page 349
- "Converting DMKRIO/DMKSYS to HCPRIO/HCPSYS" on page 352

Note that on your VM/ESA 2.4.0 system you must have a HCPSYS file with at least a SYSEND macro coded and a HCPRIO file with at least a RIOGEN CONS=DYNAMIC. HCPRIO entries are also needed for dedicated devices of V=R guests that use V=R recovery.

Converting DMKRIO Macros to SYSTEM CONFIG Statements

Table 37 lists DMKRIO macros and the corresponding SYSTEM CONFIG statements. Macros marked "Not applicable" have no equivalent SYSTEM CONFIG statement, so no conversion is necessary.

Table 37. Converting DMKRIO Macros to SYSTEM CONFIG Statements

DMKRIO Macro	Conversion
CLUSTER	Not applicable
TERMINAL	Not applicable
RDEVICE	RDEVICE statement. See Table 39 on page 349 for a summary by device.
RCTLUNIT	Not applicable
RCHANNEL	Not applicable
RIOGEN	OPERATOR_CONSOLES statement and EMERGENCY_MESSAGE_CONSOLES statement

Converting DMKSYS Macros to SYSTEM CONFIG Statements

Table 38 lists DMKSYS macros and the corresponding SYSTEM CONFIG statements. Macros marked “Not applicable” have no equivalent SYSTEM CONFIG statement, so no conversion is necessary.

Table 38. Converting DMKSYS Macros to SYSTEM CONFIG Statements

DMKSYS Macro	Conversion
SYSOWN	CP_OWNED statement
SYSRES	CP_OWNED statement and SYSTEM_RESIDENCE statement
SYSOPR	SYSTEM_USERIDS statement
SYSCOR	No equivalent. By default, CP uses all of storage. To change the amount of storage that CP uses, specify the STORE parameter on screen displayed by the Stand-alone Program Loader (SAPL). (The screen is displayed when a LOADPARM indicating a console address is specified at IPL.)
SYSTIME	TIMEZONE_DEFINITION and TIMEZONE_BOUNDARY statements. Can be set dynamically with CP SET TIMEZONE and CP DEFINE TIMEZONE commands.
SYSMON	Not applicable
SYSJRL	JOURNALING statement
SYSACNT	SYSTEM_USERIDS statement
SYSFORM	USERFORM statement and FORM_DEFAULT statement
SYSPCLAS	PRINTER_TITLE statement
SYSIPL	No equivalent, but see the AUTO_WARM_IPL operand of the FEATURES statement.
SYSID	SYSTEM_IDENTIFIER statement
SYSORD	Not applicable
SYSFCN	PRIV_CLASSES statement
SYSMIH	HOT_IO_RATE statement
SYSLOCS	Not applicable

SYSTEM CONFIG Statements for Supported Devices

Table 39 lists the devices that are also supported by VM/ESA 2.4.0. For each supported device, the table lists the SYSTEM CONFIG statement that is needed, if any.

Table 39 (Page 1 of 4). SYSTEM CONFIG Statements Needed for Supported Devices

Device	SYSTEM CONFIG File Statement
DASD Devices Supported by VM/ESA 2.4.0	
3350 Model A2, A2F, B2, B2F, C2, C2F	This device is not supported by the RDEVICE statement in SYSTEM CONFIG. It must be defined by an RDEVICE macro in the HCPRIO ASSEMBLE file.
3370 Model A1, A2, B1, B2	See note 1 on page 352.
3375 Model A1, B1, D1	See note 1 on page 352.
3380 Model AD4, AE4, AJ4, AK4, BD4, BE4, BJ4, BK4	See note 1 on page 352.

Table 39 (Page 2 of 4). SYSTEM CONFIG Statements Needed for Supported Devices

Device	SYSTEM CONFIG File Statement
3380 Model CJ2	See note 1 on page 352.
3390 Model A14, A18, A28, B14, B18, B24, B28, B1C, B2C	See note 1 on page 352.
3390 Model A34, A38, B34, B38, B3C	See note 1 on page 352.
3995 Optical Library Dataserver	No statement required.
9332 Model 400, 402, 600, 602	See note 1 on page 352.
9335 B01	See note 1 on page 352.
9336 Model 10, 20	See note 1 on page 352.
9345 Model B12, B22	See note 1 on page 352.
Tape Devices Supported by VM/ESA 2.4.0	
2440	RDEVICE statement needed with 3420 specified for TYPE.
3420 Model 3-8	RDEVICE statement needed with 3420 specified for TYPE.
3422	RDEVICE statement needed with 3422 specified for TYPE.
3430	No statement required.
3480 - All Models	No statement required.
3490 - All Models	No statement required.
3495 - All Models	No statement required (looks like a 3490 to VM).
3590 - All Models	No statement required.
9348 Model 11, 12	No statement required.
Printer Devices Supported by VM/ESA 2.4.0 (See note 2 on page 352)	
3203 Model 5	RDEVICE statement needed with 3203 specified for TYPE.
3211 Model 1 & 5	RDEVICE statement needed with 3211 specified for TYPE. (See note 3 on page 352.)
3262	No statement required.
3268 Model 2 & 2C	No statement required.
3287 Model 1, 1C, 2, 2C, 4	No statement required.
3800 Model 1	No statement required.
3800 Model 3, 6	RDEVICE statement needed only when you want to use it for Advanced Function Printing™. For this case, specify AFP for TYPE.
3800 Model 8	If you do not specify an RDEVICE statement with AFP for TYPE, the printer will be brought online but it will not be used by the spooling subsystem.
3812	Depends if Coax, BSC or SDLC attached
3816	Depends if Coax, BSC or SDLC attached
3820	If channel attached, RDEVICE statement needed with AFP specified for TYPE.
3825	RDEVICE statement needed with AFP specified for TYPE.
3827	RDEVICE statement needed with AFP specified for TYPE.

Table 39 (Page 3 of 4). SYSTEM CONFIG Statements Needed for Supported Devices

Device	SYSTEM CONFIG File Statement
3835	RDEVICE statement needed with AFP specified for TYPE.
4245 Model 1, 12, 20	No statement required.
4248 Model 1, 2	No statement required.
6262 Model 14, 22	No statement required.
Punch/Reader Devices Supported by VM/ESA 2.4.0	
3505 Model B1, B2	RDEVICE statement needed with READER specified for TYPE.
3525 Model P1, P2, P3	RDEVICE statement needed with PUNCH specified for TYPE.
Display Devices Supported by VM/ESA 2.4.0	
3277	RDEVICE statement needed with 3277 specified for TYPE.
All Others	See note 4 on page 352.
Communication Controllers Supported by VM/ESA 2.4.0	
3705	When running an emulator control program (EP), an RDEVICE statement is needed with 3705 specified for TYPE.
3725	When running an emulator control program (EP), an RDEVICE statement is needed with 3705 specified for TYPE.
3720	No statement required when running the network control program (NCP).
3725	No statement required when running the network control program (NCP).
3745	No statement required when running the network control program (NCP).
3725	Statement required when running the partition emulation program (PEP) for BSC or ASCII lines.
3745	Statement required when running the partition emulation program (PEP) for BSC or ASCII lines.
9221 ICA	No statement required.
9221 Integrated Token Ring	RDEVICE statement needed with ICA_TOKENRING specified for TYPE.
9221 Integrated Ethernet	RDEVICE statement needed with ICA_ETHERNET specified for TYPE.
ESCON Devices Supported by VM/ESA 2.4.0	
3172 Model	No statement required.
3174 Model 12L	No statement required.
3174 Model 22L	No statement required.
3490 Tape Subsystem	No statement required.
9032 Escon Director Model 2	RDEVICE statement needed with SWITCH specified for TYPE.
9033 Escon Director Model 1	RDEVICE statement needed with SWITCH specified for TYPE.
9034 Escon Converter Model 1	No statement required.
9035 Escon Converter Model 2	No statement required.

Table 39 (Page 4 of 4). SYSTEM CONFIG Statements Needed for Supported Devices

Device	SYSTEM CONFIG File Statement
Miscellaneous Devices Supported by VM/ESA 2.4.0	
3088	No statement required.
3423 Juke Box	RDEVICE statement needed with UNSUPPORTED specified for TYPE.
3737 Remote CTCA	RDEVICE statement needed with CTCA specified for TYPE.
3890™ Check Sorter	No statement required.
4753 Network Security Processor	No statement required.
7171	RDEVICE statement needed with 3270_DISPLAY specified for TYPE. The operand EMULATED_3270 YES should also be specified.
8232	No statement required.
CTCA	RDEVICE statement needed with CTCA specified for TYPE.

Notes:

1. No SYSTEM CONFIG file statement needed if the DASD are not shared. If they are shared, the following statement can be used:

   ```
   rdevice rdev type dasd shared yes
   ```

2. If you want to specify spooling characteristics for a printer when you IPL your system, you must include an RDEVICE statement for that printer in the SYSTEM CONFIG file.
3. Support in Emulation on other printers.
4. All 3270s other than 3277s can be sensed and no SYSTEM CONFIG file statement is needed unless the device has special features. For example, the following RDEVICE statement would be used if the device had an Operator Identification Card feature:

   ```
   rdevice rdev type 3270_display oper_ident_reader yes
   ```

5. Except for the supported IBM 3350 models, all devices are supported in both HCPRIO macros and in SYSTEM CONFIG statements. RDEVICE macros in HCPRIO must be coded for IBM 3350 Models.

Converting DMKRIO/DMKSYS to HCPRIO/HCPSYS

If you choose to convert your DMKRIO and DMKSYS files to HCPRIO and HCPSYS, see the *VM/ESA: Planning and Administration*. That book describes the macros used in HCPRIO and HCPSYS.

You can also run the VM/ESA MIGR tool against your DMKRIO and DMKSYS files. This tool gives you information on how to change your 370 Feature system files to HCPRIO and HCPSYS. See *VM/ESA: REXX/EXEC Migration Tool for VM/ESA* for details on this tool.

User Directory Control Statements

Table 40 lists the user directory control statements that have changed since VM/ESA 1.1.5 370 Feature. See the *VM/ESA: Planning and Administration* book for complete descriptions of user directory control statements.

Note: You can use the VMUDQ macro to query the CP user directory. See the *VM/ESA: CP Programming Services* book for more information on this macro.

Table 40 (Page 1 of 5). User Directory Control Statements Changed since VM/ESA 1.1.5 370 Feature

Statement	Changes
ACCOUNT	**Upwardly compatible changes:** • ACCOUNT can follow the new device statements DASDOPT, SPECIAL, and MINIOPT, as well as the USER user directory control statement. In VM/ESA 1.1.5 370 Feature, ACCOUNT had to follow the USER user directory control statement. • Supports new parameters to define up to eight alternative account numbers: *acctnum2...acctnum8*.
AUTOLOG	**New** Designates user IDs that can enter the XAUTOLOG command. The AUTOLOG control statement is synonymous with the XAUTOLOG control statement.
CONSOLE	**Incompatible changes:** • VM/ESA 1.1.5 370 Feature device types not supported are: 1052, 3210.
CPU	**New** Specifies a virtual processor that is to be defined automatically for the virtual machine at LOGON.
CRYPTO	**New** Authorizes the user to define virtual cryptographic facilities and specifies the domains and CAM queues the virtual machine is allowed to use.
DASDOPT	**New** Provides an extension to the DEDICATE, LINK, and MDISK statements. It supports the NOWRKALLEG option.
DATEFORMAT	**New** Specifies a user's default date format for commands that provide multiple date formats.
DEDICATE	**Incompatible changes:** • NETWORK, *resource*, and 3330V parameters are not supported. **Upwardly compatible changes:** • Supports new parameters: NOASSIGN and NOIOASSIST. If a parameter is specified, it must be in the proper position. **Note:** You can use the NOIOASSIST parameter for devices that normally use I/O assist but do not benefit by it, such as 5080 devices.
DIRECTORY	**Incompatible changes:** • Not all VM/ESA 1.1.5 370 Feature devices are supported by IBM. However, the *devtype* parameter supports all previously supported device types. See the *VM/ESA: General Information* book for information about device support. **Upwardly compatible changes:** • New parameters: *nnnnnn-xxxx*, and *sysafnid*. • Supports new values for the *devtype* parameter: 3370, 3390, 9332, 9335, 9336, 9345.
GLOBALDEFS	**New** Signifies the beginning of the global definition section.
GLOBALOPTS	**New** Used to define global settings to be used while processing user definitions.
IUCV	**Upwardly compatible changes:** • New values for the *cpsyserv* parameter: *ACCOUNT, *BLOCKIO, *MONITOR, *MSG, *MSGALL, *RPI, *SYMPTOM. Also, if your virtual machine is running the Enterprise Systems Connection Manager (ESCM) licensed program, you can specify *CONFIG.

Table 40 (Page 2 of 5). User Directory Control Statements Changed since VM/ESA 1.1.5 370 Feature

Statement	Changes
LINK	**Incompatible changes:** • The *mode* parameters have different meanings depending on whether the new stable or exclusive link modes have been accessed by other people who link to the minidisk. **R** Read access is established unless another user holds a write or an exclusive mode access to the disk. In VM/ESA 1.1.5 370 Feature, read access was established unless another user held a write mode access to the disk. **RR** Read access is established unless another user holds an exclusive access. In VM/ESA 1.1.5 370 Feature, read access was granted no matter what. **W** No change. **WR** If write access is denied, read access is established unless another user holds an exclusive mode access to the disk. In VM/ESA 1.1.5 370 Feature, if write access was denied, read access was always established. **M** Write access is established unless another user holds a write, a stable, or an exclusive mode access to the disk. In VM/ESA 1.1.5 370 Feature, write access was established unless another user held a write access. **MR** Write access is established unless another user has write, stable, or exclusive mode access to the disk. If a write or stable mode access exists, read-only access is established. If an exclusive mode access exists, read access and write access are denied. In VM/ESA 1.1.5 370 Feature, write access was established unless another user has write mode access to the disk. If a write mode access existed, read-only access was established. **MW** Write access is established unless another user holds a stable or exclusive mode access to the disk. In VM/ESA 1.1.5 370 Feature, write access was always established. **Upwardly compatible changes:** • Supports a new *suffix* to the *mode* parameter. The *suffix* can be S or E. The S-suffix authorizes the virtual machine to use the LINK command's stable access modes SR, SW, and SM. The E-suffix allows the virtual machine to use exclusive and stable access modes, SR, SW, SM, ER, and EW.
LOAD	**Upwardly compatible changes:** • New file type, POOLPART. • Supports a new G operand for global definitions.
MACHINE	**New** Specifies the architecture the virtual machine will simulate, as 370, XA, ESA, or XC. The default, if the MACHINE statement is omitted, is 370.
MDISK	**Incompatible changes:** • Not all VM/ESA 1.1.5 370 Feature devices are supported by IBM. However, the *devtype* parameter supports all previously supported device types. See the *VM/ESA: General Information* book for information on device support. **Upwardly compatible changes:** • Two new optional mode suffix letters are supported, S and E, to authorize exclusive or stable mode for the minidisk. • Supports new operands: END, ALL, DEVNO *rdevno*. • Supports new values for the *devtype* parameter: 3370, 3990, 9332, 9335, 9336, 9345. • Supports new operands: V-DISK.

Table 40 (Page 3 of 5). User Directory Control Statements Changed since VM/ESA 1.1.5 370 Feature

Statement	Changes
MINIOPT	**New** Provides an extension to MDISK which is used when defining non-full-pack minidisks.
NAMESAVE	**New** Authorizes a virtual machine to access a restricted named saved system or saved segment.
NOPDATA	**New** Authorizes a virtual machine to use NOP CCWs to transfer data to CP spool files.
OPTION	**Incompatible changes:** • The following parameters are not supported: AFFINITY, BMX, ECMODE, ISAM, REALTIMER, STFIRST, SVCOFF, VCUNOSHER, VMSAVE, and 370E. • The minimum truncation for the missing interrupt operand is now MIH. In VM/ESA 1.1.5 370 Feature, it was MI. **Upwardly compatible changes:** • Supports new parameters: LKFAC, APPLMON, DEDICATE, NODEDICATE, NOVF, VIRT=FIXED, D84NOPAS, LNKNOPAS, LINKMSG, MAINTCCW, NOMDCFS, QUICKDSP, SETORIG, TODENABLE, RMCHINFO, LNKSTABL, LNKEXCLU, CFVM, CFUSER, DIAG88.
POSIXGLIST	**New** Specifies all POSIX groups of which the user is a member.
POSIXGROUP	**New** Defines a POSIX group.
POSIXINFO	**New** Specifies a user's POSIX information.
POSIXOPT	**New** Specifies option settings related to a user's POSIX capabilities.
PROFILE	**Upwardly compatible changes:** • You no longer are required to have at least one valid directory control statement in a profile.
SHARE	**New** Specifies a virtual machine's share of CPU power.
SPECIAL	**Incompatible changes:** • Not all VM/ESA 1.1.5 370 Feature device types are supported: 3138, 3148, 3158, TIMER. **Upwardly compatible changes:** • Supports new operand: MSGPROC.
SPOOL	**Incompatible changes:** • The following *devtypes* are no longer supported: 1443, 3289. **Upwardly compatible changes:** • Supports new *devtypes*: READER, RDR, PRINTER, PRT, PUNCH, PCH.
SPOOLFILE	**New** Describes virtual machine spool file characteristics. The MAXSPOOL parameter lets you specify the maximum number of spool files allowed for a virtual machine.
STDEVOPT	**New** Specifies the optional storage device management functions available to a virtual machine; supported for guest virtual machines only.

Table 40 (Page 4 of 5). User Directory Control Statements Changed since VM/ESA 1.1.5 370 Feature

Statement	Changes
SYSAFFIN	**New** Defines how and to which systems of a multisystem complex the subsequent statements apply.
USER	**Incompatible changes:** • User IDs reserved for CP use are LOGN*xxxx*, LOGL*xxxx*, LOGV*xxxx*, LOGNSYSC, SYSC, and SYSTEM. In VM/ESA 1.1.5 370 Feature, reserved user IDs were LOGON*xxx*, LOGOL*xxx*, LOGN*xxxx* and SYSTEM. **Note:** Be careful about assigning any system keywords as user IDs. This can cause unpredictable results. You receive an error if you try to bring up a directory that uses SYSTEM as the user ID on a USER directory control statement. In previous releases, you did not receive an error, but might have encountered unpredictable results. • NOPASS is a new restricted password. • For V=V guests, in VM/ESA 2.4.0 if the storage size you specify is not a multiple of 1MB and is not greater than 16MB, it is rounded to the next 64KB boundary. If the storage size you specify is not a multiple of 1MB and is greater than 16MB, it is rounded up to the next 1MB boundary. For V=F or V=R guests, in VM/ESA 2.4.0 the storage size is always rounded up to the nearest multiple of the minimum storage size for preferred guests. This storage size is at least 1MB. Use the QUERY V=R command to determine the preferred guests' minimum storage size. In VM/ESA 1.1.5 370 Feature, if the storage size you specified was not a multiple of 4KB, it was rounded up to the next 4KB boundary. • Storage sizes below 16MB are rounded to 64KB boundaries. In VM/ESA 1.1.5 370 Feature, they were rounded to 4KB boundaries. • You can define up to 32 classes if they fit on the line. (If you want to define even more, you can use the CLASS statement.) In VM/ESA 1.1.5 370 Feature, you can define only up to eight classes. • LBYONLY is a new operand that can be specified in the password field of the USER statement. Any user defined with the LBYONLY operand may be restricted from performing operations that require password validation (including LOGON, AUTOLOG, XAUTOLOG, and DIAGNOSE X'84'). • Letters A-Z, numbers 0-9, and bytes `X'OE'` (shift out) and `X'0F'` (shift in) cannot be defined as logical line edit symbols (line-end, line-delete, character-delete, and escape). **Upwardly compatible changes:** • The storage specified for a virtual machine can be as large as 2047MB (2096128KB). In VM/ESA 1.1.5 370 Feature, the maximum storage size was 16MB. The minimum storage size is 64KB. In VM/ESA 1.1.5 370 Feature, the minimum was 8KB. • The *pri* option is accepted for compatibility with VM/ESA 1.1.5 370 Feature, but serves no function in VM/ESA 2.4.0. • You are no longer required to have SPOOL or CONSOLE in a USER definition entry.
XAUTOLOG	**New** Designates user IDs that can enter the XAUTOLOG command.
XCONFIG	**New** Specifies control parameters for the extended-configuration facilities provided in an XC virtual machine: access lists and address spaces.

Table 40 (Page 5 of 5). User Directory Control Statements Changed since VM/ESA 1.1.5 370 Feature

Statement	Changes
XSTORE	**New** You use the XSTORE user directory control statement to dedicate some or all of Expanded Storage to a virtual machine. In VM/ESA 2.4.0, Expanded Storage is used by CP and by virtual machines.

CP Utilities

Table 41 lists the utility programs that have changed since VM/ESA 1.1.5 370 Feature. See the *VM/ESA: CP Command and Utility Reference* for details on most utility programs.

Table 41 (Page 1 of 3). Utility Programs Changed since VM/ESA 1.1.5 370 Feature

Utility	Changes
DASD Dump Restore	**Incompatible changes:** • The END OF VOLUME CYL response has eight digits for the number of cylinders. In VM/ESA 1.1.5 370 Feature, it had four digits. • DDR can no longer use 8809 tape data streaming. • Messages now have a prefix of HCP instead of DMK. The following messages are new: 701I, 8100I, 8101E, 2236E, 2238E, 2239E, 2240E, 9014W. **INPUT/OUTPUT control statement changes:** • Some device types are no longer supported by IBM. See the *VM/ESA: General Information* book for information on device support. • The SCRATCH operand is no longer supported. **SYSPRINT control statement changes:** • If the SYSPRINT control statement is issued with anything other than 00E or CONS or blank, the statement is ignored and you receive the message: `SYSPRINT NOT 00E OR CONS--IGNORED`. In VM/ESA 1.1.5 370 Feature, this was not a restriction and no message was issued. **Function statement changes:** • CPVOL no longer specifies the override disk space be copied, dumped, or restored to CKD devices. It only specifies that directory or permanent disk space be copied, dumped, or restored.

Table 41 (Page 2 of 3). Utility Programs Changed since VM/ESA 1.1.5 370 Feature

Utility	Changes
DASD Dump Restore - continued	**Upwardly compatible changes:** • For input, up-to-eight-digit cylinder values are supported. In VM/ESA 1.1.5 370 Feature, up-to-four digit cylinder values were supported. • DDR can run in a 370, XA, ESA, or XC virtual machine. In VM/ESA 1.1.5 370 Feature, DDR could run only in a 370 virtual machine. • You can now invoke DDR as a stand-alone program using DDRXA. • DDR operations involving the CMS system disk may result in a message indicating the disk could not be released. During the time the disk cannot be released from read mode it is not accessed in write mode. In the case of copy or restore, the operation should complete normally. **INPUT/OUTPUT control statement changes:** • Supports new values for the *type* parameter: 3370, 3390, DASD, TAPE, 3490, 9332, 9335, 9336, 9345, 9348. • Supports additional values for MODE: XF, COMP, NOCOMP, 3490B, 3490C. • You can specify MODE 800 for 3430 and 3422 tape devices; in VM/ESA 1.1.5 370 Feature, you could not. • New options: EMSG, ESKIP.
Device support facilities	**Incompatible changes:** • VM/ESA 2.4.0 supports the device support facilities program running stand-alone or under CMS. To run it, users must have the DEVMAINT or MAINTCCW parameter in their OPTION user directory control statement.
DIRECT	**Incompatible changes:** • There is no stand-alone directory program in VM/ESA 2.4.0. As a result, you cannot recover the directory without a running system. You must bring up the VM/ESA 2.4.0 system with the NODIRECT option and proceed to recover the directory. • You use the DIRECTXA command rather than the DIRECT command to bring the directory online in VM/ESA 2.4.0. The externals are generally the same as the DIRECT command except for the changes listed here. The format of the primary response from DIRECTXA includes the version as well as the release level. **Upwardly compatible changes:** • Supports new paramters: MIXMSG, NOMIXMSG, SYSAFINID *sysafinid*. • Supports new return codes: 9, 333.
Format/allocate	**Incompatible changes:** • There is no stand-alone format/allocate program in VM/ESA 2.4.0. You can, however, use the CPFMTXA utility or the Device Support Facilities program (ICKDSF), Release 11 or later for CKD or ECKD and Release 14 or later for FBA.

Table 41 (Page 3 of 3). Utility Programs Changed since VM/ESA 1.1.5 370 Feature

<table>
<tr><th>Utility</th><th>Changes</th></tr>
<tr><td>Stand-alone dump</td><td>
See the <i>VM/ESA: Planning and Administration</i> book and the <i>VM/ESA: Virtual Machine Operation</i> book for details on the stand-alone dump utility.

<b>Incompatible changes:</b>
<ul>
<li>To invoke the stand-alone dump utility, use the HCPSADMP exec. There are no parameters for the HCPSADMP exec. In VM/ESA 1.1.5 370 Feature, you used the SADUMP exec. The following parameter was supported on the SADUMP exec: <i>filename</i> (of the ASSEMBLE file that had the SAD macro).</li>
<li>You need the following files on an accessed disk:
<table>
<tr><th>VM/ESA 2.4.0 files</th><th>VM/ESA 1.1.5 370 Feature files</th></tr>
<tr><td>HCPGPI MACLIB</td><td>DMKSAD TEXT</td></tr>
<tr><td>HCPPSI MACLIB</td><td></td></tr>
<tr><td>HCPOM1 MACLIB</td><td></td></tr>
<tr><td>HCPOM2 MACLIB</td><td></td></tr>
<tr><td>HCPVM CNTRL (the control file)</td><td>DMKVM CNTRL</td></tr>
<tr><td>HCPLDR MODULE</td><td>DMKLD00E LOADER</td></tr>
<tr><td>HCPSADLD EXEC (the load list)</td><td></td></tr>
<tr><td>HCPDR MODULE</td><td></td></tr>
<tr><td>LDT HCPSADWT</td><td>LDT DMKSADWT</td></tr>
<tr><td>$VMFPAT$ EXEC</td><td></td></tr>
<tr><td>$VMFMSG$ EXEC</td><td></td></tr>
<tr><td>HCPSADMP EXEC</td><td></td></tr>
<tr><td>HCPSDCMP MACRO</td><td></td></tr>
<tr><td>HCPSAD TEXT</td><td></td></tr>
</table></li>
<li>The default configuration name is HCPSDC. In VM/ESA 1.1.5 370 Feature, the default name was SADGEN.</li>
<li>The default control file is HCPVM. The default in VM/ESA 1.1.5 370 Feature was DMKVM.</li>
<li>HCPSADMP creates the HCPSDC file and writes the stand-alone dump utility on a real device. In VM/ESA 1.1.5 370 Feature, SADUMP created the SADGEN file and wrote the stand-alone dump program to the user's virtual card reader.</li>
<li>Some devices are no longer supported by IBM. See <i>VM/ESA: General Information</i> for information on unsupported devices.</li>
<li>Initially the stand-alone dump writes the first six pages of storage to the IPL device. In VM/ESA 1.1.5 370 Feature, it dumped the first nine pages.</li>
<li>The prompts and responses when configuring the stand-alone dump utility have changed. See the <i>VM/ESA: Planning and Administration</i> book for details.</li>
<li>You must use the Dump Viewing Facility to process the stand-alone dump. In VM/ESA 1.1.5 370 Feature, you used IPCS.</li>
<li>A wait state of 8200 indicates successful completion. In VM/ESA 1.1.5 370 Feature, a wait state of 912 indicated successful completion.</li>
<li>You probably need two tapes to hold the dump. In VM/ESA 1.1.5 370 Feature, you used only one tape.</li>
</ul>
<b>Upwardly compatible changes:</b>
<ul>
<li>You can dump all real storage. In VM/ESA 1.1.5 370 Feature, you could dump up to 64MB.</li>
<li>You can IPL the stand-alone dump program from and send dumps to new devices types: 3800, 3390, 9345, 3490.</li>
</ul>
</td></tr>
</table>

CP DIAGNOSE Codes

You can use the first four bits of the two-byte *code* field of the DIAGNOSE instruction to refer to a base register. The contents of the base register plus the last twelve bits of the *code* field equal the DIAGNOSE code. However, if the first four bits of the *code* field are zeros, then the last twelve bits alone determine the DIAGNOSE code. To remain compatible with VM/ESA 1.1.5 370 Feature, keep the first four bits of the code as zeros.

Table 42 lists the DIAGNOSE codes that have changed since VM/ESA 1.1.5 370 Feature. See the *VM/ESA: CP Programming Services* book for complete information on the DIAGNOSE codes discussed in this section.

DIAG Macro

You can now use the DIAG macro to automatically generate the correct machine format of the DIAGNOSE instruction. The format of the DIAG macro is:

```
[label] DIAG Rx,Ry,code
```

DIAGNOSE Codes and Virtual Machine Modes

With the following exceptions, all DIAGNOSE codes run in 370, XA, ESA, and XC virtual machines.

X'18'	370 virtual machines only
X'20'	370 virtual machines only
X'48'	370, XA, and ESA virtual machines only
X'248'	370, XA, and ESA virtual machines only.

Table 42 (Page 1 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

<table>
<tr><th>Code</th><th>Changes</th></tr>
<tr><td>X'00'</td><td>Storage extended identification code

Incompatible changes:

• Information at the address returned by Rx is different:

<table>
<tr><th>Field Name</th><th>Description</th></tr>
<tr><td>Environment</td><td>A 2-byte field that identifies whether VM/ESA is running in a logical partition (LPAR mode). In VM/ESA 1.1.5 370 Feature, this field was reserved and 3 bytes in length.</td></tr>
<tr><td>Version information</td><td>A new 1-byte field (formerly the third byte of the Environment field) that identifies the version number of the product (for VM/ESA 2.1.0 and later).</td></tr>
<tr><td>MCEL field</td><td>Contains zero in a 370 virtual machine and is reserved in an XA, ESA, or XC virtual machine. In VM/ESA 1.1.5 370 Feature, the MCEL field contained the maximum length of the MCEL area.</td></tr>
<tr><td>Program product bit map</td><td>If the high-order bit is off, the environment is VM/XA SP or VM/ESA (ESA Feature). If the high-order bit is on, the environment is VM/SP, VM/SP HPO, or VM/ESA (370 Feature).</td></tr>
</table>

Upwardly compatible changes:

• Supports a new entry value: access register Ax.
• The value in the program product bit map has changed to indicate the new release level. Also, Bit 13 (X'0004000000000000') indicates whether Year 2000 support is present in CP.
• There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.</td></tr>
<tr><td>X'04'</td><td>Examine real storage

Incompatible changes:

• You get an addressing exception if the first-level address that you want to examine is:
– Offline
– On a directory page
– Outside the real machine storage.

In VM/ESA 1.1.5 370 Feature, you got only an addressing exception when trying to examine storage outside the real machine storage.
• You no longer receive a specification exception if the requested real page frame is disabled.

Upwardly compatible changes:

• Supports new entry values: access registers Ax, Ay+1
• There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.</td></tr>
</table>

Table 42 (Page 2 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'08'	Virtual console function **Incompatible changes:** • When messages are returned from a double-byte character set (DBCS) message repository, new line characters are not inserted. In VM/ESA 1.1.5 370 Feature, new line characters were inserted into messages based on the physical line size of the screen. **Upwardly compatible changes:** • For entry values or Ry: – If you set the high-order byte to X'80', CP rejects a password entered on the same line as the AUTOLOG, XAUTOLOG, or LINK commands. In VM/ESA 1.1.5 370 Feature, this condition only applied to the LINK command. – If you set the high-order byte to X'20', CP returns a request to the virtual machine for a password for the XAUTOLOG, AUTOLOG, and LINK commands. In VM/ESA 1.1.5 370 Feature, this condition applied only to the AUTOLOG and LINK commands. – If you issue asynchronous commands, such as XAUTOLOG, the response is not returned in the user's buffer. • The SET D8ONECMD command can be used to control whether CP will accept multiple commands separated by X'15''s. The default setting (OFF) allows multiple commands, as in past releases. The return code from a CP command is 6890 when the virtual machine's D8ONECMD setting is FAIL. • Supports new entry values: access registers Ax, Ax+1. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.
X'00C'	Pseudo-Timer **Upwardly compatible changes:** • The data area can now wrap storage. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.
X'10'	Release pages **Upwardly compatible changes:** • Supports new entry values: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details. **Note:** If a virtual machine attempts to release a shared page within its defined virtual storage, the page is not released and no further action is taken.

Table 42 (Page 3 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'14'	Input spool file manipulation **Incompatible changes:** • Contents of the spool file blocks, SFBLOK COPY and SPLINK COPY, may be different. For information on these files, see the *VM/ESA: CP Programming Services* book. • For subcodes X'0004' and X'0008', the format definition for a VM SFBLOK can be found in the HCPGPI macro library. In VM/ESA 1.1.5 370 Feature, the definition was in DMKGPI. • For subcode X'0000': – If there is no active file when you issue this subcode, CC=2 is returned and the first data block of the first file in your virtual reader is returned to your buffer. In VM/ESA 1.1.5 370 Feature, only condition code 2 was returned. – Issuing subcode X'0000' against locked pages does not unlock them. In VM/ESA 1.1.5 370 Feature, the locked pages were unlocked. • For subcode X'001C', condition code 2 is always returned because monitor spool files are not supported in VM/ESA. In VM/ESA 1.1.5 370 Feature, monitor spool files were supported. • For subcodes X'0FFE' and X'0FFF', Rx is a 252-byte buffer address. In VM/ESA 1.1.5 370 Feature, Rx is a 332-byte buffer. • For subcodes X'0FFE' and X'0FFF', you no longer get CC=3 with return code 24. In VM/ESA 1.1.5 370 Feature, this indicated a system problem because of performance or errors in the spooling area. **Upwardly compatible changes:** • Added subfunction: X'0028' (position a spool file to the designated record). • For subcodes X'0010' and X'0020', if the count of active files is greater than 255 then the count is set to 255. In VM/ESA 1.1.5 370 Feature, there was a maximum of 255 for active files; you could not have more then 255 active files. • This DIAGNOSE code will not execute if an external security manager is installed and the user does not have proper authorization. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details. • For subcodes X'0004', X'0008', X'0FFE', and X'0FFF', a one-byte century indicator was added to the SFBLOK data area.

Table 42 (Page 4 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'18'	Standard DASD I/O **Incompatible changes:** • The channel address word (CAW) specified in Ry must be specified as follows: **Bit(s)** **Contents** 0-3 The protection key used in accessing second-level storage for the I/O operation. 4 Must be zero (suspend control). 5-7 Must be zero (unassigned). 8-31 The channel program address. 29-31 Must be zero, designating a doubleword-aligned channel program. • You can receive CC=2 and return code of X'05' for two additional reasons: – Bit 4 (suspend control) is 1 – Bits 5-7 are not B'0000'. • You can receive CC=2 and return code of X'0B' because the value in register 15 was less than the number of READs and WRITEs in the channel program. You will no longer get this return code if register 15 was not large enough for the CCW string. • You receive CC=2 and return code X'07' for different CCW problems. • The CCW chain specified by the guest must specify I/O on only one cylinder and must be of the standard format. • CP provides error recovery for this DIAGNOSE instruction. The instruction does not complete until the I/O operation has completed, including any error recovery processing. • For the 3330, 3340, 3350, 3375, and 3380 devices, the SET SECTOR CCW should precede each SEARCH CCW. **Note:** IBM no longer supports 3330s and 3340s. See the *VM/ESA: General Information* book for information about device support. • The CCW chain ends with a read or write without command chaining. • The CCW chain must not have any indirect data address (IDA), data chain (CD), or program controlled interrupt (PCI) flags. • Data crossing 4KB boundaries is handled using indirect data lists. • CCWs must be format 0. • The first CCW must be a SEEK CCW. • Cannot be used to read or write to 800-byte records; 800-byte records are not supported. **Upwardly compatible changes:** • DIAGNOSE code X'18' can be used only in a 370 virtual machine. DIAGNOSE code X'A4' can be used in a 370, XA, ESA, or XC virtual machine.
X'1C'	Clear error recording cylinders **VM/ESA 1.1.5 370 Feature only.**

Table 42 (Page 5 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'20'	General I/O **Incompatible changes:** • The channel address word (CAW) specified in Ry must be specified as follows: **Bit(s)** **Contents** 0-3 The protection key used in accessing second-level storage for the I/O operation. 4 Must be zero (suspend control). 5-7 Must be zero (unassigned). 8-31 The channel program address. 29-31 Must be zero, designating a doubleword-aligned channel program. • The CSW is stored for CC=2 and RC=X'02' or RC=X'03'. • If the user specifies an unsupported device and receives CC=3 and RC=X'0D', the CSW and the user's Ry register is set to zero. **Upwardly compatible changes:** • DIAGNOSE code X'20' can be used only in a 370 virtual machine. You can use DIAGNOSE code X'A8' in a 370, XA, ESA, or XC virtual machine.
X'24'	Device type and features **Incompatible changes:** • Contents of the real device feature, real device type class, real device type, virtual device flags, virtual device status, virtual device type class, and virtual device type fields may have changed. The *VM/ESA: CP Programming Services* book contains details on data areas. • The device information returned in Rx, Ry, and Ry+1 has changed. See the *VM/ESA: CP Programming Services* book for details. • The symbolic names returned in Rx, Ry, and Ry+1 have changed. **Upwardly compatible changes:** • This DIAGNOSE code is not upgraded to support *most* new VM/ESA 2.4.0 devices. For example, you get a X'01', device unsupported, if you try to use DIAGNOSE code X'24' with 3390, 3490, and 9345 devices. You should use DIAGNOSE code X'210' instead. However, DIAGNOSE code X'24' has been upgraded to return information on the new 3424 and 9348 devices. • Output information can contain additional information.

Table 42 (Page 6 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'28'	Dynamic channel program modification **Incompatible changes:** • The address in Rx must be absolute. In VM/ESA 1.1.5 370 Feature, it did not have to be absolute. • New return code: X'0C'. • Split return code: X'09' is split into return codes X'09' and X'0C'. • Removed return codes: X'11', X'0B', X'0F'. **Upwardly compatible changes:** • DIAGNOSE code X'28' can now be used to modify fields of a NOP CCW. This is new function for VM/ESA 1.1.5 370 Feature. • Ry can contain the subchannel number for an XA, ESA, or XC virtual machine. The contents of Ry have not changed for 370 virtual machines. • A request to modify a TIC/NOP CCW in a currently executing channel program with a CCW containing a suspend bit equal to 1, or on, is honored; the suspend bit is reset in the modified channel program.
X'2C'	DASD start of LOGREC **VM/ESA 1.1.5 370 Feature only.**
X'30'	Read page of LOGREC **VM/ESA 1.1.5 370 Feature only.**
X'34'	Read system dump spool file **Incompatible changes:** • New return code: X'14'. • The address in Rx cannot cross a page boundary. In VM/ESA 1.1.5 370 Feature, it could. • A VM/ESA 2.4.0 dump file has a different format, so you cannot use this DIAGNOSE code to read a VM/ESA 1.1.5 370 Feature dump file. • Issuing DIAGNOSE code X'34' against locked pages does not unlock them. In VM/ESA 1.1.5 370 Feature, the locked pages were unlocked. **Upwardly compatible changes:** • You can read VMDUMP records with DIAGNOSE code X'34'. In VM/ESA 1.1.5 370 Feature, you could not. • This DIAGNOSE code will not execute if an external security manager is installed and the user does not have proper authorization. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.
X'3C'	Update VM directory **Incompatible changes:** • Condition code 2 in VM/ESA 1.1.5 370 Feature is split into condition codes 1 and 2 in VM/ESA 2.4.0.

Table 42 (Page 7 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'48'	Second Level SVC 76 **Incompatible changes:** • Does not do anything. In VM/ESA 1.1.5 370 Feature, it distinguished between two or more levels of virtual device number in EREP records. In VM/ESA 2.4.0, CP itself provides this function; you do not have to invoke it using the DIAGNOSE instruction. If your virtual machine is an XC virtual machine, DIAGNOSE code X'48' is not valid and you receive a specification exception.
X'4C'	Generate accounting records **Incompatible changes:** • You can no longer get accounting record identification code, C8. In VM/ESA 1.1.5 370 Feature, code C8 indicated virtual machine console accounting records. • For subcodes other than X'0010', the counter containing the number of LINK commands attempted with an invalid password is reset. **Upwardly compatible changes:** • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details. • The data area for subcode X'10' can cross a page boundary or wrap storage. In VM/ESA 1.1.5 370 Feature, it could not.

Table 42 (Page 8 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'58'	3270 virtual console interface DIAGNOSE code X'58' in the 370 environment is similar to Start I/O (SIO) instruction processing. See the *IBM System/370 Principles of Operation* for details about SIO. In the XA, ESA, and XC environments, DIAGNOSE code X'58' is similar to the Start Subchannel (SSCH) instruction. See the *VM/ESA: Enterprise Systems Architecture/Extended Configuration Principles of Operation* for details about SSCH. **Incompatible changes:** • Some terminal devices are no longer supported for VM/ESA 2.4.0. See the *VM/ESA: General Information* book for information on device support. • Only format 0 CCWs are allowed. • The channel address word (CAW) specified in Rx must be specified as follows: **Bit(s)** **Contents** 0-3 The protection key used in accessing guest absolute storage for the I/O operation. 4 Must be zero (suspend control). 5-7 Must be zero (unassigned). 8-31 The channel program address. 29-31 Must be zero, designating a doubleword-aligned channel program. • The guest virtual machine's condition codes (0, 1, 2, and 3) have changed for 370, XA, ESA, and XC virtual machines. See the *VM/ESA: CP Programming Services* book for details. • Rules for specifying a valid channel program has changed. See the *VM/ESA: CP Programming Services* book for details. • You can no longer use the CP TERMINAL SCRNSAVE ON command to save the full screen display when the screen enters line mode. **Upwardly compatible changes:** • There are several differences between DIAGNOSE code X'58' issued in a 370 virtual machine and issued in an XA, ESA, or XC virtual machine. If you run your programs in a 370 virtual machine then you do not have to consider these differences. – In XA, ESA, and XC virtual machines, the guest receives a condition code 0, with deferred condition code 1 in the subsequent interrupt response block (IRB). In a 370 virtual machine, if the channel program ends by the initial selection for the first CCW in channel program, the guest receives condition code 1 in the channel status word. – For DIAGNOSE code X'58', the application is responsible for all I/O status and error checking. In XA, ESA, and XC virtual machines, this is done by using the Test Subchannel (TSCH) instruction and by examining the subsequent IRB. For a 370 virtual machine, this is done by using the TEST I/O (TIO) instruction and by examining the subsequent virtual CSW. • To use 31-bit addressing in an XA, ESA, or XC virtual machine, the use of the indirect data address word CCW function is required.

Table 42 (Page 9 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'5C'	Error message editing **Upwardly compatible changes:** • You can use this DIAGNOSE code to edit error messages if the user's setting is SET EMSG IUCV. In VM/ESA 1.1.5 370 Feature, the IUCV operand did not exist. • Supports a new subcode: X'20'. • Supports four-digit message numbers. In VM/ESA 1.1.5 370 Feature, message numbers over 999 did not exist. • Returns a length of 11 for a four-digit number and allows for variable-length headers. In VM/ESA 1.1.5 370 Feature, four-digit numbers could not be returned. • New modifier code X'80' edits messages according to the EMSG setting of another virtual machine. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.
X'60'	Determining the virtual machine storage **Upwardly compatible changes:** • The response can be greater than 16MB. In VM/ESA 1.1.5 370 Feature, it could not.
X'64'	Find, load, purge a named saved segment **Incompatible changes:** • Removed return codes for CC=2: X'1' (1), X'0B3' (179). • New return codes for CC=2: X'035' (53), X'1C1' (449), X'1DB' (475), X'3F7' (1015), X'547' (1351), X'548' (1352), X'54D' (1367), X'54E' (1358), X'557' (1367). **Upwardly compatible changes:** • Supports saved segments within a user's virtual machine. Saved segment support is very different in VM/ESA 2.4.0. See“Saved Segments and Named Saved Systems Differences [1.1.5]” on page 254 for information on converting saved segments. See the *VM/ESA: Planning and Administration* book for information on creating and managing saved segments in VM/ESA 2.4.0. • Supports new subcodes: X'01', X'02', X'10' and X'18'. • Return code X'0CB' (203) can be issued for V=F guests. • This DIAGNOSE code will not execute if an external security manager is installed and the user does not have proper authorization. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.

Table 42 (Page 10 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'68'	Virtual Machine Communication Facility (VMCF) **Incompatible changes:** • The VMCMHDR COPY file is located in the HCPGPI macro library. • Systems that communicated through VMCF and received messages from the user ID identified in the Authorized Specific function cannot issue SEND functions to any other user ID. In VM/ESA 1.1.5 370 Feature, you could issue SEND functions to any other user ID. • You do not get RC=9. In VM/ESA 1.1.5 370 Feature, you received RC=9 if a VMCF SEND function attempted to transmit a message to a system which was used to identify system services. **Upwardly compatible changes:** • The message ID can be reused until the VMCF protocol path completes. In VM/ESA 1.1.5 370 Feature, it could not be reused. • This DIAGNOSE code will not execute if an external security manager is installed and the user does not have proper authorization. • Supports a new function: SETLIMIT (Subcode X'000C').
X'74'	Saving and Loading an Image Library File **Incompatible changes:** • The number of bytes specified in Ry+1 is not rounded to the nearest whole page before the pages are saved or loaded. Also, partial pages are saved or loaded. In VM/ESA 1.1.5 370 Feature, the number of bytes was rounded and partial pages were not saved or loaded. • A condition code is set to 0, 1, or 3. In VM/ESA 1.1.5 370 Feature, no condition code was set. • Removed return codes: X'0C', X'10'. • Changed return code: X'18', can additionally mean a spooling error. • New return codes: X'20', X'24', X'28', X'2C'. • You can now get a specification exception if the register specified for Rx is the same as Ry, or if Rx and Ry registers are not at least two registers apart. **Upwardly compatible changes:** • Privilege class E users can now issue this DIAGNOSE code. • Does not require the NAME3800 macro to be coded because the image libraries are saved into system data files and dynamically allocated out of spooling space. • Supports a new entry value: access register Ay. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.

Table 42 (Page 11 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'7C'	Logical Device Support Facility **Incompatible changes:** • If the logical device is dialed to a virtual machine, the dialed connection is reset when the TERMINATE function is issued. In VM/ESA 1.1.5 370 Feature, the logical device was detached from that virtual machine. • For subcode X'00000001' (Initiate): – Rx+1 must contain the model number in byte 2 and the device class and type in bytes 3 and 4. In VM/ESA 1.1.5 370 Feature, bytes 1, 2, and 3 (respectively) were used for this information. • For the logical device external interrupt code, reason code X'03' indicates that the user received a CC=0 and return code of 3 after a PRESENT. In VM/ESA 1.1.5 370 Feature, it indicated that a user received a CC=2 and return code of 2. • Upon entry, in the second byte of Rx+1, the first four bits are always ignored. In VM/ESA 1.1.5 370 Feature, these bits were sometimes used to indicate the last digit of a 3279 display or a 3284 printer. • Upon entry, in the fourth byte of Rx+1, the model numbers you can specify for 328X have all changed. Also, model numbers X'92' and X'93' are no longer valid for 3278s and 3279s. • Upon entry for the PRESENT function to a printer, if Rx+1 is zero, a device end interrupt is reflected asynchronously. In VM/ESA 1.1.5 370 Feature, it was not asynchronous. • Changed return codes: – CC=1 and RC=X'02', which you can receive with the status function, indicates that the logical device is not waiting for a status function. In VM/ESA 1.1.5 370 Feature, you did not receive this return code for the status function. – CC=1 and RC=X'0A' indicates the logical channel detected an error. In VM/ESA 1.1.5 370 Feature, it indicated a fetch or store protection violation. – CC=3 and RC=X'03', which you receive with the status function, indicates that the status is not valid for this logical device. In VM/ESA 1.1.5 370 Feature, you did not receive this return code for the status function. • Removed return codes: for CC=0, RC=X'05'; for CC=3, RC=X'1'; for CC=3, RC=X'06'. • Subcode X'00000002', ACCEPT, sets a return code of 0 whether the CCW is command chained. It also issues an external interrupt subcode X'02' for each new command chained CCW. In VM/ESA 1.1.5 370 Feature, ACCEPT did not perform this function. **Upwardly compatible changes:** • Upon return from an ACCEPT function, the maximum buffer length for an ACCEPT is X'7FFFFFFF'. In VM/ESA 1.1.5 370 Feature, the maximum buffer length for ACCEPT was X'1000'. • Can be used with privilege class ANY. • Allows 4-digit device addresses. • 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.

Table 42 (Page 12 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'84'	Directory Update-in-Place **Incompatible changes:** • No longer replaces information on dispatching priority. • On entry, the parameter list must be at least 24 bytes long. In VM/ESA 1.1.5 370 Feature, the parameter list could be one byte long. • For the MAXSTOR and STORAGE operation, values specified between 64KB and 1MB are rounded up to the next 64KB boundary. Values greater than 1MB are rounded up to the next 1MB boundary. In VM/ESA 1.1.5 370 Feature, entries not on a 4KB boundary were rounded up to the next 4KB boundary. • For the MAXSTOR and STORAGE operations, the minimum value you can specify is 64KB. The minimum in VM/ESA 1.1.5 370 Feature was 8KB. • For the OPTIONS operation, ACCT, CPUID, LANG, and VIRT=REAL are recognized, validated, but ignored. • The PRIORITY operation is accepted but ignored. • New condition code: 3. • New abend code: UDU001. • Removed return codes (in hex): 0A, 0B, 14-19, 1B, 32, 33, 34, 3C, 3D, 3E, 3F, 47, 52, 53, 5A, 70, and 71. • Changed return code: X'66' - indicates the parameter list is less than 24 bytes. In VM/ESA 1.1.5 370 Feature, it indicated the parameter list was less than or equal to 0. • Users defined with LBYONLY in the password field of the USER directory control statement cannot be the target of DIAGNOSE code X'84' operations unless the virtual machine issuing the DIAGNOSE code has the D84NOPAS option in its directory entry and the operation specified is not LOGPASS or MDISK. • For EDITCHAR operation, letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be specified as logical line edit symbols (line-end, line-delete, character-delete, and escape). **Upwardly compatible changes:** • Can now replace the following new information: – XAUTOLOG user IDs (XAUTOLOG operation) – Type of virtual machine - 370, XA, ESA, XC (MACHINE operation) – Maximum number of virtual processors (CPU operation) – Maximum number of spool files (SPOOLF operation) – User default system resource share option values (SHARE operation) – Expanded storage allocation (STORAGE and XSTORE operation) – Virtual processor ID and VECTOR and CRYPTO option values (CPU operation) – Minidisk device type, allocation definition, and volume serial number (RMDISK operation) – User's default date format setting (continued on next page)

Table 42 (Page 13 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'84' - continued	**Upwardly compatible changes (continued):** • For the MAXSTOR and STORAGE operations, the maximum value you can specify is 2047MB or 2096128KB. Also, you can specify megabytes or kilobytes. In VM/ESA 1.1.5 370 Feature, you could specify a maximum of 16000KB only in kilobytes. • On entry, the parameter list can be up to 223 bytes long. In VM/ESA 1.1.5 370 Feature, the parameter list could only be 112 bytes long. • New operations: CPU, DATEFMT, MACHINE, RMDISK, SHARE, SPOOLF, XAUTOLOG, and XSTORE. • The ACCOUNT operation can accept up to eight account numbers. In VM/ESA 1.1.5 370 Feature, you could specify only one account number. • The MDISK operation has an additional format that supports four-digit device addresses. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details. • New return codes due to new function (in hex): E6, E7, E8, E9, EA, F0, F1, F2, F3, F4, F5, F6, FA, FB, 104, 105, 106, 107, 108, 10E, 10F, 110, 111, 112, 118, 119, 11B, 11C, 122, 123, 124.
X'8C'	Access Certain 3270 Display Device Information **Incompatible changes:** • Rx cannot be register 15. In VM/ESA 1.1.5 370 Feature, you could use register 15. • You receive a protection exception if the buffer is not in the virtual machine storage. In VM/ESA 1.1.5 370 Feature, you received a protection exception if the address specified in Rx was in a protected area. • The length specified in Ry cannot be zero. • You can now get a specification exception if you specify Rx as register 15 or if you specify the length as zero. • The data returned can be longer than 502 bytes long. In VM/ESA 1.1.5 370 Feature, the data could only be 502 bytes long. **Upwardly compatible changes:** • You can now use this DIAGNOSE code with privilege class ANY. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.
X'90'	Read Symbol Table **Upwardly compatible changes:** • Supports a new entry value: access register Ay. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.

Table 42 (Page 14 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
`X'94'`	VMDUMP and Symptom Record Service **Incompatible changes:** • Removed return code: `X'30'`. In VM/ESA 1.1.5 370 Feature, it meant that the parameter list or dump list spanned the page boundary. • When multiple DIAGNOSE code `X'94'` invocations are used in continuous output mode, the NORETURN parameter is only accepted on the first request that opens the file. Once successfully specified, it may be repeated on later requests. However, if it is not specified on the first request, subsequent requests (affecting the same VMDUMP file) that specify NORETURN cause an error. • When multiple DIAGNOSE code `X'94'` invocations are used in continuous output mode, the DUMP parameter may be specified on any request to dump storage. • New condition codes set: 0, 1, and 2 are set with return codes, `X'00'`, `X'08'`, `X'0C'`. • New return codes: `X'38'`, `X'3C'`, `X'6C'`. • The system data file created cannot be processed with IPCS. You can use the Dump Viewing Facility. • If you did not specify *hexloc2*, the contents of virtual machine storage starting from *hexloc1* to the end of virtual storage and the contents of any associated DCSSs are dumped. In VM/ESA 1.1.5 370 Feature, contents of DCSSs are not dumped. **Upwardly compatible changes:** • New parameters: ALL, NODUMP, *symptom-record-addr*, CLOSE, CANCEL, CONT, SR, FILE, DCSS, XA, and NOCONT. • Can use this DIAGNOSE code to request that CP process a symptom record. • A new return code with new reason codes may be returned in registers Rx+1 and Ry+1 if you are processing symptom records. • This DIAGNOSE code will not execute if an external security manager is installed and the user does not have proper authorization. • Supports new entry values: access register Ax; an ALET can be specified in the dump address list. • These new return codes have been added: 84, 92, and 96. XC virtual machines in access-register mode can receive new return codes for access-register-translation exceptions and other errors in specifying address spaces: 64, 68, 72, 76, and 80. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.

Table 42 (Page 15 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'98'	Real I/O **Incompatible changes:** • Issuing DIAGNOSE code X'14', X'30', X'34', and X'38' against locked pages does not unlock them. In VM/ESA 1.1.5 370 Feature, the locked pages were unlocked. • You can get a specification error if an unknown subfunction is specified. • SIOF-Real (subcode X'00000008') does not support suspend/resume in channel programs. **Upwardly compatible changes:** • For the LOCK subfunction: – Return code 1 indicates that the virtual machine is a V=R or V=F machine. In VM/ESA 1.1.5 370 Feature, it only indicated a V=R machine. – On entry, Ry contains the guest address in bits 0 through 31 for 31-bit addressing. – On exit, Ry+1 contains the host address in bits 0 through 31 for 31-bit addressing. • For the UNLOCK subfunction: – Return code 1 indicates that the virtual machine is a V=R or V=F machine. In VM/ESA 1.1.5 370 Feature, it only indicated a V=R machine. – You can receive an operand exception for a VM/ESA input/output subfunction if an invalid subchannel ID or invalid entries in the ORB are found. • Supports a new subcode function: X'0000000C', SSCH-real subfunction. This subcode initiates execution of a real channel program for XA, ESA, or XC virtual machines. • In 24-bit addressing mode for XA and ESA virtual machines, the LOCK subfunction requires an available real storage frame in a reserved area. This area can be defined by specifying the RIO370 operand of the STORAGE statement in the system configuration (SYSTEM CONFIG) file. It can also be defined by coding the RIO370 operand on the SYSSTORE macro in HCPSYS ASSEMBLE. • SSCH-Real (subcode X'0000000C' does not support suspend/resume in channel programs. • Supports a new entry value, access register Ay, for the LOCK, UNLOCK and SSCH-Real subfunctions. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details. • The LOCK and UNLOCK subfunctions can return new return codes 5, 6, and 7 for errors in accessing address spaces.
X'A0'	Obtain ACI Groupname **Incompatible changes:** • Condition code 1 is no longer issued. In VM/ESA 1.1.5 370 Feature, CC=1 indicated that the request failed. • Can receive a specification exception if an invalid subcode is specified. **Upwardly compatible changes:** • Supports a new subcode: X'8' requests a test to determine if an access control interface is installed. • You can now use this DIAGNOSE code with privilege class ANY.

Table 42 (Page 16 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
`X'A8'`	Synchronous Input/Output Operations (All Devices) **Upwardly compatible changes:** • Supports FBA devices. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.
`X'B0'`	Access Re-IPL Data **Incompatible changes:** • Exit values removed: `X'05'`, `X'06'`. • Exit values added: `X'07'`, `X'08'`, `X'09'`, `X'0A'`, and `X'0B'`. For these new values, additional information is returned in the user's buffer. **Upwardly compatible changes:** • Returns additional information about the IPL directory statement. • You can now use this DIAGNOSE code with privilege class ANY. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.
`X'B4'`	Reading, Writing, and Erasing the Virtual Printer External Attribute Buffer **Incompatible changes:** • Removed return code: `X'25'`. In VM/ESA 1.1.5 370 Feature, this return code indicated that the user buffer was in protected storage. • If the buffer address is outside the range of user storage, you receive an addressing exception. In VM/ESA 1.1.5 370 Feature, you received return code `X'20'`. • If register 15 was specified for Ry, you get a specification error. In VM/ESA 1.1.5 370 Feature, a return code was not given in Ry+1 and PSW condition code was set to two. **Upwardly compatible changes:** • You can now use this DIAGNOSE code with privilege class ANY. • There is no restriction that the total storage that would be changed by reading the XAB must be addressable and cannot include any portion of a protected shared segment. In VM/ESA 1.1.5 370 Feature, this restriction existed. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.

Table 42 (Page 17 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'B8'	Spool File External Attribute Buffer Manipulation **Incompatible changes:** • Removed return code: X'25'. In VM/ESA 1.1.5 370 Feature, it indicated that the user buffer was in protected storage. • If the buffer address is outside the range of user storage, you receive an addressing exception. In VM/ESA 1.1.5 370 Feature, you received return code X'20'. • If register 15 was specified for Ry, you get a specification error. In VM/ESA 1.1.5 370 Feature, a return code was not given in Ry+1 and PSW condition code was set to two. **Upwardly compatible changes:** • You can now use this DIAGNOSE code with privilege class ANY. • There is no restriction that the total storage that would be changed by reading the XAB must be addressable and cannot include any portion of a protected shared segment. In VM/ESA 1.1.5 370 Feature, this restriction existed. • This DIAGNOSE code will not execute if an external security manager is installed and the user does not have proper authorization. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.
X'BC'	Open and Query Spool File Characteristics **Incompatible changes:** • Upon entry, Rx+1 contains the length of the buffer in double words. In VM/ESA 1.1.5 370 Feature, the length was not in double words. • Upon entry, Ry contains the subcode in bytes zero and one and contains the device address in bytes two and three. In VM/ESA 1.1.5 370 Feature, Ry contained the device address. • New condition code: X'01'. • If you do not specify subcode X'00' or X'04' correctly, you receive a specification exception. **Upwardly compatible changes:** • You can now use this DIAGNOSE code with privilege class ANY. • Supports a new subcode: X'04'. • Depending on the the specified buffer length, the user's buffer will contain the security label of the spool file. If an external security manager is installed, the user may not receive all the information about the file. The fields of the response that contain information are only the spool file ID, user ID, class, date, time, and status fields. All other fields will contain asterisks. • This DIAGNOSE code will not execute if an external security manager is installed and the user does not have proper authorization. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details. • 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.

Table 42 (Page 18 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'C8'	Set Language **Incompatible changes:** • The *langid* is set in the message repository. In VM/ESA 1.1.5 370 Feature, it was defined in DMKSNT. • Removed return codes: X'04', X'08', X'0C'. • Changed return code: X'24' indicates that an unrecoverable error occurred causing a soft abend. In VM/ESA 1.1.5 370 Feature, it indicated that no more virtual page buffers were available. **Upwardly compatible changes:** • You can now use this DIAGNOSE code with privilege class ANY.
X'CC'	Saving the CP Message Repository **Incompatible changes:** • The *langid* is set in the message repository. In VM/ESA 1.1.5 370 Feature, it was defined in DMKSNT. • Removed return codes: X'04', X'08', X'0C', X'14'. • New return codes: X'24', X'28', X'2C', X'30'. **Upwardly compatible changes:** • Supports a new entry value: access register Ay+1 • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.
X'D0'	Volume Serial Support **Incompatible changes:** • Removed return code: X'2'. **Upwardly compatible changes:** • You can now use this DIAGNOSE code with privilege class ANY. • The restriction about the address specified in Rx on entry not crossing a page boundary no longer applies. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details. • DIAGNOSE code X'0D0' returns new return codes 7, 8, and 9 for errors in accessing address spaces.

Table 42 (Page 19 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'D4'	Specify an Alternate User ID **Incompatible changes:** • DD4PARM0 is provided in the HCPGPI macro library. • If Rx and Ry are the same, you now receive a specification exception. **Upwardly compatible changes:** • Removed the following restrictions: – You must be on the same system as the worker machine – You must guarantee the identity of a remote user – You must provide the user ID of the end user – The parameter list containing the user ID cannot cross a page boundary. • Supports a new subcode: X'04'. • Subcode 0 of this DIAGNOSE code will not execute if an external security manager is installed and security label checking is enabled. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.
X'E4'	Return Minidisk Real Device Information **Incompatible changes:** • Additional information is appended to the information returned from subcodes X'00' and X'01'. • On entry, the address specified in Rx must be on a doubleword boundary and cannot cross a page boundary. • When specifying a virtual device number, you must give the actual device number in the input parameter list. • For the LEN field in the input parameter list, the recommended length has changed: – For subcodes X'00' and X'01', the length should be X'30' bytes. – For subcodes X'02', the length should be X'20' bytes. – For subcodes X'03', the minimum length is X'18' bytes. • Removed return codes: X'137', X'194', and X'19B'. These return codes are now reserved. **Upwardly compatible changes:** • Supports the dual copy function of 3990-3. • For subcodes X'02' and X'03', you can specify additional link modes, SR, SW, SM, ER, EW. • Additional meaning for return codes X'05', X'133', and X'197' if you exploit new function. • This DIAGNOSE code will not execute if an external security manager is installed and the user does not have proper authorization. • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details. • The VOLSER and RDEVNO fields in the output area for functions X'00' and X'01' contain different information for new support (virtual disks in storage).

Table 42 (Page 20 of 20). DIAGNOSE Codes Changed since VM/ESA 1.1.5 370 Feature

Code	Changes
X'EC'	Query GUEST Trace Status **Upwardly compatible changes:** • Supports new entry values: access registers Ax, Ay. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.
X'F8'	Spool file Origin Information **Upwardly compatible changes:** • Supports a new entry value: access register Ay. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details.
X'210'	Retrieve Device Information **Incompatible changes:** • In the CKD version of VRDCBLOK the following fields are removed: – VRDCPGCL at location 50 – VRDCPGC0 at location 52. • In the FBA version of VRDCBLOK the following fields are removed: – VRDCPGCG at location 32 – VRDCPGAP at location 36 – VRDCPGMA at location 3A – VRDCPGFA at location 3E – VRDCBKPG at location 42 – VRDCPGXT at location 4A – VRDCBKMX at location 56. • Diagnose code X'210' is introduced by the Support for the IBM 3390 Direct Access Storage Device enhancement (VM38627). **Upwardly compatible changes:** • Supports a new entry value: access register Ax. • There may be other considerations if you exploit VM data spaces. See the *VM/ESA: CP Programming Services* book for details. • Output can contain additional information.

CP Macros

Table 43 lists the APPC/VM macro functions and IUCV macro functions for use in APPC/VM that have changed since VM/ESA 1.1.5 370 Feature.

The information in this table is based on two assumptions:

1. You have used the parameter lists as documented in the publications (that is, you have not used any undefined parts of the parameter list for your own purposes).
2. When you are converting your APPC/VM applications, your communications partners are in the same environment they were in before your conversion (that is, your communications partners are not exploiting new function).

See the *VM/ESA: CP Programming Services* book for complete descriptions of APPC/VM and IUCV macros.

Table 43 (Page 1 of 2). CP Macros Changed since VM/ESA 1.1.5 370 Feature

Macro Function	Changes
IUCV ACCEPT	**Upwardly compatible changes:** • New value in the Connection Complete Interrupt: IPPOLLFG (Byte X'24'). • Input Parameter List changes: – The IPFLAGS1 parameter supports a new option: IPCNTRL (X'04').
APPCVM CONNECT	**Incompatible changes:** • New return code for CC=1 (in decimal): 92. • Output Parameter List changes: – IPASYRC, the fourth byte of the IPAUDIT field, can return new error codes (in decimal): 91, 92. **Upwardly compatible changes:** • You can connect to a system resource in addition to a local and global resource. • Output Parameter List changes: – New fields: - IPPOLLFG (Byte X'24'), interrupts only. • Connection Pending External Interrupt changes: – New fields: - IPPOLLFG (Byte X'24'), interrupts only. • For Allocate Data that your communication partner receives: – For a VM area: - Bytes X'70'–X'78' now contain information instead of being reserved.
IUCV ALLOW	**Incompatible changes:** • Specifying IUCV ALLOW on a target node no longer authorizes remote priority.
IUCV CONNECT	**Incompatible changes:** • Priority fails on remote connections if dependent on an IUCV ALLOW directory control statement on the target node to authorize the priority. Use either the IUCV ANY or IUCV target_userid PRIORITY control statements in the invoker's directory to ensure proper authorization. **Upwardly compatible changes:** • Bit 2 of the FLAG byte of the User Data Field for CONNECT now returns the resource as a system resource. • New value in the Connection Complete Interrupt: IPPOLLFG (Byte X'24'). • New optional parameters supported: TARGET=*address* LOCAL=YES|NO
IUCV DCLBFR	**Incompatible changes:** • New return code for CC=1 (in decimal): 92.
IUCV DESCRIBE	**Upwardly compatible changes:** • New field in the Output Parameter List: IPPOLLFG (Byte X'24'), interrupts only.
APPCVM RECEIVE	**Incompatible changes:** • New return codes for CC=1 (in decimal): 91, 92. • IPASYRC, the fourth byte of IPAUDIT, can return new return codes (in decimal): 91, 92, 93, 94. **Upwardly compatible changes:** • Output Parameter List changes: – New fields: - IPPOLLFG (Byte X'24'), interrupts only.

Table 43 (Page 2 of 2). CP Macros Changed since VM/ESA 1.1.5 370 Feature

Macro Function	Changes
IUCV RECEIVE	**Incompatible changes:** • IPRCODE can return new return codes (in decimal): 91, 92.
IUCV REPLY	**Incompatible changes:** • IPRCODE can return new return codes (in decimal): 93, 94.
APPCVM SENDCNF	**Upwardly compatible changes:** • Output Parameter List changes: – New fields: - IPPOLLFG (Byte X'24'), interrupts only.
APPCVM SENDCNFD	**Upwardly compatible changes:** • Output Parameter List changes: – New fields: - IPPOLLFG (Byte X'24'), interrupts only.
APPCVM SENDDATA	**Incompatible changes:** • New return codes for CC=1 (in decimal): 91, 92, 93, 94. • IPASYRC, the fourth byte of IPAUDIT, can return new return codes (in decimal): 91, 92, 93, 94. **Upwardly compatible changes:** • Output Parameter List changes: – New fields: - IPPOLLFG (Byte X'24'), interrupts only. • New field for Message Pending External Interrupt: IPPOLLFG (Byte X'24').
APPCVM SENDERR	**Upwardly compatible changes:** • Output Parameter List changes: – New fields: - IPPOLLFG (Byte X'24'), interrupts only.
APPCVM SENDREQ	**Upwardly compatible changes:** • New field for SENDREQ Interrupt: IPPOLLFG (Byte X'24').
APPCVM SEVER	**Upwardly compatible changes:** • Output Parameter List changes: – New fields: - IPPOLLFG (Byte X'24'), interrupts only.
IUCV SEVER	**Upwardly compatible changes:** • Output Parameter List changes: – New fields: - IPPOLLFG (Byte X'24'), interrupts only.
IUCV TESTCMPL	**Upwardly compatible changes:** • New fields in the Output Parameter List: – IPPOLLFG (Byte X'24'), interrupts only.

Message Complete External Interrupt Changes

The IPAUDIT field is expanded to four bytes. The fourth byte is called IPASYRC, and it can contain the following new error codes (in decimal): 91, 92, 93, 94.

CP Messages

The following CP messages do not exist in VM/ESA 2.4.0:

DMK315
DMK387
DMK388
DMK396
DMK1502
DMK1751
DMK6002
DMK6003
DMK6062
DMK6200

The following list identifies the CP messages that have changed since VM/ESA 1.1.5 370 Feature. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CP messages.

Note that the three-character prefix for CP messages has changed from DMK to HCP.

Action indicators (for example, the “E” in message HCP006E) in VM/ESA 2.4.0 messages may differ from the corresponding messages in VM/ESA 1.1.5 370 Feature, even though the message number may be the same.

The nonzero return code and accompanying message that VM/ESA 2.4.0 returns for errors may not be the same as the nonzero return code that VM/ESA 1.1.5 370 Feature returns for corresponding errors.

VM/ESA 2.4.0 suppresses leading zeros in responses more often than VM/ESA 1.1.5 370 Feature does.

HCP002E
HCP003E
HCP006E
HCP009E
HCP010E
HCP021E
HCP024E
HCP031E
HCP050E
HCP054E
HCP091E
HCP092E
HCP094E
HCP101E
HCP102E
HCP103E
HCP104E
HCP105E
HCP106E
HCP109E
HCP114E
HCP120E
HCP121E
HCP143E
HCP145I
HCP150A
HCP160E
HCP233E
HCP296E
HCP319E
HCP332E
HCP403I
HCP410E
HCP421E
HCP481E
HCP512I
HCP574I
HCP580I
HCP592I
HCP645E
HCP659E
HCP660E
HCP661E
HCP662I
HCP663E
HCP704E
HCP711D
HCP716D
HCP717D
HCP725D
HCP751E
HCP752E
HCP753E
HCP770E
HCP772E
HCP799E
HCP904W
HCP1001E
HCP1006E
HCP1010W
HCP1011E
HCP1018E
HCP1030E
HCP1115E
HCP1116E
HCP1120E
HCP1151E
HCP1159E
HCP1280E
HCP1283I
HCP1365E
HCP1451I
HCP1550E
HCP1791E
HCP1879E
HCP2002I
HCP2508E
HCP2554I
HCP2601E
HCP2760E
HCP2768E
HCP2779E
HCP2891I
HCP2982I
HCP2984I
HCP2985I
HCP2986I
HCP6014I

HCP6077E
HCP6150E
HCP6156E
HCP6157E
HCP6186I
HCP6283I
HCP6525E
HCP6706E
HCP6721E
HCP6728E
HCP6739E
HCP6743E
HCP6788E
HCP6802E
HCP6804E
HCP8028W
HCP8080I
HCP9020W
HCP9021W
HCP9022W
HCP9036W
HCP9225I
HCP9405E
HCP9408E
HCP9410I
HCP9422E

CMS Changes

This section identifies the CMS externals that have changed since VM/ESA 1.1.5 370 Feature. It contains the following subsections:

- “CMS Commands”
- “CMS File Pool Administration and Operator Commands” on page 408
- “XEDIT Subcommands” on page 413
- “CMS Macros” on page 417
- “Common Programming Interface (CPI) Communications Routines” on page 425
- “Preferred Routines (CSL Routines)” on page 426
- “CSLCNTRL File Changes” on page 445
- “Compatibility Routines” on page 446
- “CMS Pipelines” on page 446
- “CMS Messages” on page 446

CMS Commands

Table 44 lists the CMS commands that have changed since VM/ESA 1.1.5 370 Feature. See the *VM/ESA: CMS Command Reference* for complete descriptions of CMS commands.

Note: The three character module identifiers have been removed from the messages listed in the *VM/ESA: CMS Command Reference*. For example, a message that used to be listed as DMSAMS136S is now listed as DMS136S.

Obtaining CMS Help for Border Commands

A separate Help component is provided for CMS window border commands. This allows for commands to be accessed in Help which have the same abbreviation as a border command.

For example, in VM/ESA 1.1.5 370 Feature, if you tried to obtain Help on the CMS RENAME command using the abbreviation, you typed:

```
HELP CMS R
```

or

```
HELP R
```

Because there is an R window border command, you received Help on the R border command, not on the RENAME command.

In VM/ESA 2.4.0, there is a separate border command Help component. When you type:

```
HELP CMS R
```

or

```
HELP R
```

you receive Help on the RENAME command. To obtain Help on the R window border command, type:

```
HELP BORDER R
```

or

```
HELP BORD R
```

For more information on Help components, see the *VM/ESA: CMS User's Guide*.

CMS Command Compatibility Table

Table 44 shows changes to CMS commands.

Table 44 (Page 1 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
ACCESS	**Upwardly compatible changes:** • New parameter: MODE0. • New messages and return codes for new support. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added message for use with BFS: DMS2133E.
ASSEMBLE	**Incompatible changes:** • New message for empty files: DMS1229E (RC=88). In VM/ESA 1.1.5 370 Feature, the system did not allow empty files.
ASSGN	**Upwardly compatible changes:** • Supports new parameters: 3490C, 3490B, 18TRACK, XF, COMP, NOCOMP, and DEN 38K, DISK*m*. • New messages and return code (88) for new support.
COPYFILE	**Incompatible changes:** • New messages: DMS516E (RC=32), DMS1229E (RC=32), DMS2153E, DMS2154E, DMS2155E. • Changed message (new text possible): DMS173E. • You cannot copy an empty SFS file to a minidisk in VM/ESA 2.4.0. If you do attempt to do this, you may get different responses. • Changed return code: – Return code 0 is returned when the SFS file space threshold is exceeded. In VM/ESA 1.1.5 370 Feature, you received return code 4. **Upwardly compatible changes:** • Added messages and return codes for SFS and DFSMS/VM. • In a FILELIST environment that was entered with the new MIXEDON option, COPYFILE can now be used on files with mixed case file IDs. • You can now copy to a file that has the same file ID as the work file (COPYFILE CMSUTI). • Support for the new CMS SET RORESPECT command has been added. COPYFILE will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O.

Table 44 (Page 2 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
CREATE ALIAS	**Incompatible changes:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
CREATE LOCK	**Incompatible changes:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. **Upwardly compatible changes:** • Added messages for use with BFS: DMS2040E, DMS2133E. • Added new operand for use with BFS: *bfsid*.
CREATE DIRECTORY	**Upwardly compatible changes:** • New message for authorization failure from ESM: DMS1331E.
CREATE NAMEDEF	**Upwardly compatible changes:** • Supports a new operand: *fm*. • Supports a new option: FILEMODE. • New message: DMS2509E. • New return code: 24. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
CSLFPI	**Incompatible changes:** • VM/ESA 2.4.0 generates a significantly larger expansion than prior releases. **Upwardly compatible change:** • New return code: -13.
CSLGEN	**Incompatible changes:** • Changed messages: – DMS037E (text changed) – DMS1091E (text changed) – DMS1096E (text changed) – DMS1108I (text changed) – DMS1111E (text changed). **Upwardly compatible changes:** • Supports new options: FILETYPE, COPYTYPE, NOAUTO, AUTO, NOMAP, MAP. • New message: – DMS2055I (RC=4). • Changed messages: – DMS056E (new text possible) – DMS065E (new text possible) – DMS1090E (new text possible) – DMS1094E (new text possible) – DMS1236E (new text possible) – DMS1096E (new text possible) – DMS1237E (new text possible) – DMS2055I (new text possible)

Table 44 (Page 3 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
CSLLIST	**Incompatible changes:** • New messages: – DMS213E – DMS2056E (RC=28). • Profiles for CSLLIST written for prior releases may not be compatible. **Upwardly compatible changes:** • Initial display screen from the command line now has a Path and Subgroup column. • Initial display screen changes: **PF6** Allows easy access to CSLMAP. **PF10** RTNDROP function has been replaced by the DROP function in CSLMAP, so now PF6 displays assigned attributes. **PF11** Sort by library name - the function of the key is changed only if the new SUBGROUP option on the ROUTINE input record for CSLGEN is used. If the SUBGROUP option is used, it sorts secondarily on the subgroup column rather than the routine name. • Supports a new command symbol, /g, that you can use with commands you issue from the Cmd area. • 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.
CSLMAP	**Upwardly compatible changes:** • 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.
DEBUG	**Upwardly compatible changes:** • The information that debug displays includes access registers.
DEFAULTS	**Upwardly compatible changes:** • New commands can be specified as parameters: INCLUDE, LOAD, CSLMAP, NAMES, AUTHLIST, ALIALIST. • New options can be specified as parameters for DISK LOAD and READCARD: FIFO, LIFO, NOTYPE, STACK, TYPE. • New options are supported as parameters for FILELIST and DIRLIST: OWNER, READWRITE, READONLY. • A new option is supported as a parameter for NETDATA SEND, SENDFILE, and NOTE: CLASS. • A new option is supported for NAMES: VMLINK. • Support for the new command VMLINK. • Supports new options: NOKEEPCC, KEEPCC. • New options are supported as parameters for FILELIST, NETDATA, and RDRLIST: VMDATE, SHORTDATE, FULLDATE, ISODATE.
DELETE LOCK	**Incompatible changes:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*.

Table 44 (Page 4 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
DIRATTR	**Incompatible changes:** • New message: DMS2436E. • Changed message: DMS2035W (text changed). **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DIRLIST	**Incompatible changes:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. **Upwardly compatible changes:** • Supports new options: MDISK, NOMDISK, NICK, NONICK, OWNER, READWRITE, READONLY. • New message: DMS1230E. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • 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.
DISK	**Upwardly compatible changes:** • Supports new operands for DISK LOAD: *filename*, *filetype*, *filemode*. • Supports new options: FIFO, LIFO, NOTYPE, STACK, TYPE, MSGALL, MSGSUBS. • New responses and message for empty files. In VM/ESA 1.1.5 370 Feature, empty files were not supported.
DLBL	**Upwardly compatible changes:** • File modes R and T are now supported.
DOSLIB	**Upwardly compatible changes:** • The following options can be fully spelled out: – DEL can be DELETE – COMP can be COMPRESS. Their abbreviations are still supported. • Supports a new option: TYPE.
DOSLKED	**Upwardly compatible changes:** • Supports new options: TYPE, INV, NOINV. • Can now run in XA and XC virtual machines. • New message for empty files: DMS2521E. In VM/ESA 1.1.5 370 Feature, the system did not allow empty files.
DSERV	**Upwardly compatible change:** • Supports a new option: TYPE.
EDIT	**Incompatible changes:** • The '(OLD' option is no longer supported. You must either use the XEDIT simulation by dropping the '(OLD' option or convert to using the normal XEDIT environment. • Message DMS987E was changed to DMS2520E. The old CMS editor is no longer supported.

Table 44 (Page 5 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
ERASE	**Incompatible changes:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. • Removed messages: DMS1160E, DMS1198E. **Upwardly compatible changes:** • Supports new options: DATAONLY and ENTIRE. • New messages: DMS2039E, DMS2500I, DMS1188E. • New message for authorization failure from ESM: DMS1332E. • In a FILELIST environment that was entered with the new MIXEDON option, ERASE can now be used on files with mixed case file IDs. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*.
ESERV	**Upwardly compatible changes:** • File modes R and T are now supported.
EXEC	**Upwardly compatible changes:** • New messages: – DMS042E (RC=24) – DMS240E (RC=-3) – DMS1229E (RC=88).
EXECIO	**Incompatible changes:** • A write to a file in an SFS directory that is in a VM/ESA 1.2.2 or later file pool fails when CMS detects that the file space is full. You receive an error message. In VM/ESA 1.1.5 370 Feature, your application did not receive a file space full condition, and you could continue writing. **Upwardly compatible changes:** • New message: DMS621W.
EXECLOAD	**Upwardly compatible changes:** • New message for empty files: DMS2521E (RC=88). In VM/ESA 1.1.5 370 Feature, the system did not allow empty files.
EXECUPDT	**Upwardly compatible changes:** • New responses for empty files. In VM/ESA 1.1.5 370 Feature, the system did not allow empty files. • New message: DMS1229E.
FETCH	**Upwardly compatible change:** • New message: DMS2521E.
FILEATTR	**Incompatible change:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. • Removed message: DMS1160E. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 44 (Page 6 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
FILEDEF	**Upwardly compatible changes:** • New options: DISP KEEP, DISP PASS, 3490C, 3490B, 18TRACK, XF, COMP, NOCOMP, LIBSRV. • The CONCAT option supports text libraries. • Allows LRECL definitions up to 65535 bytes for OS variable spanned records (under XLRI processing) and non-OS CMS files. • Allows BLKSIZE definitions up to 65535 bytes for non-OS CMS files.
FILELIST	**Incompatible changes:** • Changed responses: – The “Type” column displayed with the SHARE option may list the following new type: - EXTRNL for external objects. - BASE* for a base file that is in DFSMS/VM migrated status - ALIAS* for an alias of a file that is in DFSMS/VM migrated status. – The “Records” and “Blocks” columns displayed using the STATS option show a dash for external objects and a zero for empty files (these are for SFS support). • 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 LISTFILE profiles. See Appendix A of the *VM/ESA: CMS Command Reference*. • If you file the file created by FILELIST, that file might contain new and changed fields (on the far right). **Upwardly compatible changes:** • Supports new options: ALLDATES, AFTER, BEFORE, TODAY, MIXEDON, MIXEDOFF, *dirid*, OWNER, READWRITE, READONLY. • New messages: DMS2538E, DMS1230E. • Changed responses: – The ALLFILE option response includes external objects. • Within a FILELIST environment that was entered with the new MIXEDON option, the COPYFILE, RENAME, ERASE, and DISCARD commands can now be used on files with mixed case file IDs. • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • The BEFORE *date* and AFTER *date* options support 4-digit years. • Screens and responses support 4-digit years. • If a date format option is not specified on the FILELIST command, the CMS DEFAULTS date format setting for FILELIST will be used. • Added message for use with BFS: DMS2133E. • 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.
FINIS	**Incompatible changes:** • Will close files opened by the REXX STREAM I/O function.

Table 44 (Page 7 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes

FORMAT	**Incompatible changes:**
	- The format of the response to the RECOMP option has changed. `'BLKSZ'` replaces `'BLKSIZE'` in the response header. The `'FILES'` field is eight bytes, the `'CYL'` field is five bytes, and the `'VDEV'` field is four bytes. In VM/ESA 1.1.5 370 Feature, the `'FILES'` field was nine bytes, the `'CYL'` field was four bytes, and the `'VDEV'` field was four bytes.
	An example of the VM/ESA 2.4.0 response:
	`LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL`
	`TOOLIB 530  Z   R/O   300 3380 4096     5513   32084-71     12916      45000`
	An example of the VM/ESA 1.1.5 370 Feature response:
	`LABEL  VDEV M  STAT  CYL TYPE BLKSIZE  FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL`
	`TOOLIB 530  Z   R/O  300 3380 4096      5513   32084-71     12916      45000`
	Upwardly compatible change:
	- You can specify up-to-10 decimal digits of FBA blocks. In the VM/ESA 1.1.5 370 Feature, you could specify up-to-eight decimal digits.
	- Supports ECKD channel programs.

GENMOD	**Upwardly compatible changes:** - New option: XC. - New message and return code for new support: DMS988E (RC=64).
GENMSG	**Incompatible changes:** - Changed message: DMS775W (text changed). **Upwardly compatible changes:** - New message for empty files: DMS1229E (RC=88). In the VM/ESA 1.1.5 370 Feature, the system did not allow empty files.
GLOBAL	**Upwardly compatible changes:** - New message for empty files: DMS559W. In the VM/ESA 1.1.5 370 Feature, the system did not allow empty files. - New message for duplicate library name in input list: DMS045W. The duplicates are ignored.
GRANT AUTHORITY	**Incompatible changes:** - Added message: DMS149E. - Removed message: DMS1198E. **Upwardly compatible changes:** - The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
HELP	**Upwardly compatible changes:** - More component names are allowed for the *component name* operand. - New message for empty files: DMS1229E (RC=88). In the VM/ESA 1.1.5 370 Feature, the system did not allow empty files.
HELPCONV	**Upwardly compatible changes:** - Added message and return code for empty file support.

Table 44 (Page 8 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
IDENTIFY	**Upwardly compatible changes:** • New options to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses support 4-digit years. • 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. • New option for TCP/IP mail integration: TCPIP.
INCLUDE	**Incompatible changes:** • If you are converting from VM/ESA 1.1.5 370 Feature with APAR VM46641 applied, the default for turning on the high order bit in VCONs is changed to NOHOBSET, which means that no high order bit of any VCONs gets turned on. In VM/ESA 1.1.5 370 Feature with APAR VM46641 applied, the default was HOBSETSD, which meant that SD-type VCONs would get their high order bits turned on. • If you are converting from VM/ESA 1.1.5 370 Feature without APAR VM46641 applied, three new options are added: HOBSETSD, NOHOBSET, HOBSET. The default is NOHOBSET, which means that no high order bit of any VCON gets turned on. In VM/ESA 1.1.5 370 Feature, there were no HOBSETSD, NOHOBSET, HOBSET options, but the default action that took place automatically turned on the high order bit of SD-type VCONs. The default is for no VCONs of any type to get their high-order bits turned on. **Upwardly compatible changes:** • New option: FULLMAP. • New message for empty files: DMS2521E (RC=88). In VM/ESA 1.1.5 370 Feature, the system did not allow empty files. • Changed message: DMS056E (appended text).
LABELDEF	**Upwardly compatible changes:** • The CRDTE and EXDTE operands support specifying a century. • New message: DMS649E.
LISTDIR	**Incompatible changes:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 44 (Page 9 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command | **Changes**

LISTDS

Incompatible change:

- The format of the response may be different.

  For the FORMAT option, the information in the response fields are left justified. In VM/ESA 1.1.5 370 Feature, the information in the response fields was centered.

  For the EXTENT and FREE options, some fields in the response can have more digits, some fields are shifted, and the header is different.

  Examples of VM/ESA 2.4.0 responses:

  For `'LISTDS` *fm* `(EXTENT'` for CKD:

```
Extent information for VTOC on G disk:
SEQ TYPE    CYL-HEAD    (RELTRK) TO   CYL-HEAD    (RELTRK)     TRACKS
000 VTOC  00099 00000       1881    00099 00018       1899        19
```

  For `'LISTDS` *fm* `(FREE'` for CKD:

```
Freespace extents for D disk:
  CYL-HEAD    (RELTRK) TO   CYL-HEAD    (RELTRK)     TRACKS
00000 00010         10    00009 00014        149        140
```

  For `'LISTDS` *fm* `(EXTENT'` for FBA:

```
Extent information for VTOC on B disk:
SEQ TYPE     BLOCKNO  TO    BLOCKNO         BLOCKS
000 VTOC  0000000002     0000000031             30
```

  For `'LISTDS` *fm* `(FREE'` for FBA:

```
Freespace extents for B disk:
FB/E BLOCKNO  TO  FB/E BLOCKNO      BLOCKS
  0000000032       0000001999        1968
```

  Examples of VM/ESA 1.1.5 370 Feature responses:

  For `'LISTDS` *fm* `(EXTENT'` for CKD:

```
Extent information for VTOC on C disk:
SEQ TYPE   CYL-HD(RELTRK) TO  CYL-HD(RELTRK) TRACKS
000 VTOC  0099 00   1881     0099 18   1899      19
```

  For `'LISTDS` *fm* `(FREE'` for CKD:

```
Freespace extents for C disk:
 CYL-HD(RELTRK) TO  CYL-HD(RELTRK)   TRACKS
0000 10     10     0099 14     149      140
```

  For `'LISTDS` *fm* `(EXTENT'` for FBA:

```
Extent information for VTOC on B disk:
SEQ TYPE  BLOCKNO    TO     BLOCKNO         BLOCKS
000 VTOC  000002            000031             30
```

  For `'LISTDS` *fm* `(FREE'` for FBA:

```
Freespace extents for B disk:
FB/E BLOCKNO  TO  FB/E BLOCKNO      BLOCKS
     000032           001999        1968
```

Upwardly compatible changes:

- New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE.
- Responses support 4-digit years.
- 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.

Table 44 (Page 10 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
LISTFILE	**Incompatible changes:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. • Responses have changed: – The "TYPE" column displayed with the SHARE option can list the following new types: - EXTRNL for external objects. - BASE* for a base file that is in DFSMS/VM migrated status - ALIAS* for an alias of a file that is in DFSMS/VM migrated status. – The "RECS" and "BLOCKS" columns displayed when the ALLOC option is used show a dash for external objects and a zero for empty files. – The "FORMAT" and "LRECL" columns displayed when the ALLOC or FORMAT options are used show a dash for external objects. • New message: DMS389E. **Upwardly compatible changes:** • Supports new options: ALLDATES, AFTER, BEFORE, TODAY. • You can use special characters (* and %) to specify a set of files to list in an open directory. • New options for ALLDATES option: DTOC, DOLR, DTOLU, DTOLC. • New options to specify date format: SHORTDATE, FULLDATE, ISODATE. • The BEFORE *date* and AFTER *date* options support 4-digit years. • Responses support 4-digit years. • 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.
LKED	**Upwardly compatible changes:** • Supports new options: TYPE, NOTYPE. • New message for empty files: DMS1229E (RC=88). In VM/ESA 1.1.5 370 Feature, the system did not allow empty files. • The defaults for *value1* and *value2* on the SIZE option have been increased to 400K and 100K, respectively.
LOAD	**Incompatible changes:** • If you are converting from VM/ESA 1.1.5 370 Feature with APAR VM46641 applied, the default for turning on the high order bit in VCONs is changed to NOHOBSET, which means that no high order bit of any VCONs gets turned on. In VM/ESA 1.1.5 370 Feature with APAR VM46641 applied, the default was HOBSETSD, which means that only the SD-type VCONs would get their high order bits turned on. • If you are converting from VM/ESA 1.1.5 370 Feature without APAR VM46641 applied, three new options are added: HOBSETSD, NOHOBSET, HOBSET. The default is NOHOBSET, which means that no high order bit of any VCON gets turned on. In VM/ESA 1.1.5 370 Feature without the APAR applied, there were no HOBSETSD, NOHOBSET, or HOBSET options, but the default action that took place automatically turned on the high order bit of SD-type VCONs. The default is for no VCONs of any type to get their high-order bits turned on. **Upwardly compatible changes:** • Added message for empty files: DMS2521E (RC=88). In VM/ESA 1.1.5 370 Feature, the system did not allow empty files. • New message for insufficient storage above 16MB: DMS891W. • Changed message: DMS056E (appended text).

Table 44 (Page 11 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
LOADLIB	**Upwardly compatible changes:** • Supports a new operand: MAP. • Supports a new option: TYPE. • The COMP operand can be fully spelled out: COMPRESS. COMP is still supported as an abbreviation. • REP is now a supported abbreviation for the REPLACE option.
LOADMOD	**Incompatible changes:** • Changed message (new text possible): DMS639E. **Upwardly compatible changes:** • Supports new options: ORIGIN *hexloc*. • New message for empty files: DMS2521E (RC=88). In VM/ESA 1.1.5 370 Feature, the system did not allow empty files. • Changed message: DMS988E (new possible text for XC).
MACLIB	**Upwardly compatible changes:** • Supports new options: TYPE, FILEMODE *fm*. • The following operands can be fully spelled out: – GEN can be GENERATE – REP can be REPLACE – DEL can be DELETE. Their abbreviations are still supported.
MACLIST	**Incompatible changes:** • New message: DMS213W (RC=4). **Upwardly compatible changes:** • Supports new options: COMPRESS, NOCOMPRESS. • Changed operand: *libname*. You can use "*filename* MACLIB *filemode*." In VM/ESA 1.1.5 370 Feature, you could use only *filename*. • New message: DMS032E. • 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.
MACLMIG	**Upwardly compatible changes:** • ESA and 370 options are available, but only ESA libraries are substituted. These options were used in VM/ESA 1.1.0 to distinguish between the 370 Feature and the ESA Feature. These features are not available on VM/ESA 2.4.0. • New message: DMS2048W (indicates that the 370 option is ignored).
MOVEFILE	**Upwardly compatible changes:** • New response, messages, and return code for empty files. In VM/ESA 1.1.5 370 Feature, the system did not support empty files. • Can process OS variable spanned records (under XLRI processing) and non-OS CMS files with record lengths up to 65535 bytes. • Adjusts output file sizes for compatibility between CMS and OS • Allows greater FILEDEF default flexibility for file attributes (RECFM, LRECL, BLKSIZE) • If default size values are used, fixes record truncation problems when moving data files from fixed to variable format • New message: DMS1116E.

Table 44 (Page 12 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
NAMEFIND	**Incompatible changes:** • Removed message: DMS633E (RC=88). **Upwardly compatible changes:** • Supports new options: STEM, MSG, NOMSG, FIND, LOCATE, FINDWORD, LOCWORD, ALL, BREAKTAG. • The default value of the SIZE option was changed to "*." In VM/ESA 1.1.5 370 Feature, the default value was "8." • The limit on the number of tags for a names file entry was eliminated. In VM/ESA 1.1.5 370 Feature, the maximum number of tags in a names file entry was 64. • The new STEM option can be used to return more than 255 characters for tags and values from a names file. However, CMS commands, such as NOTE and TELL, are limited to 255-character tag values. • NAMEFIND is now AMODE 31. It attempts to allocate its buffer storage above the 16MB line. If storage above this line is not available, NAMEFIND attempts to allocate storage below the 16MB line. • New messages: – DMS2539E (RC=24 in NAMEFIND environment) – DMS2540T – DMS2541E (RC=38 in NAMEFIND environment). • Supports the new form of the FILE opt, FILE (fn ft fm).

Table 44 (Page 13 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
NAMES	**Incompatible changes:** • The NAMES panels have changed. Multiple panels and multiple screens per entry are supported, the PF key assignments have changed, and a panel interface was added for the Find (PF5) support. • Changed messages: – DMS660W (text changed). • Removed message: DMS657E. • Removed response: – Warning: There {is|are} *nn* undisplayed tag(s). **Upwardly compatible changes:** • Supports new options: MAIL, ALTMAIL, PANEL, FILE, COMDIR, SERVER, VMLINK. • On the MAIL panel: – The maximum number of characters for the Nickname field was extended to 21. In VM/ESA 1.1.5 370 Feature, the maximum number of characters was 8. – The maximum number of characters for the Userid and Node fields was extended to 70 characters to allow users to use the MAIL panel to create nickname entries for communications such as TCP/IP. In VM/ESA 1.1.5 370 Feature, the maximum number of characters was 8. The Userid and Node fields will no longer be uppercased. • CMS communications directory support changed with new options. • Support to create and edit the APPC/VM private resources authorization file, “$SERVER$ NAMES,” was added. • New messages: – DMS1007E – DMS1008E – DMS1009E – DMS1009W – DMS1013W – DMS1014E – DMS1014W – DMS1015E – DMS1015W. • New responses (for COMDIR panel only): – No value for Password field for this entry – Value for Password file is not displayed.
NETDATA	**Incompatible changes:** • Different message on empty reader condition: DMS205W (was DMS639E). • Changed message: DMS636W (text changed). • New message: DMS1184E • New return code: 20 **Upwardly compatible changes:** • NETDATA SEND supports a new option: CLASS. • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file. • Responses support 4-digit years. • 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.

Table 44 (Page 14 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
NOTE	**Incompatible changes:** • In a note, a nonblank line between the 'OPTIONS:' line and the 'Date:' line, called 'USEROPTIONS:', is supported. Any other nonblank lines cause an error. In VM/ESA 1.1.5 370 Feature, nonblank lines between the 'OPTIONS:' line and the 'Date:' line were ignored. • Added messages: – DMS149E – DMS653E – DMS1012E – DMS2501E **Upwardly compatible change:** • The OPTIONS: line of a note contains a new option: CLASS. • In the Date field of the note header, the year is now displayed with four digits. • TCP/IP domain names accepted as user IDs or as the resolution of nicknames. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
NUCXDROP	**Incompatible change:** • Changed message: DMS617E (text changed). **Upwardly compatible changes:** • `NUCXDROP *` does not drop nucleus extensions loaded with the new PERMANENT attribute specified on the NUCXLOAD command. They must be dropped by name. In VM/ESA 1.1.5 370 Feature, the PERMANENT attribute did not exist. • New message and return code: DMS624W, return code 4.
NUCXLOAD	**Incompatible changes:** • Changed message (new text possible): DMS639E. **Upwardly compatible changes:** • Supports a new parameter, PERMANENT, which protects the extension from being dropped by `NUCXDROP *`. • New message and return code for empty files: DMS2521E (RC=88). In VM/ESA 1.1.5 370 Feature, the system did not allow empty files.
NUCXMAP	**Incompatible changes:** • You may receive two lines of output in the same situation that you received one line in VM/ESA 1.1.5 370 Feature. **Upwardly compatible changes:** • New option: PERMANENT. • The response may contain new information if you use the new option.
OPTION	**Upwardly compatible changes:** • Supports new options: TYPE, NOTYPE.
OSRUN	**Upwardly compatible changes:** • New message and return code: DMS033E (RC=12).
PARSECMD	**Upwardly compatible changes:** • Supports new options: TRANSLSAME, TRANSLYES, TRANSLNO, TRANSLCMS. • BFS added new code.n value of PN.

Table 44 (Page 15 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
PEEK	**Incompatible changes:** • PEEK profile has changed. If you used the PROFPEEK profile to create your own, PEEK may not work anymore. Use new PROFPEEK. **Upwardly compatible changes:** • TCP/IP origin domain name address used when available and shown on PEEK message line for origin within current space and formatting limitations.
PIPE	**Incompatible changes:** • All CMS Pipelines messages have a new prefix, and many messages have new numbers and text. See “CMS Pipelines” on page 446.
PRINT	**Upwardly compatible changes:** • Supports new options: EJECT, NOEJECT. • Use of the PRINT command is restricted if an external security manager (ESM) is installed and security label checking is enabled. If the ESM fails this command, message HCP356E, HCP1561E, or HCP2514E is issued to the system operator.
PSERV	**Upwardly compatible changes:** • Supports a new option: TYPE.
PUNCH	**Upwardly compatible change:** • New message for empty files: DMS118E. In VM/ESA 1.1.5 370 Feature, the system did not allow empty files.
QUERY (in general)	**Incompatible changes:** • Removed message: DMS1160E. • See changed commands below. **Upwardly compatible changes:** • Supports new operands.
QUERY ACCESSED	**Incompatible changes:** • The format of the response has changed. The 'Files' field allows 10 digits. In VM/ESA 1.1.5 370 Feature, it allowed eight digits.
QUERY ALIAS	**Incompatible changes:** • The response has changed. The “T” column of the output has been expanded to two characters wide. It may now show the following new types: – A* for an alias of a base file migrated by DFSMS/VM – B* for a base file migrated by DFSMS/VM. • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 44 (Page 16 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
QUERY AUTHORITY	**Incompatible changes:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. **Upwardly compatible changes:** • External objects are supported. • The "Type" column of the response may now list the following new type: – EXTRNL for external objects. – BASE* for base files migrated by DFSMS – ALIAS* for aliases of base files migrated by DFSMS. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
QUERY CMSLEVEL	**Incompatible changes:** • In the response, the CMS level value has changed and is now two digits.
QUERY CMSREL	**Incompatible changes:** • The format of the response has changed to include the version. **Upwardly compatible changes:** • In the response, the release level value has changed.

QUERY DISK	**Incompatible changes:**
	• There are changes to the response:
	– The format of the response has changed. 'BLKSZ' replaces 'BLKSIZE' in the response header. The 'FILES' field is eight bytes, the 'CYL' field is five bytes, and the 'VDEV' field is four bytes. In VM/ESA 1.1.5 370 Feature, the 'FILES' field was nine bytes, the 'CYL' field was four bytes, and the 'VDEV' field was four bytes.
	An example of the VM/ESA 2.4.0 response:
	`LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL`
	`TOOLIB 530  Z   R/O   300 3380 4096     5513      32084-71      12916      45000`
	An example of the VM/ESA 1.1.5 370 Feature response:
	`LABEL  VDEV M  STAT  CYL TYPE BLKSIZE  FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL`
	`TOOLIB 530  Z   R/O  300 3380 4096      5513      32084-71      12916      45000`
	– When the calculated percentage is zero for the field that contains the percentage of blocks in use, BLKS USED-(%), but there are files on the disk, the percentage is displayed as "01." In VM/ESA 1.1.5 370 Feature, this was displayed as "00."

QUERY DLBL	**Upwardly compatible changes:** • Supports new operands: EXTENT, MULT.
QUERY ENROLL	**Incompatible changes:** • Added message: DMS149E. **Upwardly compatible changes:** • New default added: FOR ALL. In VM/ESA 1.1.5 370 Feature, you had to specify one of the following operands: FOR ALL, FOR *userid*, FOR *nickname*. • The colon (:) on the *filepoolid* option is now optional. • Added new option for use with BFS: FILESPACE.

Table 44 (Page 17 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
QUERY FILEATTR	**Incompatible changes:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. • The response has changed: – The "Type" column of the response may now list the following new type: - EXTRNL for external objects. - BASE* for a base file that is in DFSMS/VM migrated status - ALIAS* for an alias of a file that is in DFSMS/VM migrated status. – The "Recovery" and "Overwrite" columns show a dash for external objects. **Upwardly compatible changes:** • You can now use QUERY FILEATTR on an open directory. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*.
QUERY FILEDEF	**Upwardly compatible changes:** • Supports a new optional operand, ATTRIBUT, and its response.
QUERY FILEPOOL	See"CMS File Pool Administration and Operator Commands" on page 408 for information about the changes to this command.
QUERY LIMITS	**Incompatible change:** • File pool administration authority is no longer needed to query limits on a filespace other than your own. • Added message: DMS149E. **Upwardly compatible change:** • New default added: *. In VM/ESA 1.1.5 370 Feature, you had to issue one of the following operands: FOR, *, or ALL. • The colon (:) on the *filepoolid* option is now optional.
QUERY LOCK	**Incompatible changes:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. • The "Type" column of the response may list the following new types: – BASE* for a base file that is in DFSMS/VM migrated status – ALIAS* for an alias of a file that is in DFSMS/VM migrated status **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*.
QUERY NAMEDEF	**Upwardly compatible changes:** • Added new operand for use with BFS: *bfsid*.
QUERY OPTION	**Upwardly compatible changes:** • The TERM or NOTERM setting is added to the end of the response information.
QUERY RORESPECT	**Upwardly compatible changes:** • Displays the current RORESPECT setting. This setting controls whether or not updates to files using XEDIT and/or COPYFILE in SFS will respect the read-only access of a directory.

Table 44 (Page 18 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes

QUERY SYSNAMES	**Upwardly compatible changes:**
	• There is a change to the response. The CMSGUI saved system name is displayed.
	An example of the VM/ESA 2.4.0 response:
	`SYSNAMES:  CMSVSAM  CMSAMS   CMSDOS   CMSBAM   CMSGUI`
	`ENTRIES:   CMSVSAM  CMSAMS   CMSDOS   CMSBAM  VMGUILIB`

RDRLIST	**Incompatible changes:** • If you file the file created by RDRLIST, that file might contain new and changed fields (on the far right). **Upwardly compatible changes:** • If an external security manager is installed and security label checking is enabled, all fields on the RDRLIST panel except for Class, User, Node, Hold, Date, and Time may be masked with asterisks. This occurs when the user does not have the necessary security label authorization to work with that reader file. • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Screen supports 4-digit years. • If a date format option is not specified on the RDRLIST command, the CMS DEFAULTS date format setting for RDRLIST will be used. • 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. • TCP/IP origin domain name addresses used when available and shown on RDRLIST panel origin area within current space and formatting limitations.
READCARD	**Upwardly compatible changes:** • Supports new options: FIFO, LIFO, NOTYPE, STACK, TYPE, MSGALL, MSGSUB. • New message for empty files: DMS701W. In VM/ESA 1.1.5 370 Feature, the system did not allow empty files.
RECEIVE	**Incompatible changes:** • When you receive DISK DUMP format files, all the files get created on the specified file mode. Entries are made in your NETLOG file for all the files created. In VM/ESA 1.1.5 370 Feature, just the first file was created on the specified file mode and logged in your NETLOG file. The other files, if any, were created on the A-disk and not logged. • When receiving DISK DUMP format files to a file mode other than A, you are not required to have the A file mode accessed as read/write. In VM/ESA 1.1.5 370 Feature, you were required to have an A-disk accessed as read/write. • Added messages: – DMS653E – DMS1012E – DMS1138E **Upwardly compatible changes:** • Supports empty files. • New messages: DMS636W, DMS3033W. • Supports new options: NOKEEPCC, KEEPCC. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
RELEASE	**Upwardly compatible changes:** • The DET option can be fully spelled out: DETACH. DET is still supported as an abbreviation. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 44 (Page 19 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
RELOCATE	**Incompatible changes:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
RENAME	**Incompatible changes:** • The restriction that the specified directory or directories must not be open no longer exists. In VM/ESA 1.1.5 370 Feature, you received message DMS1160E if the specified directories were open. **Upwardly compatible changes:** • In a FILELIST environment that was entered with the new MIXEDON option, RENAME can now be used on files with mixed case file IDs. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
REVOKE AUTHORITY	**Incompatible changes:** • Added message: DMS149E. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
RSERV	**Upwardly compatible change:** • Supports a new option: TYPE.
RTNDROP	**Incompatible changes:** • Changed messages: – DMS1087E (text changed) – DMS1129W (text changed). **Upwardly compatible changes:** • Changed message: – DMS066E (new text possible) – DMS1088W (new text possible) – DMS1090E (new text possible).
RTNLOAD	**Upwardly compatible changes:** • New messages: – DMS2502 (RC=4) – DMS2503 (RC=1,2,3,9). • Changed messages: – DMS066E (new text possible) – DMS1097 (new text possible) – DMS1098 (new text possible). • New message and return code for empty files: DMS2521E (RC=88). In VM/ESA 1.1.5 370 Feature, the system did not allow empty files. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 44 (Page 20 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
RUN	**Upwardly compatible changes:** • For the COBOL file type, the COBOL processor module called is COBOL, TESTCOB, or COBOL2. In VM/ESA 1.1.5 370 Feature, the COBOL processor module called was COBOL or TESTCOB.
SEGGEN	**Incompatible changes:** • The load map built for logical segments may contain additional information.
SEGMENT	**Incompatible changes:** • For LOAD and RESERVE, if you are trying to load or reserve a saved segment that is in the process of being saved, you get error message DMS1083E and the saved segment is not loaded or reserved. In VM/ESA 1.1.5 370 Feature, your load or reserve request was delayed until the save was complete. Then the saved segment was loaded or reserved. • For PURGE and RELEASE, if the segment is a logical saved segment containing minidisk information, the minidisks are released. In VM/ESA 1.1.5 370 Feature, the minidisks were not released.
SENDFILE	**Incompatible changes:** • The SENDFILE screen has changed. A new line is added for the new CLASS option. Also, if your screen size is 24 lines, some input fields may be in different locations on the screen. • When sending a note, if incorrect information is found in the new *class* field of the OPTIONS: line, the note is not sent. In VM/ESA 1.1.5 370 Feature, information found after the NOTEBOOK option field on the OPTIONS: line was ignored. • Added messages: – DMS149E – DMS653E – DMS1012E – DMS2501E. **Upwardly compatible changes:** • Supports a new option: CLASS. • Changed messages: – DMS081E. • TCP/IP domain names accepted as user IDs or as the resolution of nicknames. • New options to specify the transmission method: SMTP, MIME, UFTSYNC, UFTASYNC. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
SET (in general)	**Upwardly compatible changes:** • Supports new operands.
SET AUTODUMP	**Upwardly compatible changes:** • For some severe irrecoverable file system errors detected by CMS, an entire virtual machine dump is generated unless SET AUTODUMP OFF was specified. In VM/ESA 1.1.5 370 Feature, your AUTODUMP setting was always respected.
SET CMSPF	**Upwardly compatible changes:** • Accepts mixed DBCS data in line mode for the *pseudonym* parameter.
SET COMDIR	**Incompatible changes:** • The BOTH operand for RELOAD causes the user directory, the system directory, and the IPC names file, $QUEUES$ NAMES, to be loaded. In VM/ESA 1.1.5 370 Feature, BOTH caused only the user directory and the system directory to be loaded.
SET DOS	**Incompatible changes:** • Added message: DMS1101I. • Changed message: DMS333E.

Table 44 (Page 21 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
SET DOSPART	**Incompatible changes:** • Added message: DMS1101I. • Changed message: DMS333E.
SET FILEPOOL	**Upwardly compatible changes:** • The colon (:) on the *filepoolid* option is now optional.
SET LANGUAGE	**Upwardly compatible changes:** • New message for empty files: DMS1229E (RC=88). In VM/ESA 1.1.5 370 Feature, the system did not allow empty files.
SET RORESPECT	**Upwardly compatible changes:** • Used to ensure updates to files using XEDIT and/or COPYFILE in SFS will respect the read-only access mode of a directory.
SET SYSNAME	**Upwardly compatible changes:** • Supports new operand: CMSGUI
SET THRESHOLD	**Upwardly compatible changes:** • The colon (:) on the *filepoolid* option is now optional.
SET WMPF	**Upwardly compatible changes:** • Accepts mixed DBCS data in line mode for the *pseudonym* parameter.
SORT	**Upwardly compatible changes:** • New message and return code for empty files: DMS2521E (RC=88). In VM/ESA 1.1.5 370 Feature, the system did not allow empty files.
SSERV	**Upwardly compatible change:** • Supports a new option: TYPE.
STATE/STATEW	**Upwardly compatible change:** • STATE finds empty SFS files. In VM/ESA 1.1.5 370 Feature, the system did not allow empty files.
STORMAP	**Incompatible changes:** • The layout of the CMS nucleus is different in VM/ESA 2.4.0 so the data written using the SUM option is different.
SVCTRACE	**Upwardly compatible changes:** • Added messages and return codes for empty files.
SYNONYM	**Incompatible changes:** • Removed message: DMS032E. **Upwardly compatible changes:** • Supports a new parameter: *filetype*. In VM/ESA 1.1.5 370 Feature, file type had to be SYNONYM. • New message for empty files: DMS1229E (RC=88). In VM/ESA 1.1.5 370 Feature, the system did not allow empty files.

Table 44 (Page 22 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	**Changes**
TAPE	**Incompatible changes:** • Changed options: TRANSFER BUFF, TRANSFER IMMED. These options do nothing and may be removed in future releases. • TAPE loads itself as a nucleus extension. In VM/ESA 1.1.5 370 Feature, it was not loaded as a nucleus extension. **Upwardly compatible changes:** • Supports new options: EODTM, NOEODTM, 3490C, 3490B, COMP, NOCOMP. Also, NOMAP is a new option that replaces the NOPRINT option. NOPRINT is still functional, but is not supported. • Supports a new operand: QUERY. • Changed message: DMS115S (new possible text). • New messages added: DMS173W, DMS217R, DMS636W. • 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. • New message for Tape Library Dataserver support: DMS2147W.
TELL	**Incompatible changes:** • Added messages: – DMS149E – DMS653E – DMS1012E. **Upwardly compatible changes:** • Accepts a TCP/IP domain name as part of the destination information.
TXTLIB	**Incompatible changes:** • Changed message: DMS106S (text changed). **Upwardly compatible changes:** • Supports a new option: TYPE. • The GEN operand can be fully spelled out: GENERATE. GEN is still supported as an abbreviation. • New message: DMS056W (RC=4). • The total number of members in the TXTLIB cannot exceed 6000. In VM/ESA 1.1.5 370 Feature, it could not exceed 2000. • The total number of entry points in a member cannot exceed 4048. In VM/ESA 1.1.5 370 Feature, it could not exceed 255. • The total number of ALIAS linkage editor control statements in a member cannot exceed 64. In VM/ESA 1.1.5 370 Feature, this limit did not exist.
TYPE	**Upwardly compatible change:** • New message for empty files: DMS1229I (RC=0). In VM/ESA 1.1.5 370 Feature, the system did not allow empty files.

Table 44 (Page 23 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes

UPDATE	**Incompatible changes:**
	• With the STK option, comments on MACS records in the source input file are ignored. In VM/ESA 1.1.5 370 Feature, comments were stacked.
	For example, consider the following record:
	`TEXT MACS CMS120   * private MACLIB`
	In VM/ESA 2.4.0, '* private MACLIB' is ignored. In VM/ESA 1.1.5 370 Feature, '* private MACLIB' was stacked along with 'CMS120.'
	• Some return codes are changed:
	– When an update file listed in an AUX file is not found, return code 12 is always returned. In VM/ESA 1.1.5 370 Feature, if at least one update file listed in an AUX file was found and one was not found, return code 12 was returned; if no update files listed in an AUX file were found, return code 40 was returned.
	– When message DMS1259E (File pool *filepoolid* has run out of physical space in the storage group) is issued, return code 31 is returned. In VM/ESA 1.1.5 370 Feature, return code 40 was returned.
	– In other situations where return code 40 was returned for SFS-related errors, return code 100 is now returned.
	Upwardly compatible changes:
	• Supports new options: TYPE, NOTYPE.
	• New response during processing when you are using an empty file:
	`Updating fn1 with fn2 (empty file)`
	`    Update Log - - Page nn`
	• If the input file, update control file, or auxiliary control file is empty, you get new warning messages DMS1229I and DMS178I. If the base or control file is empty, you get new error message DMS1229E.
	• New message: DMS1213W.

VMFPLC2	**Incompatible changes:** • Changed message: DMS671E (text changed). **Upwardly compatible changes:** • The NOMAP option replaces the NOPRINT option. NOPRINT is still functional, but not supported. • The TYPE option replaces the TERM option. TERM is still functional, but is not supported. • New operand and options: QUERY, 3490C, 3490B, COMP, NOCOMP, EODTM, NOEODTM. • New messages for empty files: DMS636E, DMS173W. In VM/ESA 1.1.5 370 Feature, the system did not allow empty files. • Changed message: DMS115S (new text values for new support). • 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. • New message for Tape Library Dataserver support: DMS2147W.
VSCREEN WRITE	**Upwardly compatible changes:** • Accepts mixed DBCS data in line mode for the *text* parameter when *text* is preceded by the FIELD or DATA keyword.

Table 44 (Page 24 of 24). CMS Commands Changed since VM/ESA 1.1.5 370 Feature

<table>
<tr><th>Command</th><th>Changes</th></tr>
<tr><td>XEDIT</td><td><b>Incompatible changes:</b>
<ul>
<li>The file mode number of the autosave file may be different:
<pre>
         Edited   Autosave
         File's   File's
Version File Mode File Mode
Release Number   Number

1.1.5   anything   1

2.4.0   0   0
         any other 1

</pre></li>
<li>Added messages: DMS2154E, DMS2155E.</li>
<li>Supports empty files in an SFS directory. When you issue subcommands FILE or SAVE on an empty file not in an SFS directory, you get the new message DMS595E. In VM/ESA 1.1.5 370 Feature, you received message DMS559W.</li>
<li>Changed messages:
<ul>
<li>DMS559E (text changed)</li>
<li>DMS1255E (text changed)</li>
</ul></li>
<li>Added return codes: 50, 51.</li>
</ul>
<b>Upwardly compatible changes:</b>
<ul>
<li>New messages and return codes for DFSMS/VM support.</li>
<li>Added new option for BFS: NAMETYPE If NAMETYPE BFS is used, your profile must be in REXX and it is loaded as a REXX function.</li>
<li>Added new option for BFS: BFSLINE</li>
<li>New messages for BFS: DMS033E, DMS512E, DMS2105E, and DMS2134E.</li>
</ul></td></tr>
<tr><td>XMITMSG</td><td><b>Incompatible changes:</b>
<ul>
<li>The ERRMSG option overrides the HEADER option. You cannot specify ERRMSG with the NOHEADER, DISPLAY, or NODISPLAY options. In VM/ESA 1.1.5 370 Feature, ERRMSG would override the HEADER and DISPLAY options, and you could not specify ERRMSG with the NOHEADER or NODISPLAY options.</li>
<li>New messages: DMS813E, DMS814E.</li>
</ul></td></tr>
</table>

CMS File Pool Administration and Operator Commands

Table 45 lists the CMS file pool administration and operator commands that have changed since VM/ESA 1.1.5 370 Feature. See the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book for complete descriptions of file pool administration and operator commands.

Table 45 (Page 1 of 6). CMS File Pool Administration and Operator Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
AUDIT	**Upwardly compatible changes:** • New operands: *fn ft*, REPLACE. • Added messages: DMS024E, DMS1258E, DMS3253I, DMS3254E, DMS3255E. • Changed message: DMS3470W (new text possible).

Table 45 (Page 2 of 6). CMS File Pool Administration and Operator Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
CRR QUERY LOGTABLE	**Incompatible changes:** • In VM/ESA 2.4.0, when the TPN in the log name table entry contains unprintable characters, the hexadecimal translation of the TPN will follow on the next line in the display. In VM/ESA 1.1.5 370 Feature, the unprintable characters were displayed. • Other parts of the response have also changed.
CRR ERASE LU	**Upwardly compatible changes:** • Supports a new operand: *log_name_token*. • Supports a new option: LNTOKEN. • You may get new messages if you use the new parameter, *log_name_token*.
DELETE ADMINISTRATOR	**Incompatible changes:** • Added message: DMS149E. **Upwardly compatible changes:** • The colon (:) on the *filepoolid* option is now optional.
DELETE LOCK	**Incompatible changes:** • Added message: DMS149E.
DELETE PUBLIC	**Upwardly compatible changes:** • The colon (:) on the *filepoolid* option is now optional.
DELETE USER	**Incompatible changes:** • Added message: DMS149E. **Upwardly compatible changes:** • The colon (:) on the *filepoolid* option is now optional. • New options: DELAUTH | KEEPAUTH. • Added message: DMS2023E.
ENABLE	**Incompatible changes:** • For the GROUP parameter, new error messages DMS3074E and DMS3075E are issued if linked FBA minidisks are not aligned on 4KB boundaries. See “Aligning Minidisks on 4KB Boundaries for Minidisk Cache and for SFS [1.1.5]” on page 233. • Added messages: DMS3072W, DMS3073I, DMS3074E, DMS3074I, DMS3075E. **Upwardly compatible changes:** • Supports new operands: FUNCTION *function_name*. • You can also use the new administrator FILEPOOL ENABLE command.
ENROLL ADMINISTRATOR	**Incompatible changes:** • Added message: DMS149E. **Upwardly compatible changes:** • The colon (:) on the *filepoolid* option is now optional.
ENROLL PUBLIC	**Upwardly compatible changes:** • The colon (:) on the *filepoolid* option is now optional.

Table 45 (Page 3 of 6). CMS File Pool Administration and Operator Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
ENROLL USER	**Incompatible changes:** • Assigning user IDs beginning with "DFSMS" can cause unpredictable results if your installation is using the new DFSMS/VM support. • Added message: DMS149E. **Upwardly compatible changes:** • The colon (:) on the *filepoolid* option is now optional. • Supports new options: SFS | BFS, *userid*, *gname*. • Added messages for use with BFS: DMS1209E, DMS2023E, DMS2132E.
ERASE LUNAME	**Upwardly compatible changes:** • Supports new operands: ALL, LOGTABLE, FORCED.
FILEPOOL BACKUP	**Incompatible changes:** • You may notice a reduction in the size of the backup file. In VM/ESA 2.4.0, only the allocated blocks in the storage group are backed up. In VM/ESA 1.1.5 370 Feature, the entire storage group was backed up. • If a file space or storage group is locked because of an incomplete rename of an SFS-enrolled user ID, it cannot be backed up. (FILEPOOL RENAME is the new command that renames an SFS-enrolled user ID.) A new response indicates this. **Upwardly compatible changes:** • Added messages: DMS3501I, DMS3502I, DMS3503I, DMS3509I, DMS3518E, DMS3520E, DMS3523E, DMS3593E. • The colon (:) on the *filepoolid* option is now optional. • If your file pool is managed by DFSMS/VM, there are additional considerations. • Supports new BFS parameter: *bfsid*.
FILEPOOL CLEANUP	**Upwardly compatible changes:** • The colon (:) on the *filepoolid* option is now optional. • New message: DMS3518E.
FILEPOOL RELOAD	**Upwardly compatible changes:** • New message: DMS3455I.
FILEPOOL RESTORE	**Incompatible changes:** • New informational messages are issued for each FBA minidisk in the file pool that is not aligned on 4KB boundaries. See "Aligning Minidisks on 4KB Boundaries for Minidisk Cache and for SFS [1.1.5]" on page 233. **Upwardly compatible changes:** • A new error message, DMS3074E, is issued for FBA minidisks in the file pool that are not aligned on 4KB boundaries. • Added messages: DMS3501E, DMS3502E, DMS3503E, DMS3509I, DMS3518E, DMS3520E, DMS3523E, DMS3634W. • If your file pool is managed by DFSMS/VM, there are additional considerations. • The colon (:) on the *filepoolid* option is now optional.
FILEPOOL UNLOAD	**Upwardly compatible changes:** • New message: DMS3455I.
FILESERV BACKUP	**Incompatible changes:** • You must run in an XC or XA virtual machine because the server is not supported in a 370 virtual machine. In VM/ESA 1.1.5 370 Feature, the server was supported in a 370 virtual machine.

Table 45 (Page 4 of 6). CMS File Pool Administration and Operator Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
FILESERV CRRLOG	**Incompatible changes:** • You must run in an XC or XA virtual machine because the server is not supported in a 370 virtual machine. In VM/ESA 1.1.5 370 Feature, the server was supported in a 370 virtual machine.
FILESERV GENERATE	**Incompatible changes:** • New informational messages are issued for each FBA minidisk in the file pool that is not aligned on 4KB boundaries. See "Aligning Minidisks on 4KB Boundaries for Minidisk Cache and for SFS [1.1.5]" on page 233. • Added messages: DMS3074I, DMS3072W. **Upwardly compatible changes:** • The DDNAME=MDK*nnnnn* control statement of the input file can have a BLOCKS=*nnnnnnnn* parameter. • You must run in an XC or XA virtual machine because the server is not supported in a 370 virtual machine. In VM/ESA 1.1.5 370 Feature, the server was supported in a 370 virtual machine.
FILESERV LIST	**Incompatible changes:** • You must run in an XC or XA virtual machine because the server is not supported in a 370 virtual machine. In VM/ESA 1.1.5 370 Feature, the server was supported in a 370 virtual machine.
FILESERV LOG	**Incompatible changes:** • You must run in an XC or XA virtual machine because the server is not supported in a 370 virtual machine. In VM/ESA 1.1.5 370 Feature, the server was supported in a 370 virtual machine. • A new message, DMS3007R, is issued if data corruption is possible. The message asks if you want to continue or not. **Note:** If your installation has applied APAR VM52812, this message exists in your system.
FILESERV MINIDISK	**Incompatible changes:** • You must run in an XC or XA virtual machine because the server is not supported in a 370 virtual machine. In VM/ESA 1.1.5 370 Feature, the server was supported in a 370 virtual machine. • A new error message, DMS3074E, is issued for each FBA minidisk in the file pool that is not aligned on 4KB boundaries. See "Aligning Minidisks on 4KB Boundaries for Minidisk Cache and for SFS [1.1.5]" on page 233.
FILESERV MOVEUSER	**Incompatible changes:** • You must run in an XC or XA virtual machine because the server is not supported in a 370 virtual machine. In VM/ESA 1.1.5 370 Feature, the server was supported in a 370 virtual machine. **Upwardly compatible changes:** • Added message if DFSMS/VM is used: DMS3723E.
FILESERV REGENERATE	**Incompatible changes:** • You must run in an XC or XA virtual machine because the server is not supported in a 370 virtual machine. In VM/ESA 1.1.5 370 Feature, the server was supported in a 370 virtual machine.

Table 45 (Page 5 of 6). CMS File Pool Administration and Operator Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
FILESERV REORG	**Incompatible changes:** • You must run in an XC or XA virtual machine because the server is not supported in a 370 virtual machine. In VM/ESA 1.1.5 370 Feature, the server was supported in a 370 virtual machine. • A new response message, DMS3009R, is issued to recommend that you cancel the processing of the FILESERV REORG command if your control data backup is not current.
FILESERV START	**Incompatible changes:** • The start-up does not continue processing if the DMS3110E message is issued because of a SEGMENT ASSIGN failure. • You must run in an XC or XA virtual machine because the server is not supported in a 370 virtual machine. In VM/ESA 1.1.5 370 Feature, the server was supported in a 370 virtual machine. • A new informational message is issued for each FBA minidisk in the file pool that is not aligned on 4KB boundaries. See “Aligning Minidisks on 4KB Boundaries for Minidisk Cache and for SFS [1.1.5]” on page 233. • Added messages: DMS3074I, DMS3072W.
ITRACE	**Incompatible changes:** • The data is in a format compatible with the Dump Viewing Facility. In VM/ESA 1.1.5 370 Feature, the data was in a format compatible for use with IPCS.
MODIFY USER	**Incompatible changes:** • Added message: DMS149E. **Upwardly compatible changes:** • The colon (:) on the *filepoolid* option is now optional.
QUERY DISABLE	**Incompatible changes:** • A new response indicates when a file space or storage group is disabled because a FILEPOOL RENAME was attempted.
QUERY FILEPOOL - in general	**Incompatible changes:** • See individual QUERY FILEPOOL commands below. **Upwardly compatible changes:** • Supports new operands: AGENT, CATALOG, COUNTER, CRR, DISABLE, LOG, MINIDISK, OVERVIEW, REPORT, STORGRP.
QUERY FILEPOOL CONFLICT	**Incompatible changes:** • Added message: DMS149E. **Upwardly compatible changes:** • The Holder column may include new information. • The *Wait* column may include the wait state, *DFSMS/VM_Wait* if you are exploiting function of DFSMS/VM. • The *Wait* column may include the wait state, *Storage_Wait* if an agent is waiting for storage to be available on the server. • The *Lock* column may include the lock type, *File_Recall* or *Recall_Exit* if you are exploiting function of DFSMS/VM. • The colon (:) on the *filepoolid* option is now optional.

Table 45 (Page 6 of 6). CMS File Pool Administration and Operator Commands Changed since VM/ESA 1.1.5 370 Feature

Command	Changes
QUERY FILEPOOL STATUS	**Incompatible changes:** • The following Counter Information has been added to the output: – Precoordination Requests – Connect User Requests – Create External Objects Requests – Create File Requests – Query User Storage Group Requests – DMSSFSEX Exit Calls – DMSSFSEX Exit Time (msec) – File Recall Lock Conflicts – Recall Exit Lock Conflicts – Change DFSMS Related Attribute Requests – Migrate Requests – Recall Requests – Send DFSMS Data Requests – Recall DFSMS File Exit Calls – Recall DFSMS File Exit Time (msec) – Other DFSMS Exit Calls – Other DFSMS Exit Time (msec) – Rename Userid Requests – File Pool Control Backup Requests – Add Minidisk Requests – Query Disable Requests – Locks Denied Due to Timeout – Virtual Storage Reclaim Value – Open File CreateMig Requests • The *Wait* column may include the new wait state, *DFSMS/VM_Wait*, if you are exploiting function of DFSMS/VM. • The *Wait* column may include the new wait state, *Storage_Wait* if an agent is waiting for storage to be available on the server. • The colon (:) on the *filepoolid* option is now optional.
QUERY LIMITS	**Incompatible changes:** • File pool administration authority is no longer required. • Added message: DMS149E. **Upwardly compatible changes:** • New default added: *. In VM/ESA 1.1.5 370 Feature, you had to issue one of the following operands: FOR, *, or ALL. • The colon (:) on the *filepoolid* option is now optional.

XEDIT Subcommands

Table 46 lists the XEDIT subcommands that have changed since VM/ESA 1.1.5 370 Feature. For complete descriptions of XEDIT subcommands, see the *VM/ESA: XEDIT Command and Macro Reference*.

Table 46 (Page 1 of 4). XEDIT Subcommands Changed since VM/ESA 1.1.5 370 Feature

Subcommand	Changes
A	**Incompatible changes:** • If you issue the A subcommand in non-display mode: – In VM/ESA 1.1.5 370 Feature, XEDIT returns message DMS529E with RC=3. – In VM/ESA 2.4.0, the subcommand completes successfully with no message and RC=0.
ADD	**Incompatible changes:** • If you issue the ADD subcommand in non-display mode: – In VM/ESA 1.1.5 370 Feature, XEDIT returns message DMS529E with RC=3. – In VM/ESA 2.4.0, the subcommand completes successfully with no message and RC=0.
CMS	**Incompatible changes:** • If you issue a command with or without the CMS prefix and you are not in CMS subset mode and CMS returns a RC=-2 to XEDIT: – In VM/ESA 1.1.5 370 Feature, XEDIT returns message DMS512E with RC=-2 or RC=-1 – In VM/ESA 2.4.0, XEDIT returns message DMS514E with RC=-2. • If you issue a command with the CMS prefix and CMS finds the command name in the plist and returns an RC=-3 to XEDIT: – In VM/ESA 1.1.5 370 Feature, XEDIT returns message DMS513E with a return code other than -2 – In VM/ESA 2.4.0, XEDIT returns message DMS514E with the return code passed back from CMS. • If you issue a command without the CMS prefix and CMS finds the command name in the plist and returns an RC=-3 to XEDIT: – In VM/ESA 1.1.5 370 Feature, XEDIT returns message DMS542E with RC=-1 – In VM/ESA 2.4.0, XEDIT returns message DMS514E with the return code passed back from CMS. • New message added: DMS642E
COUNT	**Incompatible changes:** • The number of occurrences of the target string is returned regardless of the CP EMSG or XEDIT MSGMODE settings. In VM/ESA 1.1.5 370 Feature, CP EMSG and XEDIT MSGMODE settings were respected.
CP	**Upwardly compatible change:** • New message added: DMS642E
EXTRACT	**Upwardly compatible change:** • New operands added: BFSLine, EPName, NAMetype, PName, and GUI.

Table 46 (Page 2 of 4). XEDIT Subcommands Changed since VM/ESA 1.1.5 370 Feature

Subcommand	Changes
FILE	**Incompatible changes:** • When you enter FILE while editing an empty file, you get new message 595E and remain in the file. Then, if you enter FFILE: – If it is a VM/ESA 1.2.2 or later file pool file, the empty file is saved and you exit XEDIT – If it is: - A pre-VM/ESA 1.1.1 file pool file - A minidisk file - Any other type of file that does not support empty files (such as, MACLIB members, autosave files), you get new message 559E and remain in XEDIT. In VM/ESA 1.1.5 370 Feature, you received message 559W and exited the file. **Upwardly compatible changes:** • Changed messages: – 104S (text appended) – 105S (text appended). • Support for the new CMS SET RORESPECT command has been added. FILE will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O. • Support for BFS files has been added. • New messages for BFS support: 024E, 054E, 1184E, 2105E, 2120E, 2131E, and 2134E.
GET	**Upwardly compatible changes:** • Changed message: 104S (text appended). • Support for BFS files has been added. • New messages for BFS support: 033E, 054E, 512E, 2105E, 2131E, and 2134E.
I	**Incompatible changes:** • If you issue the I subcommand in non-display mode: – In VM/ESA 1.1.5 370 Feature, XEDIT returns message DMS529E with RC=3. – In VM/ESA 2.4.0, the subcommand completes successfully with no message and RC=0.
LOAD	**Upwardly compatible changes:** • Changed message: 104S (text appended). • Support for writing to BFS files has been added. • Added new options: NAMetype and BFSLine. • New messages for BFS support: 033E, 512E, 2105E, and 2134E.
MODIFY	**Upwardly compatible changes:** • New operands added: BFSLine, NAMetype, and PName.
PRESERVE	**Upwardly compatible changes:** • New operands added: BFSLine, NAMetype, and PName.

Table 46 (Page 3 of 4). XEDIT Subcommands Changed since VM/ESA 1.1.5 370 Feature

Subcommand	Changes
PUT	**Incompatible changes:** • Fails when using PUT to a packed file; RC 40 and message DMS743E are issued. In VM/ESA 1.1.5 370 Feature, it did not fail; RC 0 and no message were issued. • The temporary file created (when a *fileid* is not specified) for use by GET is erased when all rings are exited. In VM/ESA 1.1.5 370 Feature, the temporary file was erased when any one of the rings was exited. **Upwardly compatible changes:** • Changed message: 105S (text appended) • Support for the new CMS SET RORESPECT command has been added. PUT will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O. • Support for writing to BFS files has been added. • New return code for BFS support: 32. • New messages for BFS support: 033E, 054E, 512E, 1184E, 2105E, 2120E, 2131E, and 2134E.
PUTD	**Incompatible changes:** • Fails when using PUTD to a packed file; RC 40 and message DMS743E are issued. In VM/ESA 1.1.5 370 Feature, it did not fail; RC 0 and no message were issued. • The XEDIT temporary file created when no file name is specified can be used by a GET in the current or in any subsequent rings. In VM/ESA 1.1.5 370 Feature, the XEDIT temporary file was erased if any of the subsequent rings were exited. **Upwardly compatible changes:** • Changed message: 105S (text appended). • Support for the new CMS SET RORESPECT command has been added. PUTD will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O. • Support for writing to BFS files has been added. • New return code for BFS support: 32. • New messages for BFS support: 033E, 054E, 512E, 1184E, 2105E, 2120E, 2131E, and 2134E.
QUERY	**Upwardly compatible change:** • New operand added.
READ	**Incompatible changes:** • If the console stack is not empty and you issue READ, you receive a return code of 8. In VM/ESA 1.1.5 370 Feature, you received a return code of 0 and the subcommand was a no-op.
RENUM	**Upwardly compatible changes:** • Changed messages: – 104S (text appended) – 105S (text appended).

Table 46 (Page 4 of 4). XEDIT Subcommands Changed since VM/ESA 1.1.5 370 Feature

Subcommand	Changes
SAVE	**Incompatible changes:** • New message 595E is issued when you attempt to enter SAVE when you are editing an empty file. If you enter SSAVE, you do not get a message and you remain in the file. In VM/ESA 1.1.5 370 Feature, when you entered SAVE you received message 559W. When you entered SSAVE you also received message 559W. **Upwardly compatible changes:** • Changed messages: – 104S (text appended) – 105S (text appended). • Support for the new CMS SET RORESPECT command has been added. SAVE will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O. • Support for writing to BFS files has been added. • New return code for BFS support: 32. • New messages for BFS support: 024E, 054E, 512E, 1184E, 2105E, 2120E, 2131E, and 2134E.
SET	**Upwardly compatible change:** • New operand added.
SET AUTOSAVE	**Incompatible changes:** • When the AUTOSAVE is actually issued, you still get message 559W, but its text is different.
XEDIT	**Incompatible changes:** • See the entry for XEDIT in Table 44 on page 385 for details on the incompatibilities. • In update mode, the XEDIT subcommand no longer allows the same file in the ring multiple times. When a *fileid* match in the ring is found, that file becomes the current file.

CMS Macros

Table 47 lists the CMS preferred macros that have changed since VM/ESA 1.1.5 370 Feature. Table 48 on page 424 lists the CMS compatibility macros that have changed. Table 49 on page 424 lists the OS simulation macros that have changed. Table 50 on page 425 lists the simulated OS/MVS supervisor calls that have changed.

Preferred Macros

See the *VM/ESA: CMS Application Development Reference for Assembler* for complete descriptions of these macros.

Table 47 (Page 1 of 8). CMS Preferred Macros Changed since VM/ESA 1.1.5 370 Feature

Macro	Changes
ABNEXIT	**Upwardly compatible changes:** • New parameter: SYSTEM. • Upon completion, R1 contains the address of an area of storage mapped by the CMSSDWA DSECT, whose expansion can be obtained by calling the DMSSDWA macro. • If you are using ABNEXIT in an XC virtual machine, your exit routine always receives control in primary space address translation mode and must return control to CMS in primary space mode.

Table 47 (Page 2 of 8). CMS Preferred Macros Changed since VM/ESA 1.1.5 370 Feature

Macro	Changes
AMODESW	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370
ANCHOR	**Upwardly compatible changes:** • In an XC virtual machine, anchor support does not save and restore translation mode or access register.
APPLMSG	**Upwardly compatible changes:** • Added new parameter: CSECTA. • The APPLID parameter is optional when you specify TEXT or TEXTA unless you specify the SUB parameter with the *type* of DICT.
CMSCALL	**Upwardly compatible changes:** • Added new return codes: -0014, -0015.
CMSCVT	**Incompatible changes:** • The CVTSECT that is generated is larger. The CVTOSLVLs field are included. Also, the CVTDCB field now also indicates whether the CVTOSLVL fields are available to the application. • The CVTSECT that is generated has changed. The CVTNUCB field is reserved. In VM/ESA 1.1.5 370 Feature, it was the lowest storage address not in the nucleus and was initialized to X'20000'. • Field CVTTZ in CVTSECT is updated to reflect current timezone offset when x'2004' interrupt is processed by CMS. **Upwardly compatible changes:** • Added the fields: CVTECVT, CVTFLAG2.
CMSIUCV CONNECT	**Incompatible changes:** • New return code: X'5C'. This return code indicates that an invalid security tag was found in the communications directory. In VM/ESA 1.1.5 370 Feature, the security tag was not validated.
CMSIUCV RESOLVE	**Incompatible changes:** • New return code: X'5C'. This return code indicates that an invalid security tag was found in the communications directory. In VM/ESA 1.1.5 370 Feature, the security tag was not validated.
CMSLEVEL	**Incompatible changes:** • The CMS level has been frozen at X'0F' for CMS Level 12 (VM/ESA Version 2 Release 1.0) and later. Use the new DMSQEFL macro or the DMSQEFL CSL routine instead.
CONSOLE READ	**Upwardly compatible changes:** • Supports a new OPTIONS= option: *reg*.
CONSOLE WRITE	**Upwardly compatible changes:** • Supports a new OPTIONS= option: *reg*. • Supports a new parameter: BRKKEY=.
CSLENTRY	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370

Table 47 (Page 3 of 8). CMS Preferred Macros Changed since VM/ESA 1.1.5 370 Feature

Macro	Changes
CSLFPI	**Upwardly compatible changes:** mode and access registers at the time of the CSLFPI TYPE=CALL transfer if the application is executing in access-register mode (and the DMSSTATE macro has been called with ASCENV=ARM). • You may have other considerations if you use this macro in an XC virtual machine. See the *VM/ESA: CMS Application Development Reference for Assembler* for information. • Supports new parameter: MODE=NO370
CSLGETP	**Upwardly compatible changes:** • New data type codes, equated to the following new labels, can be returned by the TYPE operand: **Label** — **Data Type** CSLTUBIN — Unsigned binary number CSLTBIT — Bit string CSLTLEN — Unsigned binary length parameter for a previous parameter; can also indicate the number of rows for a table CSLTCHAR — Character string The existing CSLTFCHR label and data type code remains available for compatibility with previous releases only.
DIRBUFF	**Incompatible changes:** • Output buffer is restructured to support new function. **Upwardly compatible changes:** • The CONSTANTS record contains the following new field: DIRCEXT. • The FILE record contains the following new fields: DIRFUNQD, DIRFDAXD, DIRFDAXC, DIRFDAXI, DIRFCEND, DIRFLV13. • In the FILE record, the following field has changed: DIRFMIGR. • The FILEEXT record contains the following new fields: DIREMAXB, DIREDATB, DIRESYSB, DIREUNQD, DIRETDFM, DIREDAXD, DIREDAXC, DIREDAXI, DIREDRXD, DIREDRXC, DIREDRXI, DIRECDXD, DIRECDXC, DIRECDXI, DIREDCXD, DIREDCXC, DIREDCXI, DIRECEND, DIRER21L, DIRELV13. • In the FILEEXT record, the following fields have changed: DIREDRA1, DIREDRA2, DIREDDR3, DIREDRAS. • The SEARCHALL/SEARCHAUTH record contains the following new fields: DIRSMIGR, DIRSDAXD, DIRSDAXC, DIRSDAXI, DIRSCEND, DIRSLV13. • The ALIAS record contains the following new field: DIRAMIGR. • The AUTH record contains the following new fields: DIRUMIGR, DIRUCEND, DIRUR7L, DIRUR6L. • The LOCK record contains the following new fields: DIRLMIGR, DIRLCEND, DIRLR1L.
DMSSDWA	**Upwardly compatible changes:** • The following new fields have been added: SDWSA, SDWSALNT, SDWFPRS, SDWOPSWS, SDWEOPSW, SDWSOPSW, SDWPOPSW, SDWMOPSW, SDWIOPSW, SDWVSTR, SDWTXID, SDWFLAG1, SDWMCKAB, SDWPCKAB, SDWSVCAB, SDWFLAG2, SDWFSPRL, SDWMCIC, SDWIINFO, SDWPILC, SDWINTCD, SDWABNRC, SDWEXPTR, SDWENDBS. The following new fields have been added for the SDWXCS DSECT, which applies to XC virtual machines: SDWREGAR, SDWALET, SDWASIT, SDWFSA, SDWENDXC. Two new fields, SDWXCSP and SDWENDPS, have been added for the SDWEPTRS DSECT. The following contain length calculations: SDWPTLEN, SDWBSLEN, SDWXCLEN, SDWLNTH.

Table 47 (Page 4 of 8). CMS Preferred Macros Changed since VM/ESA 1.1.5 370 Feature

Macro	Changes
ENABLE	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370
EXSBUFF	**Incompatible changes:** • Output buffer is restructured to support new function. **Upwardly compatible changes:** • The FILE record contains the following new fields: EXSFSTO, EXSFMAXB, EXSFDATB, EXSFSYSB, EXSFMIGR, EXSFDRA1, EXSFDRA2, EXSFDRA3, EXSFDRAS, EXSFUNQD, EXSFTDFM, EXSFDIRL, EXSFDIRD, EXSFCONV, EXSFDAXD, EXSFDAXC, EXSFDAXI, EXSFDRXD, EXSFDRXC, EXSFDRXI, EXSFCDXD, EXSFCDXC, EXSFCDXI, EXSFDCXD, EXSFDCXC, EXSFDCXI, EXSF2000, EXSFRES, EXSFLV13, EXSFTDFL, EXSFCDTL. • The DIR record contains the following new fields: EXSDDRA1, EXSDDRA2, EXSDDRA3, EXSDDRAS, EXSDUNQD, EXSDDCXD, EXSDDCXC, EXSDDCXI, EXSDCDXD, EXSDCDXC, EXSDCDXI, EXSD2000, EXSDRES, EXSDLV13, EXSDTDFL, EXSDMCDL.
EXTUAREA	**Upwardly compatible changes:** • The following new fields were added: EXTUARS, EXTUAR0, EXTUAR1, EXTUAR2, EXTUAR3, EXTUAR4, EXTUAR5, EXTUAR6, EXTUAR7, EXTUAR8, EXTUAR9, EXTUAR10, EXTUAR11, EXTUAR12, EXTUAR13, EXTUAR14, EXTUAR15.
FPERROR	**Incompatible changes:** • A new field, FPEDETFP, has been inserted in the FPERROR buffer. This new field is used for DFSMS/VM support.
FSOPEN	**Upwardly compatible changes:** • New return codes: 49, 50, 51. • RC=99 now also indicates “File is migrated and DFSMS/VM is not enabled.”
FSREAD	**Upwardly compatible changes:** • New return codes for new support: 49, 50, 51. • RC=99 now also indicates “File is migrated and DFSMS/VM is not enabled.”
FSSTATE	**Upwardly compatible changes:** • 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*nn*, 1=20*nn*, where *nn* is the 2-digit year). In VM/ESA 1.1.5 370 Feature, this bit was not used.
FSTD	**Upwardly compatible changes:** • The FSTFLAGS section contains the new FSTCNTRY field, which is a bit that indicates the century (first two digits of the year) the file was last written or updated (0=19*nn*, 1=20*nn*, where *nn* is the 2-digit year).
FSWRITE	**Incompatible changes:** • An FSWRITE to a file in an SFS directory that is in a VM/ESA 1.2.1 or later file pool fails when CMS detects that the file space is full. Your application receives an error return code (RC=13). In VM/ESA 1.1.5 370 Feature, your application did not receive a file space full condition, and you could continue writing. • Changed return code when writing to SFS files: 13. • If the caller's buffer overlaps internal CMS file system buffers, an error code 2 will be returned, or message 1307T will be issued and the system will be abnormally terminated, if it is necessary to avoid file corruption. **Upwardly compatible changes:** • RC=99 now also indicates “File is migrated and DFSMS/VM is not enabled.”

Table 47 (Page 5 of 8). CMS Preferred Macros Changed since VM/ESA 1.1.5 370 Feature

Macro	Changes
HNDEXT	**Upwardly compatible changes:** • You may have changes to make if you use this macro in an XC virtual machine. See the *VM/ESA: CMS Application Development Reference for Assembler* for information.
HNDINT	**Upwardly compatible changes:** • You may have to make changes if you use this macro in an XC virtual machine. See the *VM/ESA: CMS Application Development Reference for Assembler* for information.
HNDIO	**Upwardly compatible changes:** • Supports new parameter: PERSIST • You may have to make changes if you use this macro in an XC virtual machine. See the *VM/ESA: CMS Application Development Reference for Assembler* for information.
HNDSVC	**Upwardly compatible changes:** • You may have changes to make if you use this macro in an XC virtual machine. See the *VM/ESA: CMS Application Development Reference for Assembler* for information.
LINERD	**Incompatible changes:** • In VM/ESA 1.1.5 370 Feature, if a LINERD was issued while running in a batch machine, I/O would be done from the console. In VM/ESA 2.4.0, it is done from the card reader. **Upwardly compatible changes:** • LINERD is supported in a batch machine.
LINEWRT	**Incompatible changes:** • In VM/ESA 2.4.0, if FULLSCREEN is ON and you issue the LINEWRT NOCR=YES macro, a new field will be defined immediately after the data written and the cursor will be positioned at the start of the new field. In VM/ESA 1.1.5 370 Feature, the cursor was positioned directly after the data and the data field filled the entire line of the virtual screen. If your application issues the LINEWRTE MOCR=YES macro, followed by a LINERD, the entire field that is modified is returned to your application. In VM/ESA 1.1.5 370 Feature, if you entered data where the cursor was positioned from the LINEWRT, then the LINEWRT data, in addition to the input data, would be returned. In VM/ESA 2.4.0, only the input data will be returned to your application, because the cursor is in a separate field from the LINEWRT data. Using the LINEWRT and LINERD macros this way is now consistent with issuing LINERD with the PROMPT option.
NUCEXT SET	**Upwardly compatible changes:** • Added new parameter: PERM.
PARSECMD	**Incompatible changes:** • A new return code, 30, is issued if the CALLTYP parameter is not included.
RDTAPE	**Incompatible changes:** • New return code: 0 • Changed: return code: 1—also means that the specified device is incapable of writing in the specified format, in addition to invalid function or parameter list. In VM/ESA 1.1.5 370 Feature, this return code only meant invalid function or parameter list. • The *buffer* specifies the address of the buffer into which a block is to be read. In VM/ESA 1.1.5 370 Feature, this was a record. • The *length* specifies the length of the buffer into which the block is to be read. In VM/ESA 1.1.5 370 Feature, *length* indicated the largest record to be read. **Upwardly compatible changes:** • New value for MODE parameter: 3490C, 3490B, XF, COMP, NOCOMP, 1838K, 9625K, 91600, and 9800.

Table 47 (Page 6 of 8). CMS Preferred Macros Changed since VM/ESA 1.1.5 370 Feature

Macro	Changes
SEGMENT	**Incompatible changes:** • If you are trying to load or find a saved segment that is in the process of being saved, you get return code 44 and the saved segment is not loaded or found. In VM/ESA 1.1.5 370 Feature, your load or find request was delayed until the save was complete, and then the saved segment was loaded or found. • New return code for SEGMENT LOAD: 76. • New return code for SEGMENT FIND: 41, 76.
TAPECTL	**Incompatible changes:** • BLKBUFF=RDBLKID—gives the address of an 8 byte buffer in which CMS returns 2 block IDs. In VM/ESA 1.1.5 370 Feature, this returned the address of a locations that contained 0. The LOCBLK, RDBLKID, and BLKBUFF are useful only with a tape which is recorded in a recording format that contains block IDs. These formats include 3490 (compacted and basic), 3480 (compacted and basic). See the *VM/ESA: CMS Application Development Reference for Assembler* for details. • New return code: 0 • Changed return code: 6. In VM/ESA 2.4.0, this means that the tape is write protected. In VM/ESA 1.1.5 370 Feature,it meant that the tape is file protected. **Upwardly compatible changes:** • New values for the MODE= parameter: 3490C, 3490B, XF, COMP, NOCOMP, 1838K, 96250, 91600, 9800. • 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.
TAPESL	**Incompatible changes:** • Changed return code: 100—means that the virtual tape device is not attached or the device is not capable of writing in the recording format specified by the MODE parameter. This is in addition to meaning that a tape I/O error occurred. In VM/ESA 1.1.5 370 Feature, it meant only that a tape I/O error occurred. **Upwardly compatible changes:** • New values for MODE parameter: 3490C, 3490B, XF, COMP, NOCOMP, 1838K, 96250, 91600, 9800.

Table 47 (Page 7 of 8). CMS Preferred Macros Changed since VM/ESA 1.1.5 370 Feature

<table>
<tr><th>Macro</th><th>Changes</th></tr>
<tr><td>TVISECT</td><td>

Incompatible changes:

- TVIMODE is a synonym for TVIRFMT. The TVIMODE field may contain different values in VM/ESA 2.4.0. Also, some of the VM/ESA 1.1.5 370 Feature hex codes have been removed.

  Note: TVIMODE is a synonym for TVIRFMT.

<table>
<tr><th>VM/ESA
1.1.5
Hex Codes</th><th>Tape Format</th><th>VM/ESA
2.4.0
Hex Codes</th></tr>
<tr><td>X'10'</td><td>3480 Basic</td><td>X'10' (Unchanged)</td></tr>
<tr><td>X'13'</td><td></td><td>Removed</td></tr>
<tr><td></td><td>3490 Basic</td><td>X'20' (New)</td></tr>
<tr><td>X'23'</td><td></td><td>Removed</td></tr>
<tr><td>X'2B'</td><td></td><td>Removed</td></tr>
<tr><td></td><td>3490 Comp</td><td>X'30' (New)</td></tr>
<tr><td>X'33'</td><td></td><td>Removed</td></tr>
<tr><td>X'3B'</td><td></td><td>Removed</td></tr>
<tr><td></td><td>3590 Basic</td><td>X'40' (New)</td></tr>
<tr><td></td><td>9346</td><td>X'50' (New)</td></tr>
<tr><td>X'53'</td><td></td><td>Removed</td></tr>
<tr><td>X'60'</td><td>3480 Comp</td><td>X'60' (Unchanged)</td></tr>
<tr><td>X'63'</td><td></td><td>Removed</td></tr>
<tr><td>X'6B'</td><td></td><td>Removed</td></tr>
<tr><td></td><td>3590 Comp</td><td>X'70' (New)</td></tr>
<tr><td>X'73'</td><td></td><td>Removed</td></tr>
<tr><td>X'7B'</td><td></td><td>Removed</td></tr>
<tr><td>X'93'</td><td></td><td>Removed</td></tr>
<tr><td>X'A3'</td><td></td><td>Removed</td></tr>
<tr><td>X'AB'</td><td></td><td>Removed</td></tr>
<tr><td>X'B3'</td><td></td><td>Removed</td></tr>
<tr><td>X'BB'</td><td></td><td>Removed</td></tr>
<tr><td>X'C3'</td><td>PE</td><td>X'C3' (Unchanged)</td></tr>
<tr><td>X'CB'</td><td>NRZI</td><td>X'CB' (Unchanged)</td></tr>
<tr><td>X'D3'</td><td>GCR</td><td>X'D3' (Unchanged)</td></tr>
</table>

Upwardly compatible changes:

- New fields are supported: TVIFLAGS, TVIRECFM, TVILRECL, TVIBLKSI.

</td></tr>
<tr><td>USERSAVE</td><td>

Upwardly compatible changes:

- The USEAR field has been added to indicate whether the caller was in AR mode when CMSCALL was issued.

</td></tr>
</table>

Table 47 (Page 8 of 8). CMS Preferred Macros Changed since VM/ESA 1.1.5 370 Feature

Macro	Changes
WRTAPE	**Incompatible changes:** • New return code: 0 • Changed: return code: 1—also means that the specified device is incapable of writing in the specified format, in addition to invalid function or parameter list. In VM/ESA 1.1.5 370 Feature, this return code only meant invalid function or parameter list. • Changed return code: 2—in VM/ESA 2.4.0 means in end of volume area. In VM/ESA 1.1.5 370 Feature, it meant end of file or end of tape. • Changed return code: 6—In VM/ESA 2.4.0 means that the tape is write protected. In VM/ESA 1.1.5 370 Feature, it meant that the tape is file protected. • The *buffer* parameter specifies the address of the buffer into which a block is to be read. In VM/ESA 1.1.5 370 Feature, this was a record. **Upwardly compatible changes:** • New values for MODE parameter: 3490C, 3490B, XF, COMP, NOCOMP, 1838K, 96250, 91600, 9800.

Compatibility Macros

See *VM/ESA: Planning and Administration* for information about the DEFNUC macro.

Table 48. CMS Compatibility Macros Changed since VM/ESA 1.1.5 370 Feature

Macro	Changes
DEFNUC	**Incompatible changes:** • The CYLADDR= parameter supports up-to-10 digits. In VM/ESA 1.1.5 370 Feature, it supported up to five. • The format of the default CMS IPL heading (used when the VERSION= parameter is specified without a value in DEFNUC) has changed to include the version. Also, the date in the default IPL heading is presented in ISO format (*yyyy-mm-dd*). **Upwardly compatible changes:** • Supports new parameters: • New message: DMS2104R.

OS Simulation Macros and Supervisor Calls

See the *VM/ESA: CMS Application Development Guide for Assembler* for information about the OS macros and supervisor calls that CMS simulates.

Table 49 (Page 1 of 2). OS Simulation Macros Changed since VM/ESA 1.1.5 370 Feature

Macro	Changes
CLOSE	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
DCB	**Upwardly compatible changes:** • 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.

Table 49 (Page 2 of 2). OS Simulation Macros Changed since VM/ESA 1.1.5 370 Feature

Macro	Changes
GET	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.
OPEN	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
PUT	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.
TIME	**Upwardly compatible changes:** • The 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.

Table 50. OS Simulation Supervisor Calls Changed since VM/ESA 1.1.5 370 Feature

SVC	Changes
SVC 19 (OPEN)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
SVC 20 (CLOSE)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.

Common Programming Interface (CPI) Communications Routines

Table 51 lists the Systems Application Architecture (SAA) CPI Communications routines that have changed between VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0. Table 52 on page 426 lists the changes to VM/ESA's implementation of and extensions to CPI Communications routines.

SAA CPI Communications Routines

Table 51. SAA CPI Communications Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Explanation
CMINIT	Initialize_Conversation **Upwardly compatible changes:** If the specified *sym_dest_name* has a matching entry in side information but that entry does not include a value for the :tpn. tag, then the name provided as the *sym_dest_name* is used to set the *TP_name* conversation characteristic.

VM/ESA-Specific CPI Communications Routines

Table 52. VM/ESA-Specific CPI Communications Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Explanation
XCIDRM	Identify_Resource_Manager **Upwardly compatible changes:** • Supports a new value for the *resource_manager_type* parameter: XC_SYSTEM.
XCSUE	Signal_User_Event **Upwardly compatible changes:** • You can use this routine even if all CPI Communications conversations and resources have been terminated. In VM/ESA 1.1.5 370 Feature, this was not allowed. **Note:** XCSUE is new for you unless your installation has applied APAR VM44911.
XCWOE	Wait_on_Event **Upwardly compatible changes:** • *info_input_length* parameter is now called *event_info_length*. • *console_input_buffer* parameter is now called *event_buffer*. • There is a new value for the *event_type* parameter, XC_USER_EVENT. • You can use this routine even if all CPI Communications conversations and resources have been terminated. In VM/ESA 1.1.5 370 Feature, this was not allowed.

Preferred Routines (CSL Routines)

“Changes in Common Reason Codes” describes changes made to the common reason codes used by many I/O CSL routines.

Table 53 lists the CSL routines that have changed since VM/ESA 1.1.5 370 Feature. See the *VM/ESA: CMS Application Development Reference* for complete descriptions of the routines in this table.

Changes in Common Reason Codes

If you initiate work that takes up all of server resource allocated to that work unit, any subsequent request that requires that resource fails and reason code 40000 is returned. For this same situation in VM/ESA 1.1.5 370 Feature, reason code 71000 was returned.

An example where this could happen is if you issued several DMSOPDIR SEARCHALL requests on directories with many files. If an internal server resource limit is reached for this work unit, the next request that requires that resource fails with reason code 40000. To solve this problem, some of the open directories can be closed, thereby freeing up resources in that work unit, or the request could be performed in a different work unit.

For a complete description of all the common reason codes, see the *VM/ESA: CMS Application Development Reference*.

Table 53 (Page 1 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSCATTR	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • New parameters: *requestid*, *filemode*. • New reason codes: 90472, 90601, 90611. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSCLBLK	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • Reason code 50500 has a different meaning. **Upwardly compatible changes:** • New reason codes when using BFS: 10220, 65400. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330, 90492, 90495.
DMSCLCAT	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.
DMSCLDIR	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • Reason code 50500 has a different meaning.
DMSCLOSE	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • Reason code 50500 has a different meaning. • Removed reason code: 95777. **Upwardly compatible changes:** • DMSCLOSE now works on minidisk files. • New reason codes: 90130, 90472, 90617, 97600. • Supports a new parameter *requestid*. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330, 90495.

Table 53 (Page 2 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSCOMM	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • Reason code 50500 has a different meaning.
DMSCRALI	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • Supports a new option: UNRESOLVED. • New parameter: *filemode*. • New reason codes: 90601, 90611, 61620, 98700. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSCRDIR	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • New parameter: *filemode*. • New reason codes: 90601, 90611. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSCRLOC	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • New parameter: *filemode*. • New reason codes: 90601, 90611. • You receive the new reason code, 66300, if your application tries to create a session lock on behalf of a virtual machine other than the one it is running in. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New reason codes when using BFS: 65400, 69200, and 69300. • Supports new BFS parameter: *bfsid*.

Table 53 (Page 3 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSDELOC	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • New parameter: *filemode*. • New reason codes: 90601, 90611. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid*.
DMSDEUSR	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • New parameters: DELAUTH, KEEPAUTH, and *length4*. • New reason code: 98700.
DMSDIRAT	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • New function that can affect DMSDIRAT includes: – Attempting to use DMSDIRAT to change the attribute to DIRCONTROL on a directory containing files in DFSMS/VM migrated status will cause DMSDIRAT to fail. • Supports new parameters: *requestid*, *filemode*. • New reason codes: 65700, 90472, 90601, 90611. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSDISFS	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • You can receive the new reason code, 02000, if you try to disable the file space because another user has an exclusive lock on the storage group. **Upwardly compatible changes:** • New function that can affect DMSDISFS includes: – DFSMS/VM support. While a SHARE lock is in effect, files cannot be migrated, expired, or recalled. • Supports new parameters: *owner*, *length3*. • New reason code: 30000.

Table 53 (Page 4 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSDISSG	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • New function that can affect DMSDISSG includes: – DFSMS/VM support. While a SHARE or EXCLUSIVE lock is in effect, files cannot be migrated, expired, or recalled. • New parameters: *owner*, *length4*. • Reason code 30000 has an additional meaning related to the new *owner* parameter. The additional meaning is that the *owner* is not authorized to perform the request.
DMSENAFS	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • New parameters: *function*, *length4*. • New reason codes: 02070, 67000, 90320, 92410, 98700.
DMSENASG	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • You can receive a new reason code, 66400, if FBA minidisks are not aligned on 4KB boundaries. See “Aligning Minidisks on 4KB Boundaries for Minidisk Cache and for SFS [1.1.5]” on page 233.
DMSENUSR	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • New function that can affect DMSENUSR includes: – DFSMS/VM support. Do not assign user IDs beginning with “DFSMS”; these are reserved for use by DFSMS/VM. • New reason codes when using BFS: 10210, 10240, 69000, 90300, 90320, and 98700. • Supports new options: SFS|BFS, *userid*, *gname*. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE, *length7*. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90330, 90495.

Table 53 (Page 5 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSERASE	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • Reason code 50500 has a different meaning. **Upwardly compatible changes:** • DMSERASE now works on minidisk files. • New parameters: DATAONLY, ENTIRE, *filemode*. • New reason codes: 65400, 90351, 90601, 90614, 90685, 90700, 98700. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid*.
DMSERP	**Upwardly compatible changes:** • Supports new functions: GETENV, RES:*envir*, REP:*envir*, EXT:*envir*. • If you try to use DMSERP to extract file data information about erased aliases, external objects, or aliases whose authority has been revoked, you will get an error return code because these file pool objects do not contain any data. • New information names have been added, including names for Year 2000 support (FILE_DATE_CENTURY, ACT_FILE_DATE_CENTRY (note that U is omitted), and YEAR2000_SUPPORT). • New return codes: 125, 134. • Changed return codes: 112, 117, 124.
DMSEXIDI	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • Reason code 50500 has a different meaning. • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.1.5 370 Feature, it was returned as 8 characters. **Upwardly compatible changes:** • Supports external objects. • New parameters: *dra_values*, *filemode*, *update_date*, *update_time*, *create_date*, *create_time*. • New value for *status* parameter: 6. • New reason codes: 90601, 90611. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameters: *bfsid*, *filesystemtype* • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *last_change_date* and *create_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330.

Table 53 (Page 6 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSEXIFI	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • Reason code 50500 has a different meaning. • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.1.5 370 Feature, it was returned as 8 characters. **Upwardly compatible changes:** • DMSEXIFI now works on minidisks. • New parameters: *max_blocks*, *system_blocks*, *data_blocks*, *uniqueid*, *migrated*, *dra_values*, *filemode*, *fmode*, *real_fmode*, *update_date*, *update_time*. • New values for *status* parameter: 6, 7, 8. • New reason code: 90700. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameters: *bfsid*, *filesystemtype* • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date*, *dateref*, *create_date*, and *last_change_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330.
DMSEXIST	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • Reason code 50500 has a different meaning. **Upwardly compatible changes:** • External objects are supported. • DMSEXIST now works on minidisk files. • New parameters: *filemode*, *uniqueid_fpid*, UNIQUEID. • New reason codes: 90445, 90476, 90481, 90486, 90601, 90700. • Changed reason code: 90300. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid*

(continued on next page)

Table 53 (Page 7 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSEXIST - continued	**Upwardly compatible changes (continued):** • The following offsets have changed in the FILE data record (and the record length has increased to 436 bytes):

OFFSET	Field Name	Description of Change
1 (X'1')	*file_system_type*	Reserved
39 (X'27')	*file_mode*	Added
43 (X'2B')	*real_file_mode*	Added
60 (X'3C')	*owner_userid*	Returns blanks for minidisk files
68 (X'44')	*status*	Returns '6' for external objects, '7' for minidisk files, '8'for OS disk file
72 (X'48')	*dec_dateref*	Returns '0' for external objects and minidisk files
76 (X'4C')	*dateref*	Returns a blank for external objects and minidisk files
84 (X'54')	*dec_cr_date*	Returns '0' for minidisk files
88 (X'58')	*dec_cr_date*	Returns '0' for minidisk files
92 (X'5C')	*cr_date*	Returns a blank for minidisk files
100 (X'64')	*cr_time*	Returns a blank for minidisk files
108 (X'6C')	*max_blocks*	Added
112 (X'70')	*data_blocks*	Added
116 (X'74')	*system_blocks*	Added
120 (X'78')	*migrated*	Added
121 (X'79')	*dra_values*	Added
145 (X'91')	*unique_id*	Added
161 (X'A1')	*dirname_len*	Added
162 (X'A2')	*dirname*	Added
315 (X'13B')	*dec_update_date*	Added
318 (X'13E')	Reserved	Added
319 (X'13F')	*dec_update_time*	Added
322 (X'142')	Reserved	Added
323 (X'143')	*update_date*	Added
331 (X'14B')	*update_time*	Added
339 (X'153')	*dec_date_ext*	Added
343 (X'157')	*date_ext*	Added
353 (X'161')	*iso_date_ext*	Added
363 (X'16B')	*dec_dateref_ext*	Added
367 (X'16F')	*dateref_ext*	Added
377 (X'179')	*iso_dateref_ext*	Added
387 (X'183')	*dec_cr_date_ext*	Added
391 (X'187')	*cr_date_ext*	Added
401 (X'191')	*iso_cr_date_ext*	Added
411 (X'19B')	*dec_last_change_date_ext*	Added
415 (X'19F')	*last_change_date_ext*	Added
425 (X'1A9')	*iso_last_change_date_ext*	Added
435 (X'1B3')	Reserved	Added

(continued on next page)

Table 53 (Page 8 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSEXIST - continued</td><td><b>Upwardly compatible changes (continued):</b>
• The following offsets have changed in the DIRECTORY data record (and the record length has increased to 308 bytes):
<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>1 (X'1')</td><td><i>file_system_type</i></td><td>Reserved</td></tr>
<tr><td>166 (X'A6')</td><td><i>dra_values</i></td><td>Added</td></tr>
<tr><td>190 (X'BE')</td><td><i>unique_id</i></td><td>Added</td></tr>
<tr><td>206 (X'CE')</td><td><i>dec_update_date</i></td><td>Added</td></tr>
<tr><td>209 (X'D1')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>210 (X'D2')</td><td><i>dec_update_time</i></td><td>Added</td></tr>
<tr><td>213 (X'D5')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>214 (X'D6')</td><td><i>update_date</i></td><td>Added</td></tr>
<tr><td>222 (X'DE')</td><td><i>update_time</i></td><td>Added</td></tr>
<tr><td>230 (X'E6')</td><td><i>dec_cr_date</i></td><td>Added</td></tr>
<tr><td>233 (X'E9')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>234 (X'EA')</td><td><i>dec_cr_time</i></td><td>Added</td></tr>
<tr><td>237 (X'ED')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>238 (X'EE')</td><td><i>cr_date</i></td><td>Added</td></tr>
<tr><td>246 (X'F6')</td><td><i>cr_time</i></td><td>Added</td></tr>
<tr><td>254 (X'FE')</td><td><i>dec_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>258 (X'102')</td><td><i>last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>268 (X'10C')</td><td><i>iso_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>278 (X'116')</td><td><i>dec_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>282 (X'11A')</td><td><i>cr_date_ext</i></td><td>Added</td></tr>
<tr><td>292 (X'124')</td><td><i>iso_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>302 (X'12E')</td><td>Reserved</td><td>Added</td></tr>
</table></td></tr>
<tr><td>DMSFILEC</td><td><b>Incompatible changes:</b>
• You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation.
• You can receive a new reason code, 90415, if the length specified for the <i>wuerror</i> parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.
• Reason code 50500 has a different meaning.
<b>Upwardly compatible changes:</b>
• DMSFILEC now works on minidisks.
• Supports new parameters: <i>filemode1</i>, <i>filemode2</i>, <i>buffer</i>, <i>length6</i>.
• New reason codes: 64600, 64700, 64800, 64900, 65200, 65300, 65400, 65500, 66100, 90160, 90610, 90623, 90700.
• The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.</td></tr>
</table>

Table 53 (Page 9 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSGETDA	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.1.5 370 Feature, it was returned as 8 characters. **Upwardly compatible changes:** • Supports a new parameter: *migrated*. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330.
DMSGETDD	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.
DMSGETDF	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.1.5 370 Feature, it was returned as 8 characters. **Upwardly compatible changes:** • DMSGETDF now works on minidisks. • Supports external objects. • Supports new parameters: *uniqueid*, *filemode*, *migrated*. • The *status* parameter can now return a value of “6” for external objects and “7” for minidisk files. • New reason code: 90472. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330.

Table 53 (Page 10 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSGETDI	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • Supports external objects. • DMSGETDI now works on minidisks. • The following offsets have changed in the FILE data record (and the record length has increased to 112 bytes):

OFFSET	Field Name	Description of Change
23 (X'17')	*filemode*	Reserved in VM/ESA 1.1.5 370 Feature
43 (X'2B')	*migrated*	Reserved in VM/ESA 1.1.5 370 Feature
60 (X'3C')	*owner_userid*	Returns user ID of owner for external objects and blanks if a minidisk file.
68 (X'44')	*status*	Returns '6' for external objects and '7' for minidisk files
69 (X'45')	*rd_auth*	For minidisk files, always indicates read access
70 (X'46')	*wr_auth*	For minidisk files, indicates if minidisk accessed read/write
71 (X'47')	*extern_protect*	Returns a blank for minidisk file
72 (X'48')	*unique_id*	Added
88 (X'58')	*dec_date_ext*	Added
92 (X'5C')	*date_ext*	Added
102 (X'66')	*iso_date_ext*	Added

(continued on next page)

Table 53 (Page 11 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSGETDI - continued	**Upwardly compatible changes (continued):** • The following offsets have changed in the FILEEXT data record (and the record length has increased to 284 bytes):

OFFSET	Field Name	Description of Change
1 (X'1')	*file_system_type*	Reserved
43 (X'2B')	*migrated*	Added
68 (X'44')	*status*	Returns '6' for external objects
108 (X'6C')	*max_blocks*	Reserved in VM/ESA 1.1.5 370 Feature
112 (X'70')	*data_blocks*	Added
116 (X'74')	*system_blocks*	Added
121 (X'79')	*dra_values*	Added
145 (X'91')	*uniqueid*	Added
161 (X'A1')	*dec_update_date*	Added
164 (X'A4')	Reserved	Added
165 (X'A5')	*dec_update_time*	Added
168 (X'A8')	Reserved	Added
169 (X'A9')	*update_date*	Added
177 (X'B1')	*update_time*	Added
185 (X'B9')	*dec_date_ext*	Added
189 (X'BD')	*date_ext*	Added
199 (X'C7')	*iso_date_ext*	Added
209 (X'D1')	*dec_dateref_ext*	Added
213 (X'D5')	*dateref_ext*	Added
223 (X'DF')	*iso_dateref_ext*	Added
233 (X'E9')	*dec_cr_date_ext*	Added
237 (X'ED')	*cr_date_ext*	Added
247 (X'F7')	*iso_cr_date_ext*	Added
257 (X'101')	*dec_last_change_date_ext*	Added
261 (X'105')	*last_change_date_ext*	Added
271 (X'10F')	*iso_last_change_date_ext*	Added
281 (X'119')	Reserved	Added

(continued on next page)

Table 53 (Page 12 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine | **Changes**

DMSGETDI - continued

Upwardly compatible changes (continued):

- The following offsets have changed in the SEARCHALL and SEARCHAUTH data records (and the record length has increased to 252 bytes):

OFFSET	Field Name	Description of Change
43 (X'2B')	*migrated*	Reserved in VM/ESA 1.1.5 370 Feature
68 (X'44')	*status*	Returns '6' for external objects
226 (X'E2')	Reserved	Added
228 (X'E4')	*dec_date_ext*	Added
232 (X'E8')	*date_ext*	Added
242 (X'F2')	*iso_date_ext*	Added

- The following offset has changed in the ALIAS data record:

OFFSET	Field Name	Description of Change
22 (X'16')	*migrated*	Reserved in VM/ESA 1.1.5 370 Feature

- The following offsets have changed in the AUTH data record:

OFFSET	Field Name	Description of Change
21 (X'15')	*status*	Returns '6' for external objects
38 (X'26')	*migrated*	Reserved in VM/ESA 1.1.5 370 Feature

- The following offsets have changed in the LOCK data record:

OFFSET	Field Name	Description of Change
1 (X'1')	*file_system_type*	Reserved
33 (X'21')	*migrated*	Reserved in VM/ESA 1.1.5 370 Feature

DMSGETDK

Incompatible changes:

- You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.

Upwardly compatible changes:

- Has a new parameter: *migrated*.
- Supports new BFS parameter: *filesystemtype*

DMSGETDL

Incompatible changes:

- You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.

DMSGETDS

Incompatible changes:

- You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.
- When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.1.5 370 Feature, it was returned as 8 characters.

Upwardly compatible changes:

- Has a new parameter: *migrated*.
- New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*.
- The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.
- New reason codes: 90310, 90320, 90330.

Table 53 (Page 13 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSGETDT	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • Supports external objects. • Has a new parameter: *migrated*. • The *status* parameter can return a value of “6” for external objects.
DMSGETDX	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.1.5 370 Feature, it was returned as 8 characters. **Upwardly compatible changes** • Supports external objects. • Suuports new parameters: *max_blocks*, *system_blocks*, *data_blocks*, *uniqueid*, *migrated*, *dra_values*, *update_date*, *update_time*. • The *status* parameter can return a value of “6” for external objects. • Supports new BFS parameter: *filesystemtype* • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*. • The *date*, *dateref*, *create_date*, and *update_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330.
DMSGETWU	**Upwardly compatible changes:** • New parameters: *userid*, *sectoken*, *slength*, *acctdata*. • New reason codes: 90410, 90540, 96100.
DMSGRANT	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • Reason code 50500 has a different meaning. **Upwardly compatible changes:** • New parameter: *filemode*. • New reason codes: 90601, 90611. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 53 (Page 14 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSOPBLK	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • Reason code removed: 95500. • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.1.5 370 Feature, it was returned as 8 characters. **Upwardly compatible changes:** • Supports a new option: RESOLVE. • Supports new parameters: ALLOWEMPTY, *filemode*, NORECALL, CREATEMIG, *unique_id*, *userid*, *dateref*, *bfsid* • Changed parameter: *length2*—includes the length of ALLOWEMPTY if specified. • Supports new reason codes: 30000, 50500, 63800 64600, 64700, 64800, 64900, 65200, 65300, 65400, 66100, 90601, 90611, 98700. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New reason codes when using BFS: 10220, 65400, 98700. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date*, *create_date*, and *dateref* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSOPCAT	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • New reason code: 72000. (You may receive this new reason code if another user is in the process of renaming the user ID for a file space.) **Upwardly compatible changes:** • Supports a new parameter: FILEATTR. • Changed parameter: *length2*—includes the length of READ, WRITE, and FILEATTR if specified. • New reason code: 50103. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid* • Supports new BFS options: FILEATTR SFS|BFS, READEXT • New reason codes when using BFS: 90300, 90350.
DMSOPDIR	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • You can open a given directory multiple times. • DMSOPDIR now works on minidisks. • New parameters: *filemode*, EXT, NOEXT, *buffersize*, *number*. • New reason codes: 30000, 63800, 81058, 90601, 90700, 98700. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid* • New reason code when using BFS: 65400.

Table 53 (Page 15 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSOPEN	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes** • DMSOPEN now works on minidisks. • You can open CMS files multiple times for READ and once for output (NEW, WRITE, REPLACE). • New parameters: ALLOWEMPTY, *lrecl*, *requestid*, *filemode*. • New reason codes: 44035, 50500, 50600, 64600, 64700, 64800, 64900, 65200, 65300, 65400, 65900, 66100, 81058, 90306, 90307, 90472, 90494, 90496, 90498, 90499, 90602, 90603, 90604, 90606, 90620, 90621, 90680, 90700, 96100, 96200, 96300. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid* • New reason code when using BFS: 10220. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *create_date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSPURWU	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.
DMSQEFL	**Upwardly compatible changes:** • Supports new values for *cp_level* and *cms_level*
DMSQLIMA	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • File pool administration authority no longer required. **Upwardly compatible changes:** • Supports new BFS parameter: *filesystemtype*
DMSQLIMD	**Upwardly compatible changes:** • Supports new BFS parameter: *filesystemtype*
DMSQLIMU	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • File pool administration authority no longer required. **Upwardly compatible changes:** • Supports new BFS parameter: *filesystemtype*
DMSQSFSL	**Upwardly compatible changes:** • Supports new value for *server_level*
DMSQUSG	**Upwardly compatible changes:** • New parameter: *requestid*. • New reason code 90472.

Table 53 (Page 16 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSRDBLK</td><td><b>Incompatible changes:</b>
• You can receive a new reason code, 90415, if the length specified for the <i>wuerror</i> parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.</td></tr>
<tr><td>DMSRDCAT</td><td><b>Incompatible changes:</b>
• You can receive a new reason code, 90415, if the length specified for the <i>wuerror</i> parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.
• New reason code: 72100. (You may receive this new reason code if another user is in the process of renaming the user ID for a file space.)
<b>Upwardly compatible changes:</b>
• When the new FILEATTR parameter is used for DMSOPCAT, the following new information is returned for different objects:
<b>Directory</b>
a DIRCAT record (CATTYPE=D) followed by an OBJECTCAT (CATTYPE=O) record for each file, alias, external object, and subdirectory in the directory.
<b>File Space</b>
a SPACECAT record (CATTYPE=S) followed by a series of DIRCAT / OBJECTCAT groups for each directory in the file space
<b>Storage Groups</b>
a series (in no particular order) of SPACECAT / DIRCAT / OBJECTCAT groups for each file space in the storage group (or for all storage groups).
• New ACAT record for unresolved aliases.
• New EOCAT record for external objects.
• New catalogs for BFS support: NAMECAT, NOVCAT.
• New OBJECTCAT record for BFS object.
• SPACECAT record — the four reserved CHAR(8) fields at the end of the record (following the STORAGEGROUP field) have been restructured for BFS support as follows:
<table>
<tr><th>Field Name</th><th>Field Type/Description</th></tr>
<tr><td>HIGHINO</td><td>INTEGER(4) High OBJECTCAT INO value</td></tr>
<tr><td>HIGHNID</td><td>INTEGER(4) High NAMECAT NID value</td></tr>
<tr><td>FLAGS</td><td>CHAR(1) File space attributes</td></tr>
<tr><td>RESERVED01</td><td>CHAR(7) Reserved</td></tr>
<tr><td>RESERVED02</td><td>CHAR(8) Reserved</td></tr>
<tr><td>RESERVED03</td><td>CHAR(8) Reserved</td></tr>
</table>
• DIRCAT record — the second bit of the DIRATTS field indicates whether the directory is data space eligible or not. In VM/ESA 1.1.5 370 Feature, the second bit had no meaning. There are also new bit settings in the DIRATTS field for BFS support.
(continued on next page)</td></tr>
</table>

Table 53 (Page 17 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSRDCAT - continued</td><td><b>Upwardly compatible changes (continued):</b>
• OBJECTCAT record for SFS:
– New bit settings in TYPE and FILEFLAGS fields for BFS support.
– New bit settings in FILEFLAGS field to support 4-digit years (century setting for DATE and DATEREF fields).
– The STATUS field can have a new value of K to indicate an external object.
– The fields following the BSCID field are:
<table>
<tr><th>Field Name</th><th>Field Type/Description</th></tr>
<tr><td></td><td>CHAR(8) reserved</td></tr>
<tr><td>CHGDATE_CENTURY</td><td>CHAR(1) Century byte for LAST_CHANGE_DATE</td></tr>
<tr><td>LAST_CHANGE_DATE</td><td>CHAR(3) UTC date of last change</td></tr>
<tr><td></td><td>CHAR(1) reserved</td></tr>
<tr><td>LAST_CHANGE_TIME</td><td>CHAR(3) UTC time of last change</td></tr>
<tr><td>DRA1</td><td>CHAR(8) DRA1</td></tr>
<tr><td>DRA2</td><td>CHAR(8) DRA2</td></tr>
<tr><td>DRA3</td><td>CHAR(8) DRA3</td></tr>
<tr><td></td><td>INTEGER(4) reserved</td></tr>
<tr><td>CREATIONDATE_CENTURY</td><td>CHAR(1) Century byte for CREATIONDATE</td></tr>
<tr><td>CREATIONDATE</td><td>CHAR(3) UTC date of file creation</td></tr>
<tr><td></td><td>CHAR(1) reserved</td></tr>
<tr><td>CREATIONTIME</td><td>CHAR(3) UTC time of file creation</td></tr>
<tr><td>DATEREF</td><td>CHAR(3) UTC date of last reference</td></tr>
</table>
• FQFN record — new H type record for open intent READEXT.
• AUTHCAT record — new G type record for open intent READEXT.
• New reason codes: 44900, 78112.</td></tr>
<tr><td>DMSREAD</td><td><b>Incompatible changes:</b>
• You can receive a new reason code, 90415, if the length specified for the <i>wuerror</i> parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.
<b>Upwardly compatible changes:</b>
• DMSREAD now works on minidisks.
• New parameter: <i>requestid</i>.
• New reason codes: 90472, 97600.</td></tr>
<tr><td>DMSRELBK</td><td><b>Incompatible changes:</b>
• You can receive a new reason code, 90415, if the length specified for the <i>wuerror</i> parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.</td></tr>
</table>

Table 53 (Page 18 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSRELOC	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • Supports external objects. • New parameters: *requestid*, *filemode*. • New reason codes: 90472, 90601, 90611. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSRENAM	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • Reason code 50500 has a different meaning. **Upwardly compatible changes:** • Supports external objects. • Supports minidisks. • New parameters: *requestid*, *filemode1*, *filemode2*. • New reason codes: 61610, 90472, 90535, 90601, 90614. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSRETWU	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. **Upwardly compatible changes:** • If you used the new parameters of the DMSGETWU routine to associate work units with a user ID, security token, or user accounting data, the work unit returned is no longer associated with these items. In VM/ESA 1.1.5 370 Feature, you could not associate work units with user IDs, security tokens, or user accounting data.
DMSREVOK	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid. • Reason code 50500 has a different meaning. **Upwardly compatible changes:** • New parameter: *filemode*. • New reason codes: 90601, 90611. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 53 (Page 19 of 19). CSL Routines Changed since VM/ESA 1.1.5 370 Feature

Routine	Changes
DMSROLLB	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.
DMSUDATA	**Upwardly compatible changes:** • New parameters: *requestid*, *filemode*. • Add reason codes: 44000, 90472, 90601, 90611. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSWRACC	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.
DMSWRBLK	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.
DMSWRCAT	**Incompatible changes:** • You can receive a new reason code, 90415, if the length specified for the *wuerror* parameter is less than 12 bytes. In VM/ESA 1.1.5 370 Feature, a length less than 12 might have resulted in data being overlaid.
DMSWRITE	**Upwardly compatible changes:** • DMSWRITE now works on minidisks. • New parameter: *requestid*. • New reason codes: 90415, 90472, 97600.
DMSWUERR	**Upwardly compatible changes:** • New parameter: *failing_filepoolid*. • The *error_reascode_info* parameter may include information about the 50500 reason code. Also, the information may contain a CSL error reason code returned from DFSMS/VM in the event of a recall failure.

CSLCNTRL File Changes

The CSL control file has new and revised statements, which are summarized in Table 54.

Table 54. CSLCNTRL File Changes since VM/ESA 1.1.5 370 Feature

Line	Explanation
ROUTINE	**Upwardly compatible changes:** • Supports new options: PATH, PROTECT, SUBGROUP, COPY, MP, CSECT, FILETYPE, COPYTYPE, MAP.
ALIAS	New statement
TXTLIB	New statement
TEXT	New statement
INCLUDE	New statement

Compatibility Routines

Table 55 lists the CMS compatibility routines that have changed between VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0. See the *VM/ESA: CMS Application Development Guide for Assembler* for more information about the routines in this table.

Table 55. CMS Compatibility Routines Changed since VM/ESA 1.1.5 370 Feature

CMS Routine	Explanation
DMSTVS	**Upwardly compatible changes:** • Supports new LIBSRV plist parameter. • 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.

CMS Pipelines

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.
- 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.

  Note: User-written applications that are sensitive to these changes may require alterations. Published externals (such as PIPGFTXT) have **not** been changed.
- 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. Some specific enhancements that have occurred since VM/ESA 2.1.0 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 *VM/ESA: CMS Pipelines Reference* can be found in the *CMS/TSO Pipelines: Author's Edition*, which is included with the VM/ESA 2.4.0 library.

CMS Messages

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 H, “CMS Pipelines Message Cross-Reference [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]” on page 669.

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/ESA.

The following CMS messages do not exist in VM/ESA 2.4.0:

DMS035E
DMS045E
DMS401S
DMS633E
DMS683E
DMS833E
DMS834E
DMS835W
DMS836E
DMS837E
DMS838E
DMS839W
DMS840I
DMS841I
DMS3205I
DMS3275R
DMS3503I

The following CMS messages have changed since VM/ESA 1.1.5 370 Feature.

Note: All of the SFS error messages that used to have the prefix DMSSDM now have the prefix DMS*xxx*, where *xxx* is the three-character identifier of the module that discovered the error.

Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CMS messages.

DMS002E
DMS003E
DMS005E
DMS007E
DMS017E
DMS023E
DMS024E
DMS029E
DMS033E
DMS036E
DMS037E
DMS042E
DMS054E
DMS056E
DMS062E
DMS065E
DMS069I
DMS081E
DMS099E
DMS106S
DMS107S
DMS115S
DMS132S
DMS149E
DMS165I
DMS173E
DMS178I
DMS240E
DMS333E
DMS344E
DMS358E
DMS360E
DMS361E
DMS363R
DMS364I
DMS365I
DMS366R
DMS516E
DMS531E
DMS616W
DMS618E
DMS621E
DMS622E
DMS624I
DMS636W
DMS639E
DMS647E
DMS651E
DMS660W
DMS771E
DMS988E
DMS1087E
DMS1088E
DMS1090E
DMS1091E
DMS1094E
DMS1096E
DMS1097E
DMS1098E
DMS1104R
DMS1105E
DMS1108I
DMS1111E
DMS1127I
DMS1129W
DMS1131E
DMS1137E
DMS1138E
DMS1139E
DMS1146E
DMS1151E
DMS1153E
DMS1155E
DMS1162E
DMS1164E
DMS1174E
DMS1176E
DMS1184E
DMS1205I
DMS1209E
DMS1215E
DMS1227E
DMS1229E
DMS1235E
DMS1236E
DMS1237E
DMS1239E
DMS1254E
DMS1258E
DMS1259E
DMS1292S
DMS1297I
DMS1307T
DMS1311E
DMS2008E
DMS2010E
DMS2012E
DMS2013E
DMS2023E
DMS2031E
DMS2035W
DMS2040E
DMS2047I
DMS2514E
DMS2516E
DMS2523E
DMS2534I
DMS3088R
DMS3110E
DMS3208E
DMS3277E
DMS3284E
DMS3288E
DMS3299R
DMS3300E
DMS3310E
DMS3311E
DMS3312E
DMS3313E
DMS3328E
DMS3337E
DMS3338E
DMS3339E
DMS3340I
DMS3342E
DMS3345R
DMS3381I
DMS3382I
DMS3470W
DMS3494R
DMS3501I
DMS3502I
DMS3511E
DMS3514E
DMS3515E
DMS3518E

DMS3519E
DMS3520E
DMS3521E
DMS3522E
DMS3524E
DMS3526E
DMS3527E
DMS3528E
DMS3529E
DMS3530E
DMS3531E
DMS3532E
DMS3557E
DMS3561E
DMS3562E
DMS3585E
DMS3593I
DMS3594R
DMS3603E
DMS3603W
DMS3608E
DMS3616W
DMS3617I
DMS3618W
DMS3622E
DMS3623E
DMS3624W
DMS3628E
DMS3631W
DMS3632W
DMS3711E
DMS3926E

REXX/VM Changes

This section identifies the REXX/VM externals that have changed since VM/ESA 1.1.5 370 Feature. It contains the following subsections:

- "REXX/VM General Concepts"
- "REXX/VM Instructions"
- "REXX/VM Functions" on page 449
- "External Functions" on page 450
- "REXX/VM Messages" on page 450

REXX/VM General Concepts

REXX/VM now has support for binary literal strings. This is an incompatible change. For example, `'01001'b` is a binary string. Any program with a variable "b" that was abutted at the end of a literal string is now considered a binary string. Previously, this resulted in the literal string being concatenated to the value of the variable "b." **Your programs may no longer work properly. Therefore, a tool called REXXCHEK is available to help you convert your programs.** Enter REXXCHEK on the CMS command line for details.

REXX/VM Instructions

Table 56 lists the REXX/VM instructions that have changed since VM/ESA 1.1.5 370 Feature. For complete descriptions of these instructions, see the *VM/ESA: REXX/VM Reference*.

Table 56 (Page 1 of 2). REXX/VM Instructions Changed since VM/ESA 1.1.5 370 Feature

Instruction	Changes
CALL	**Upwardly compatible changes:** • Added conditions: OFF NOTREADY, ON NOTREADY.
DROP	**Upwardly compatible changes:** • Variable support has been improved.
OPTIONS	**Incompatible changes:** • An error message is issued if you use OPTIONS without any expression. In VM/ESA 1.1.5 370 Feature, it was valid to use OPTIONS without an expression.
PARSE	**Upwardly compatible changes:** • Added a new variant: PARSE LINEIN.
PROCEDURE	**Upwardly compatible changes:** • Variable support has been improved.

Table 56 (Page 2 of 2). REXX/VM Instructions Changed since VM/ESA 1.1.5 370 Feature

Instruction	Changes
SIGNAL	**Incompatible changes:** • In VM/ESA 1.1.5 370 Feature, the label derived from VALUE *expression* or *expression* was translated to uppercase before searching for the label. In VM/ESA 2.4.0, uppercasing of *expression* is no longer done. Because all labels in REXX are stored in uppercase, an *expression* that resolves to lowercase or mixed case characters will not be found. So, make sure that you uppercase *expression*. *labelname* is still uppercased in VM/ESA 2.4.0, so its use is not affected by this change. • Quotes around the condition of a SIGNAL ON or OFF are not supported. (For example, SIGNAL ON "SYNTAX".) In VM/ESA 1.1.5 370 Feature, this was supported, even though the syntax documented in the *VM/ESA: Procedures Language VM/REXX Reference* did not show it. • If you use a literal string for the label on a SIGNAL statement, the characters need to be in uppercase. This has always been in the language definition, but there was an error in the processing that allowed mixed case specification. **Upwardly compatible changes:** • Added conditions: OFF NOTREADY, ON NOTREADY.

REXX/VM Functions

Table 57 lists the REXX/VM functions that have changed since VM/ESA 1.1.5 370 Feature. For complete descriptions of these functions, see the *VM/ESA: REXX/VM Reference*.

Note: Two new functions have been added to REXX that may conflict with functions you have created for yourself. These are B2X and X2B, which translate binary to hexadecimal and hexadecimal to binary. They return a string in character format.

Table 57. REXX/VM Functions Changed since VM/ESA 1.1.5 370 Feature

Function	Changes
DATE	**Incompatible changes:** • Does not support a null variable for an option. In VM/ESA 1.1.5 370 Feature, a null variable for an option was supported. **Upwardly compatible changes:** • New parameters allow you to specify a date to be converted to a different format. • New parameters: *output_separator_char*, *input_separator_char*.
TIME	**Incompatible changes:** • Does not support a null variable for an option. In VM/ESA 1.1.5 370 Feature, a null variable for an option was supported.
VALUE	**Upwardly compatible changes:** • Variable support has been improved.

External Functions

Table 58 lists external functions which can be used by REXX/VM that have changed since VM/ESA 1.1.5 370 Feature. For complete descriptions of these functions, see the *VM/ESA: REXX/VM Reference*.

Table 58. External Functions Changed since VM/ESA 1.1.5 370 Feature

Function	Changes
CMSFLAG	**Upwardly compatible changes:** • New values can be specified for *flag*: 370, XA, XC, YEAR2000.
DIAG DIAGRC	**Incompatible changes:** • Subcodes N and Q are not supported. **Upwardly compatible changes:** • New DIAGNOSE codes are supported: X'F8', X'210', X'270'.

REXX/VM Messages

The following REXX/VM messages have been added or changed since VM/ESA 1.1.5 370 Feature. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of REXX/VM messages.

219E
459E
460E
462E
469E
481E

GCS Changes

This section identifies the GCS externals that have changed since VM/ESA 1.1.5 370 Feature. It contains the following subsections:

- “GCS Commands and Macros”
- “GCS Messages” on page 452

GCS Commands and Macros

Table 59 lists the GCS commands and macros that have changed since VM/ESA 1.1.5 370 Feature. See *VM/ESA: Group Control System* for complete descriptions of all GCS commands and macros.

Table 59 (Page 1 of 2). GCS Commands and Macros Changed since VM/ESA 1.1.5 370 Feature

Command/Macro	Changes
CVT macro	**Upwardly compatible changes:** • A new field is added to the OS communications vector table as simulated by GCS: CVTTZ. • Added the fields: CVTECVT, CVTFLAG2.
GCSLEVEL macro	**Upwardly compatible changes:** • Additional equates for release levels.

Table 59 (Page 2 of 2). GCS Commands and Macros Changed since VM/ESA 1.1.5 370 Feature

Command/Macro	Changes
GETMAIN macro	**Upwardly compatible changes:** • New parameters: LOC= RES|BELOW|ANY.
ITRACE command	**Upwardly compatible changes:** • New operands: DSP, EXT, FRE, GET, I/O, PRG, SIO, SSS, SP, SVC, SYN.
LOAD macro	**Upwardly compatible changes:** • Directed load into common storage by applications that are in supervisor state is supported. Use the ADDR parameter to specify where a module or table is to be loaded.
QUERY command -- in general	**Incompatible changes:** • See individual QUERY commands below. **Upwardly compatible changes:** • New operands: IPOLL, AUTHUSER, DUMPVM, TRACETAB, ADDRESS, COMMON.
QUERY ADDRESS command	**Upwardly compatible changes:** • Operand *variable* changed. It now supports either *name* or *address*.

QUERY DISK command	**Incompatible changes:**
	• The format of the response has changed. `'BLKSZ'` replaces `'BLKSIZE'` in the response header. The `'FILES'` field is eight bytes; the `'CYL'` field is five bytes. In VM/ESA 1.1.5 370 Feature, the `'FILES'` field was nine bytes; the `'CYL'` field was four bytes.
	The VM/ESA 2.4.0 response looks like this:
	`LABEL  VDEV M  STAT   CYL TYPE BLKSZ   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL`
	`label  vdev m   R/W nnnnn type nnnn nnnnnnnn nnnnnnnnnn-%% nnnnnnnnnn nnnnnnnnnn`
	The VM/ESA 1.1.5 370 Feature response looked like this:
	`LABEL  CUU M  STAT  CYL TYPE BLKSIZE   FILES  BLKS USED-(%) BLKS LEFT  BLK TOTAL`
	`label  cuu m   R/W nnnn type nnnn  nnnnnnnnn nnnnnnnnnn-%% nnnnnnnnnn nnnnnnnnnn`

QUERY GCSLEVEL command	**Incompatible changes:** • The format of the response has changed to include the version and the service level: `VM/ESA Version 2 Release 4.0, Service Level 0`
QUERY MODDATE command	**Incompatible changes:** • A full 4-digit year is now returned in the date field of the response instead of a 2-digit year.
SET command	**Upwardly compatible changes:** • New parameter: IPOLL.
TIME macro	**Upwardly compatible changes:** • The 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.

GCS Messages

The following GCS messages have changed:

019E
134I

You should not need to make any changes because of these differences. See *VM/ESA: System Messages and Codes* for complete descriptions of all GCS messages.

AVS Changes

This section identifies the AVS externals that have changed since VM/ESA 1.1.5 370 Feature. It contains the following subsection:

- "AVS Messages"

AVS Messages

The following AVS messages have changed between VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0:

301I
302I
303I

See *VM/ESA: System Messages and Codes* for complete descriptions of AVS messages.

VMSES/E Changes

This section identifies the VMSES/E externals that have changed since VM/ESA 1.1.5 370 Feature. It contains the following subsection:

- "VMSES/E Messages"

VMSES/E Messages

The following VM/ESA 1.1.5 370 Feature message does not exist in VM/ESA 2.4.0:

VMF2709E

The following VMSES/E messages have changed since VM/ESA 1.1.5 370 Feature. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of VMSES/E messages.

VMF002E
VMF066E
VMF071E
VMF395E
VMF1015E
VMF1016E
VMF1059E
VMF1068E
VMF1444I
VMF1507E
VMF1821E
VMF1832E
VMF1851E
VMF1851I
VMF1861E
VMF1861I
VMF1861W
VMF1868W
VMF1905E
VMF1908E
VMF1944E
VMF1953W
VMF2066E
VMF2111E
VMF2112I
VMF2114R
VMF2118I
VMF2119I
VMF2120W
VMF2152E
VMF2162E
VMF2173I
VMF2181E
VMF2200E
VMF2201E
VMF2202E

VMF2206E
VMF2206W
VMF2215E
VMF2225E
VMF2228E
VMF2242E
VMF2500E
VMF2507I
VMF2509I
VMF2704E
VMF2740E
VMF2760I
VMF2780E
VMF2786E
VMF2830E
VMF2831E
VMF2835E
VMF2866I
VMF2867W
VMF2930E

Programmable Operator Facility Changes

This section identifies the Programmable Operator Facility externals that have changed since VM/ESA 1.1.5 370 Feature. It contains the following subsection:

- “Programmable Operator Facility Commands and Routing Table Statements”

Also see “Summary of Service Exec Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]” on page 250.

Programmable Operator Facility Commands and Routing Table Statements

Table 60 lists the Programmable Operator Facility commands and routing table statements that have changed between VM/ESA 1.1.5 370 Feature and VM/ESA 2.4.0. See *VM/ESA: Planning and Administration* for complete descriptions of programmable operator facility commands and routing table statements.

Table 60. Programmable Operator Facility Commands and Routing Table Statements Changed since VM/ESA 1.1.5 370 Feature

Programmable Operator Command or Statement	**Explanation**
SET LOGGING	**Upwardly compatible changes:** • New operands: UPCASE, LOWCASE.
QUERY LOGGING	**Upwardly compatible changes:** • A response has changed to show {UPCASE | LOWCASE} when LOGGING is ON or ALL.
LOGGING	**Upwardly compatible changes:** • New parameters: UPCASE, LOWCASE.
MSGLIMIT	**Upwardly compatible changes:** • New routing table configuration entry statement.

Chapter 18. Compatibility Tables for Converting from VM/ESA 1.2.0

This chapter identifies the VM/ESA externals that have changed between VM/ESA 1.2.0 and VM/ESA 2.4.0. It contains the following major sections:

- "CP Changes"
- "CMS Changes" on page 469
- "REXX/VM Changes" on page 499
- "GCS Changes" on page 500
- "Dump Viewing Facility Changes" on page 501
- "VMSES/E Changes" on page 501
- "Programmable Operator Facility Changes" on page 502

Use the information provided in this chapter to determine if you need to make any changes in the way you use these functions. See "Terminology and Symbols" on page 3 for the meaning of compatibility terms used in the tables.

CP Changes

This section identifies the CP externals that have changed since VM/ESA 1.2.0. It contains the following subsections:

- "CP Commands"
- "SYSTEM CONFIG Statements" on page 463
- "User Directory Control Statements" on page 464
- "CP Utilities" on page 465
- "CP DIAGNOSE Codes" on page 466
- "CP Macros" on page 467
- "CP Messages" on page 468

CP Commands

Table 61 lists the CP commands that have changed since VM/ESA 1.2.0. Refer to the *VM/ESA: CP Command and Utility Reference* for complete descriptions of CP commands.

Table 61 (Page 1 of 9). CP Commands Changed since VM/ESA 1.2.0

Command	Changes
AUTOLOG	**Incompatible changes:** • Users defined with the LBYONLY operand in the password field of the USER directory control statement cannot be logged on by the AUTOLOG command. **Upwardly compatible changes:** • Supports the System Console.
CPLISTFILE	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses include 4-digit years for FULLDATE and ISODATE operands.

Table 61 (Page 2 of 9). CP Commands Changed since VM/ESA 1.2.0

Command	Changes
CPU	**Incompatible changes:** • On nested CPU commands, multiple *cpuaddr* entries for virtual CPUs are valid on only one of the commands. VM/ESA 1.2.0 supported multiple CPUs on multiple nested CPU commands. • New message: HCP1460E.
DEFINE (in general)	**Upwardly compatible changes:** • Supports new operands: MSGPROC, VFB-512. • Additional messages: HCP045E, HCP260E, HCP1014E, HCP2800E, HCP2801E, HCP2802E, HCP2803E, HCP2804I, HCP2806E, HCP2811E. • See DEFINE commands below.
DEFINE CRYPTO	**Upwardly compatible changes:** • New response. • New message: HCP1716E.
DETACH (in general)	**Upwardly compatible changes:** • Supports new operand: MSGPROC. • Additional messages: HCP260E, HCP2805E, HCP2807E.
DISCONN	**Upwardly compatible changes:** • Supports the System Console.
DISPLAY Linkage Stack	**Upwardly compatible changes:** • Response indicates the called-space identification (CSID) if the linkage-stack entry type is a program-call state entry with a called-space ID.
DISPLAY Registers	**Upwardly compatible changes:** • Supports new operand: FPC. • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New response when FPC operand is used. • New messages: HCP6153E, HCP6154E.
DUMP Linkage Stack	**Upwardly compatible changes:** • Response indicates the called-space identification (CSID) if the linkage-stack entry type is a program-call state entry with a called-space ID.
INDICATE (in general)	**Upwardly compatible changes:** • Supports new operands: NSS, SPACES. • New message: HCP1001E.
INDICATE USER	**Upwardly compatible changes:** • Supports new operand: EXPANDED.

Table 61 (Page 3 of 9). CP Commands Changed since VM/ESA 1.2.0

<table>
<tr><th>Command</th><th>Changes</th></tr>
<tr><td>IPL</td><td><b>Incompatible changes:</b>
• Changed message: HCP2895I.
• If your IPL tries to load a named saved system that is in the process of being saved, you get error message HCP044E and the named saved system is not loaded. In VM/ESA 1.2.0, your IPL request was delayed until the save was complete. Then the named saved system was loaded.
• A new response is issued if tracing is active when a preferred guest is IPLed. Tracing must be turned off and the guest reIPLed.
• In VM/ESA 1.2.0, when you IPL CMS with the PARM operand, CMS initialization attaches a fence of 8 bytes of X'FF's to the end of the PARM data before passing it to the SYSPROF EXEC. In VM/ESA 2.4.0, 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.</td></tr>
<tr><td>LINK</td><td><b>Upwardly compatible changes:</b>
• Changed messages:
– HCP091E (possible text change).</td></tr>
<tr><td>LOADBUF</td><td><b>Incompatible changes:</b>
• The function of the INDEX parameter has changed:
<code>In the New Release:</code>
<code>-------------------</code><table>
<tr><th><i>INDEX Parameter</i></th><th><i>What happens in VM/ESA 2.4.0</i></th></tr>
<tr><td>&lt;blank&gt;</td><td>Indexing is turned off.</td></tr>
<tr><td>INDEX &lt;blank&gt;</td><td>If an index exists in FCB <i>fcbname</i>, it uses that. If an index does not exist in FCB <i>fcbname</i>, it issues message HCP036E.</td></tr>
<tr><td>INDEX 0</td><td>Turns off indexing.</td></tr>
<tr><td>INDEX <i>number</i></td><td>If <i>number</i> is a value 1 to 31, this value is used as the indexing value. If <i>number</i> is greater than 31, message HCP036E is issued.</td></tr>
</table><code>In VM/ESA 1.2.0:</code>
<code>-------------------</code><table>
<tr><th><i>INDEX Parameter</i></th><th><i>What happened in VM/ESA 1.2.0</i></th></tr>
<tr><td>&lt;blank&gt;</td><td>Unpredictible results.</td></tr>
<tr><td>INDEX &lt;blank&gt;</td><td>Unpredictible results.</td></tr>
<tr><td>INDEX 0</td><td>Turned off indexing.</td></tr>
<tr><td>INDEX <i>number</i></td><td>If <i>number</i> was a value 1 to 31, the results were unpredictible. If <i>number</i> was greater than 31, message HCP036E was issued.</td></tr>
</table></td></tr>
<tr><td>LOGOFF or LOGOUT</td><td><b>Upwardly compatible changes:</b>
• Supports the System Console.</td></tr>
<tr><td>LOGON or LOGIN</td><td><b>Incompatible changes:</b>
• A LOGON of a user with LBYONLY in the password field of the USER directory control statement can only be done by an authorized user using the new BY option of the LOGON command.
<b>Upwardly compatible changes:</b>
• Supports the System Console.
• New message: HCP2808E.</td></tr>
</table>

Table 61 (Page 4 of 9). CP Commands Changed since VM/ESA 1.2.0

Command	Changes
PURGE	**Incompatible changes:** • Interrupts an in-process SAVESEG or SAVESYS operation and purges the saved segment or named saved system before it is saved. In VM/ESA 1.2.0, it would wait until the SAVESEG or SAVESYS operation completed before purging the saved segment or named saved system.
QUERY (in general)	**Incompatible changes:** • See QUERY commands below. **Upwardly compatible changes:** • See QUERY commands below.
QUERY ABEND	**Upwardly compatible changes:** • Supports a new operand: SNAPDUMP.
QUERY CACHEFW	**Upwardly compatible changes:** • Response indicates if the cache fast write function is suspended for the subsystem.
QUERY CHPID	**Upwardly compatible changes:** • New operand: TYPE. • New responses if TYPE is specified.
QUERY CPLEVEL	**Incompatible changes:** • The format of the response has changed to include the version. In addition, the service level is returned as *yynn*, where *yy* is the last two digits of the year and *nn* is the sequential number of the RSU tape for that year. In VM/ESA 1.2.0, the service level was always returned as 0000. **Note:** If your installation has APAR VM56239 applied, your system already has the change to the service level information. **Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • In the response, the release level value has changed. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q CPLEVEL command, the output from Q 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.
QUERY CRYPTO	**Upwardly compatible changes:** • Supports new operand: CAMQS. • Two new responses.
QUERY DASDFW	**Upwardly compatible changes:** • Response indicates if the DASD fast write function is suspended for the subsystem.
QUERY IMG	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q IMG command, the output from Q 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.
QUERY LDEVS	**Upwardly compatible changes:** • The response may include the IP address for TCP/IP.

Table 61 (Page 5 of 9). CP Commands Changed since VM/ESA 1.2.0

Command	Changes
QUERY NAMES	**Incompatible changes:** • Response has changed: – Can contain new responses if your installation is using System Console support: `LOGNSYC   - SYSC` *`userid`*`   - SYSC` **Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY NLS	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q NLS command, the output from Q 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.
QUERY NSS	**Incompatible changes:** • The *userid* field in the response may contain SYSTEM. If SYSTEM is the only user ID listed in the *userid* field and the NSS was defined with the VMGROUP option, the NSS or saved segment is temporarily unavailable for loading. In VM/ESA 1.2.0, this did not occur. • For MAP, ENVIRONMENT, and ATTRIBUTES, message HCP1375I is issued when a saved segment or saved system is being loaded onto the system by SPTAPE LOAD, and your QUERY is terminated. In VM/ESA 1.2.0, your QUERY request was delayed until the saved segment or saved system had completed loading. **Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q NSS command, the output from Q 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.
QUERY READER / PRINTER / PUNCH	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses include 4-digit years for FULLDATE and ISODATE operands. • 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. • 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 Q RDR|PRT|PUN command, the output from Q 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).
QUERY SET	**Upwardly compatible changes:** • The response appends additional information: 370ACCOM ON, 370ACCOM OFF.
QUERY TIME	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q TIME command, the output from Q TIME uses the new default date format. This causes the date to be expanded to include the 4-digit year.

Table 61 (Page 6 of 9). CP Commands Changed since VM/ESA 1.2.0

Command	Changes
QUERY TRACE	**Incompatible changes:** • Guest-use of the Subspace-Group Facility is supported only when the machine in which CP is running provides it. In VM/ESA 1.2.0, CP would try to run Subspace-Group Facility whether or not the machine in which CP was running supported it. • Response may have added information.
QUERY TRFILES	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q TRFILES command, the output from Q 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.
QUERY TRSAVE	**Upwardly compatible changes:** • Supports a new operand: FRAMES. • Responses can contain additional information about DEFERIO, FRAMES, and FILLED.
QUERY TRSOURCE	**Upwardly compatible changes:** • Response may contain additional information. • New message: HCP6163E.
QUERY UCR	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q UCR command, the output from Q 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.
QUERY USERID	**Upwardly compatible changes:** • The response can be different if you are using the System Console: `userid   - SYSC` • Supports the EXTended parameter to display the network qualifiers.
QUERY USERS	**Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL ALL	**Upwardly compatible changes:** • Supports virtual message processors.
QUERY VIRTUAL CONSOLE	**Incompatible changes:** • The response may include a new line containing TCP/IP information. **Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL CRYPTO	**Upwardly compatible changes:** • New response.
QUERY VIRTUAL DASD	**Incompatible changes:** • The response has eight digits for cylinders; in VM/ESA 1.2.0 it had four digits. **Upwardly compatible changes:** • In the response, the *volid* field can be (VDSK) and the *rdev* field can be VDSK.

Table 61 (Page 7 of 9). CP Commands Changed since VM/ESA 1.2.0

Command	Changes
QUERY (Virtual Device)	**Upwardly compatible changes:** • Response indicates REAL-MPLF or SIMULATED-MPLF if enabled for the device. • Supports virtual message devices. The device type MSGD appears in the response.
QUERY VIRTUAL GRAF	**Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL LINES	**Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
SAVESEG	**Incompatible changes:** • New message: HCP1374I.
SAVESYS	**Incompatible changes:** • New message: HCP1374I.
SET (in general)	**Upwardly compatible changes:** • See SET commands below.
SET ABEND	**Upwardly compatible changes:** • Supports a new operand: SNAPDUMP.
SET CPTRACE	**Upwardly compatible changes:** • Output from the ALLcodes operand may include type 3 trace codes. • New trace category and trace codes for QDIO instructions.
SET CRYPTO	**Upwardly compatible changes:** • New operands: MODIFY, ON, OFF. • Changed responses. • Changed messages: HCP1706I, HCP1709E, HCP1710E, HCP1711I. • New messages: HCP1714E, HCP1715E.
SET PF*nn*	**Upwardly compatible changes:** • For IMMED and NODISP, the length limit for the input data passed to CP when you press the PF key has changed. The limit is now the greater of the following: – 240 – The length of the input area In VM/ESA 1.2.0, this limit was always the length of the input area. (The length of the input area is twice the terminal line size minus 21.) **Notes:** 1. You can use the LINESIZE field in the QUERY TERMINAL response to determine the terminal line size. Remember that this needs to be the terminal where you will be pressing the PF key. 2. For DELAYED, the limit of the length of data displayed when you press the PF key remains unchanged. It is the length of the input area.
SET SHARED	**Upwardly compatible changes:** • New message: HCP2007E.
SHUTDOWN	**Upwardly compatible changes:** • New message: HCP6425I.
SPTAPE	**Upwardly compatible changes:** • The first two digits of the 4-digit year are included in hexadecimal format in the SFBLOK dumped to tape.

Table 61 (Page 8 of 9). CP Commands Changed since VM/ESA 1.2.0

Command	Changes
STORE (Registers)	**Upwardly compatible changes:** • Supports new operand: FPC *hexword*. • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New messages: HCP6153E, HCP6154E.
STORE STATUS	**Upwardly compatible changes:** • 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.
TRACE (in general)	**Incompatible changes:** • New message: HCP1038E. • See TRACE commands below. **Upwardly compatible changes:** • New command responses. • See TRACE commands below.
TRACE BRANCH	**Incompatible changes:** • More information about address spaces may be trapped when address space options are used (PRI, SECO, ANS, AREG, ALET).
TRACE *mnemonic1*	**Upwardly compatible changes:** • Supports new mnemonics: BSA, BSG.
TRACE *mnemonic2*	**Upwardly compatible changes:** • New mnemonic: SIGA.
TRACE STORE	**Incompatible changes:** • More information about address spaces may be trapped when address space options are used (PRI, SECO, ANS, AREG, ALET).
TRSAVE	**Upwardly compatible changes:** • Supports new operands: DEFERIO, FRAMES. • New messages: HCP6089E, HCP6099E, HCP6426I, HCP6427I, HCP6091I.
TRSOURCE	**Upwardly compatible changes:** • New messages: HCP6089E, HCP6099E, HCP6426I, HCP6427I, HCP6091I, HCP6160E, HCP6161E, HCP6162E, HCP6163E.
TRSOURCE ID	**Upwardly compatible changes:** • Supports new conditional operands for system trace type DATA: IF, THEN, ELSE, ENDIF.
VARY	**Upwardly compatible changes:** • A new response is displayed if the PATH is OFFLINE because of a request initiated by a control unit.
XAUTOLOG	**Incompatible changes:** • Users defined with the LBYONLY operand in the password field of the USER directory control statement cannot be logged on by the XAUTOLOG command if the password validation is required. **Upwardly compatible changes:** • Supports the System Console.

Table 61 (Page 9 of 9). CP Commands Changed since VM/ESA 1.2.0

Command	Changes
XLINK DISPLAY	**Incompatible changes:** • The response has changed; the starting cylinder is five digits. In VM/ESA 1.2.0 it was four digits. • Changed message: HCP2877E. **Upwardly compatible changes:** • The CYL(*cyl*) operand can support up to five digits. In VM/ESA 1.2.0, it supported up to four digits.
XLINK FORMAT	**Incompatible changes:** • Responses and messages have changed; cylinders are five digits rather than four: HCP2868I, HCP2877E. • New responses and messages: HCP2896I, HCP2897E.

SYSTEM CONFIG Statements

Table 62 lists SYSTEM CONFIG file statements that have changed since VM/ESA 1.2.0. See *VM/ESA: Planning and Administration* for complete descriptions of SYSTEM CONFIG statements.

Table 62. SYSTEM CONFIG Statements Changed since VM/ESA 1.2.0

Statement	Changes
CHARACTER_DEFAULTS	**Incompatible changes:** • Letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be defined as the system default logical line edit symbols (line-delete, character-delete, escape, line-end, and tab).
EMERGENCY_MESSAGE_CONSOLES	**Upwardly compatible changes:** • Supports a new operand: SYSTEM_CONSOLE.
OPERATOR_CONSOLES	**Upwardly compatible changes:** • Supports a new operand: SYSTEM_CONSOLE.
XLINK_VOLUME_INCLUDE	**Upwardly compatible changes:** • CYLINDER and MAP_RECORD_LENGTH operands support up to five digits. In the old release, they supported up to four digits.
XLINK_DEVICE_DEFAULTS	**Upwardly compatible changes:** • CYLINDER and MAP_RECORD_LENGTH operands support up to five digits. In the old release, they supported up to four digits. • The 3390 CLASS operand supports a new value.

User Directory Control Statements

Table 63 lists the user directory control statements that have changed since VM/ESA 1.2.0. See the *VM/ESA: Planning and Administration* book for complete descriptions of user directory control statements.

Note: You can use the VMUDQ macro to query the CP user directory. See the *VM/ESA: CP Programming Services* book for more information on this macro.

Table 63 (Page 1 of 2). User Directory Control Statements Changed since VM/ESA 1.2.0

Statement	Changes
CRYPTO	**Upwardly compatible changes:** • New operand: MODIFY.
DASDOPT	**Upwardly compatible changes:** • Supports new operands: NOWRKALLEG, WRKALLEG.
DATEFORMAT	**New** Specifies a user's default date format for commands that provide multiple date formats.
GLOBALDEFS	**New** Signifies the beginning of the global definition section.
GLOBALOPTS	**New** Used to define global settings to be used while processing user definitions.
LOAD	**Upwardly compatible changes:** • Supports a new G operand for global definitions.
MDISK	**Upwardly compatible changes:** • Supports new operand: V-DISK.
MINIOPT	**Upwardly compatible changes:** • Supports new operands: NOWRKALLEG, WRKALLEG.
OPTION	**Upwardly compatible changes:** • 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. • Specifying the TODENABLE operand allows a user to change the virtual machine TOD clock with the new CP SET VTOD command. • Supports new operands: CFVM, CFUSER, DIAG88.
POSIXGLIST	**New** Specifies all POSIX groups of which the user is a member.
POSIXGROUP	**New** Defines a POSIX group.
POSIXINFO	**New** Specifies a user's POSIX information.
POSIXOPT	**New** Specifies option settings related to a user's POSIX capabilities.
SPECIAL	**Upwardly compatible changes:** • Supports new operand: MSGPROC.

Table 63 (Page 2 of 2). User Directory Control Statements Changed since VM/ESA 1.2.0

Statement	Changes
STDEVOPT	**Upwardly compatible changes:** • Supports new DASDSYS operands: DATAMOVER and NODATAMOVER.
USER	**Incompatible changes:** • User IDs LOGNSYSC and SYSC are reserved for CP use. • LBYONLY is a new operand that can be specified in the password field of the USER statement. Any user defined with the LBYONLY operand may be restricted from performing operations that require password validation (including LOGON, AUTOLOG, XAUTOLOG, and DIAGNOSE X'84'). • Letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be defined as logical line edit symbols (line-end, line-delete, character-delete, and escape).

CP Utilities

Table 64 lists the utility programs that have changed since VM/ESA 1.2.0. See the *VM/ESA: CP Command and Utility Reference* for details on utility programs.

Table 64. Utility Programs Changed since VM/ESA 1.2.0

Utility	Changes
CPFMTXA	**Incompatible changes:** • Some messages and responses contain five digits for the number of cylinders. They used to contain four digits. **Upwardly compatible changes:** • The *sext*, *eext*, and *num* operands support a 1- to 5-digit decimal integer. In VM/ESA 1.2.0, it supported a 1- to 4-digit decimal integer.
DASD Dump Restore	**Incompatible changes:** • The END OF VOLUME CYL response has eight digits for the number of cylinders. In VM/ESA 1.2.0, it had four digits. • Some device types are no longer supported by IBM. See the *VM/ESA: General Information* book for information on device support. **Upwardly compatible changes:** • For input, up-to-eight-digit cylinder values are supported. In VM/ESA 1.2.0, up-to-four digit cylinder values were supported.
DIRECTXA	**Incompatible changes:** • The format of the response has changed to include the version. • 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'0F' (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.
HCPLDR	**Upwardly compatible changes:** • Supports a new operand: SMALLMAP.
SALIPL	**Incompatible changes:** • Message HCP039E deleted, replaced by new message HCP394E with same text. • Date field on the file list panel displays the year with 4 digits.

CP DIAGNOSE Codes

Table 65 lists the DIAGNOSE codes that have changed since VM/ESA 1.2.0. See the *VM/ESA: CP Programming Services* book for complete information on the DIAGNOSE codes discussed in this section.

Table 65 (Page 1 of 2). DIAGNOSE Codes Changed since VM/ESA 1.2.0

Code	Changes
X'00'	Storage extended identification code **Incompatible changes:** • The Environment field has changed from 3 bytes to 2 bytes. • The new Version Information field (formerly the third byte of the Environment field) identifies the version number of the product (for VM/ESA 2.1.0 and later). **Upwardly compatible changes:** • The value in the program product bit map has changed to indicate the new release level. Also, Bit 13 (X'0004000000000000') indicates whether Year 2000 support is present in CP.
X'14'	Input spool file manipulation **Upwardly compatible changes:** • For subcodes X'0004', X'0008', X'0FFE', and X'0FFF', a one-byte century indicator was added to the SFBLOK data area.
X'24'	Device type and features **Upwardly compatible changes:** • Output information can contain additional information.
X'64'	Find, load, purge a named saved segment **Incompatible changes:** • If you try to load or find a saved segment that is in the process of being saved, you get return code 44 (X'02C'), and the saved segment is not loaded or found. In VM/ESA 1.2.0, DIAGNOSE code X'64' completion was delayed until the save was complete. Then the saved segment was loaded or found. • If an alternate user ID (established using DIAGNOSE code X'D4') is manipulating a restricted saved segment, CP verifies that the alternate user ID has a NAMESAVE statement in its entry in the user directory, which authorizes it to use the restricted segment. In VM/ESA 1.2.0, if the original user ID did not have authority to manipulate the restricted segment, the alternate user ID also was not allowed to use the restricted segment, even if the alternate user ID had authority.
X'68'	Virtual Machine Communication Facility (VMCF) **Upwardly compatible changes:** • Supports a new function: SETLIMIT (Subcode X'000C').
X'7C'	Logical Device Support Facility **Upwardly compatible changes:** • 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.

Table 65 (Page 2 of 2). DIAGNOSE Codes Changed since VM/ESA 1.2.0

Code	Changes
X'84'	Directory Update-in-Place **Incompatible changes:** • Users defined with LBYONLY in the password field of the USER directory control statement cannot be the target of DIAGNOSE code X'84' operations unless the virtual machine issuing the DIAGNOSE code has the D84NOPAS option in its directory entry and the operation specified is not LOGPASS or MDISK. • For EDITCHAR operation, letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be specified as logical line edit symbols (line-end, line-delete, character-delete, and escape). **Upwardly compatible changes:** • Can now replace the following new information: – User's default date format setting • New operation: DATEFMT. • New return codes due to new function (in hex): 122, 123, 124.
X'D8'	Read spool file blocks on the system spool file queues **Upwardly compatible changes:** • For subcode X'0000', a one-byte century indicator was added to the SFBLOK data area.
X'E4'	Return Minidisk Real Device Information **Upwardly compatible changes:** • The VOLSER and RDEVNO fields in the output area for functions X'00' and X'01' contain different information for new support (virtual disks in storage).
X'BC'	Open and Query Spool File Characteristics **Upwardly compatible changes:** • 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.
X'210'	Retrieve Device Information **Upwardly compatible changes:** • Output can contain additional information.

CP Macros

Table 66 lists the IUCV macro functions for use in APPC/VM that have changed since VM/ESA 1.2.0.

The information in this table is based on two assumptions:

1. You have used the parameter lists as documented in the publications (that is, you have not used any undefined parts of the parameter list for your own purposes).
2. When you are converting your APPC/VM applications, your communications partners are in the same environment they were in before your conversion (that is, your communications partners are not exploiting new function).

See the *VM/ESA: CP Programming Services* book for complete descriptions of APPC/VM and IUCV macros.

Table 66. CP Macros Changed since VM/ESA 1.2.0

Macro Function	Changes
IUCV ALLOW	**Incompatible changes:** • Specifying IUCV ALLOW on a target node no longer authorizes remote priority.
IUCV CONNECT	**Incompatible changes:** • Priority fails on remote connections if dependent on an IUCV ALLOW directory control statement on the target node to authorize the priority. Use either the IUCV ANY or IUCV target_userid PRIORITY control statements in the invoker's directory to ensure proper authorization. **Upwardly compatible changes:** • Bit 2 of the FLAG byte of the User Data Field for CONNECT now returns the resource as a system resource. • New value in the Connection Complete Interrupt: IPPOLLFG (Byte X'24'). • New optional parameters supported: TARGET=*address* LOCAL=YES|NO

CP Messages

The following CP messages do not exist in VM/ESA 2.4.0:

HCP543A
HCP1365E
HCP8611T

The following list identifies the CP messages that have changed since VM/ESA 1.2.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CP messages.

Action indicators (for example, the “E” in message HCP006E) in VM/ESA 2.4.0 messages may differ from the corresponding messages in VM/ESA 1.2.0, even though the message number may be the same.

The nonzero return code and accompanying message that VM/ESA 2.4.0 returns for errors may not be the same as the nonzero return code that VM/ESA 1.2.0 returns for corresponding errors.

VM/ESA 2.4.0 suppresses leading zeros in responses more often than VM/ESA 1.2.0 does.

HCP006E
HCP031E
HCP054E
HCP091E
HCP114E
HCP145I
HCP233E
HCP296E
HCP319E
HCP332E
HCP389E
HCP403I
HCP481E
HCP512I
HCP574I
HCP580I
HCP592I
HCP704E
HCP711D
HCP716D
HCP717D
HCP725D
HCP751E
HCP752E
HCP753E
HCP772E
HCP799E
HCP904W
HCP1003E
HCP1006E
HCP1010W
HCP1011E
HCP1016E
HCP1018E
HCP1115E
HCP1116E
HCP1120E
HCP1365E
HCP1512E
HCP1706I
HCP1709E
HCP1710E
HCP1711I
HCP1791E
HCP1879E
HCP2002I
HCP2234E
HCP2252E
HCP2760E
HCP2768E
HCP2779E
HCP2868I

HCP2877E
HCP2895I
HCP6014I
HCP6091I
HCP6111I
HCP6283I
HCP6706E
HCP6739E
HCP6743E
HCP6788E
HCP6802E
HCP6804E
HCP8028W
HCP8080I
HCP9020W
HCP9021W
HCP9022W
HCP9036W
HCP9225I
HCP9405E
HCP9408E
HCP9410I
HCP9422E

CMS Changes

This section identifies the CMS externals that have changed since VM/ESA 1.2.0. It contains the following subsections:

- “CMS Commands”
- “CMS File Pool Administration and Operator Commands” on page 479
- “XEDIT Subcommands” on page 480
- “CMS Macros” on page 482
- “Preferred Routines (CSL Routines)” on page 486
- “CSLCNTRL File” on page 496
- “Compatibility Routines” on page 497
- “CMS Pipelines” on page 497
- “CMS Messages” on page 498

CMS Commands

Table 67 lists the CMS commands that have changed since VM/ESA 1.2.0. See the *VM/ESA: CMS Command Reference* for complete descriptions of CMS commands.

Note: The three character module identifiers have been removed from the messages listed in the *VM/ESA: CMS Command Reference*. For example, a message that used to be listed as DMSAMS136S is now listed as DMS136S.

Table 67 (Page 1 of 11). CMS Commands Changed since VM/ESA 1.2.0

Command	Changes
ACCESS	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added message for use with BFS: DMS2133E.
ALIALIST	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
AUTHLIST	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 67 (Page 2 of 11). CMS Commands Changed since VM/ESA 1.2.0

Command	Changes
COPYFILE	**Incompatible changes:** • Changed return code: – Return code 0 is returned when the SFS file space threshold is exceeded. In VM/ESA 1.2.0, you received return code 4. • New message: DMS516E. **Upwardly compatible changes:** • Support for the new CMS SET RORESPECT command has been added. COPYFILE will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O.
CREATE ALIAS	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
CREATE DIRECTORY	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New message for authorization failure from ESM: DMS1331E.
CREATE FILE	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
CREATE LOCK	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added messages for use with BFS: DMS2040E, DMS2133E. • Added new operand for use with BFS: *bfsid*.
CREATE NAMEDEF	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
CSLGEN	**Upwardly compatible changes:** • Supports new options: FILETYPE, COPYTYPE, NOAUTO, AUTO, NOMAP, MAP. • Changed messages: – DMS056E (new text possible) – DMS1096E (new text possible) – DMS2055I (new text possible)
CSLLIST	**Upwardly compatible changes:** • 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.
CSLMAP	**Upwardly compatible changes:** • 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.

Table 67 (Page 3 of 11). CMS Commands Changed since VM/ESA 1.2.0

Command	Changes
DEFAULTS	**Upwardly compatible changes:** • New options can be specified as parameters for DISK LOAD and READCARD: FIFO, LIFO, NOTYPE, STACK, TYPE. • New options are supported as parameters for FILELIST and DIRLIST: OWNER, READWRITE, READONLY. • A new option is supported as a parameter for NETDATA SEND, SENDFILE, and NOTE: CLASS. • A new option is supported for NAMES: VMLINK. • Support for the new command VMLINK. • Supports new options: NOKEEPCC, KEEPCC. • New options are supported as parameters for FILELIST, NETDATA, and RDRLIST: VMDATE, SHORTDATE, FULLDATE, ISODATE.
DELETE LOCK	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*.
DIRATTR	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DIRLIST	**Upwardly compatible changes:** • Supports new options: OWNER, READWRITE, READONLY. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • 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.
DISCARD	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DISK	**Upwardly compatible changes:** • Supports new operands for DISK LOAD: *filename*, *filetype*, *filemode*. • Supports new options: FIFO, LIFO, NOTYPE, STACK, TYPE, MSGALL, MSGSUBS.
DOSLKED	**Upwardly compatible changes:** • Supports new options: INV, NOINV. • Can now run in XA and XC virtual machines.
EDIT	**Incompatible changes:** • The '(OLD' option is no longer supported. You must either use the XEDIT simulation by dropping the '(OLD' option or convert to using the normal XEDIT environment. • Message DMS987E was changed to DMS2520E. The old CMS editor is no longer supported.
ERASE	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*. • New message for authorization failure from ESM: DMS1332E.

Table 67 (Page 4 of 11). CMS Commands Changed since VM/ESA 1.2.0

Command	Changes
EXECIO	**Incompatible changes:** • A write to a file in an SFS directory that is in a VM/ESA Version 1 Release 2.2 or later file pool fails when CMS detects that the file space is full. You receive an error message. In VM/ESA 1.2.0, your application did not receive a file space full condition, and you could continue writing.
FILEATTR	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
FILEDEF	**Upwardly compatible changes:** • New option: LIBSRV. • Allows LRECL definitions up to 65535 bytes for OS variable spanned records (under XLRI processing) and non-OS CMS files. • Allows BLKSIZE definitions up to 65535 bytes for non-OS CMS files.
FILELIST	**Incompatible changes:** • 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 LISTFILE profiles. See Appendix A of the *VM/ESA: CMS Command Reference*. • If you file the file created by FILELIST, that file might contain new and changed fields (on the far right). **Upwardly compatible changes:** • Supports new options: OWNER, READWRITE, READONLY. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • The BEFORE *date* and AFTER *date* options support 4-digit years. • Screens and responses support 4-digit years. • If a date format option is not specified on the FILELIST command, the CMS DEFAULTS date format setting for FILELIST will be used. • Added message for use with BFS: DMS2133E. • 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.
FILEPOOL CONTROL BACKUP	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
FILEPOOL MINIDISK	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
FILESERV BACKUP	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
FILESERV DEFBACKUP	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
FINIS	**Incompatible changes:** • Will close files opened by the REXX STREAM I/O function.

Table 67 (Page 5 of 11). CMS Commands Changed since VM/ESA 1.2.0

Command	Changes
GLOBAL	**Upwardly compatible changes:** • New message for duplicate library name in input list: DMS045W. The duplicates are ignored.
GRANT AUTHORITY	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
IDENTIFY	**Upwardly compatible changes:** • New options to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses support 4-digit years. • 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. • New option for TCP/IP mail integration: TCPIP.
LISTDIR	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
LISTDS	**Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Responses support 4-digit years. • 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.
LISTFILE	**Incompatible changes:** • New message: DMS389E. • Message DMS550E changed to DMS765E. **Upwardly compatible changes:** • New options for ALLDATES option: DTOC, DOLR, DTOLU, DTOLC. • New options to specify date format: SHORTDATE, FULLDATE, ISODATE. • The BEFORE *date* and AFTER *date* options support 4-digit years. • Responses support 4-digit years. • 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.
LKED	**Upwardly compatible changes:** • The defaults for *value1* and *value2* on the SIZE option have been increased to 400K and 100K, respectively.
LOAD	**Upwardly compatible changes:** • New message for insufficient storage above 16MB: DMS891W.
LOADMOD	**Incompatible changes:** • Changed message (new text possible): DMS639E.
MACLIST	**Upwardly compatible changes:** • 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.

Table 67 (Page 6 of 11). CMS Commands Changed since VM/ESA 1.2.0

Command	Changes
MOVEFILE	**Upwardly compatible changes:** • Can process OS variable spanned records (under XLRI processing) and non-OS CMS files with record lengths up to 65535 bytes. • Adjusts output file sizes for compatibility between CMS and OS • Allows greater FILEDEF default flexibility for file attributes (RECFM, LRECL, BLKSIZE) • If default size values are used, fixes record truncation problems when moving data files from fixed to variable format • New message: DMS1116E.
NAMEFIND	**Upwardly compatible changes:** • Supports the new form of the FILE opt, FILE (fn ft fm).
NAMES	**Upwardly compatible changes:** • Supports new option: VMLINK. • On the MAIL panel: – The Userid and Node fields will no longer be uppercased.
NETDATA	**Incompatible changes:** • Different message on empty reader condition: DMS205W (was DMS639E). • New message: DMS1184E • New return code: 20 **Upwardly compatible changes:** • NETDATA SEND supports a new option: CLASS. • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file. • Responses support 4-digit years. • 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.
NOTE	**Incompatible changes:** • In a note, a nonblank line between the 'OPTIONS:' line and the 'Date:' line, called 'USEROPTIONS:', is supported. Any other nonblank lines cause an error. In VM/ESA 1.2.0, you received an error message for any nonblank line between 'OPTIONS:' and 'Date:'. **Upwardly compatible change:** • The OPTIONS: line of a note contains a new option: CLASS. • In the Date field of the note header, the year is now displayed with four digits. • Changed message: – DMS2501E (text change for new function). • TCP/IP domain names accepted as user IDs or as the resolution of nicknames. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
NUCXLOAD	**Incompatible changes:** • Changed message (new text possible): DMS639E.
PARSECMD	**Upwardly compatible changes:** • BFS added new code.n value of PN.

Table 67 (Page 7 of 11). CMS Commands Changed since VM/ESA 1.2.0

Command	Changes
PEEK	**Incompatible changes:** • PEEK profile has changed. If you used the PROFPEEK profile to create your own, PEEK may not work anymore. Use new PROFPEEK. **Upwardly compatible change:** • TCP/IP origin name address used when available and shown on PEEK message line for origin within current space and formatting limitations.
PIPE	**Incompatible changes:** • All CMS Pipelines messages have a new prefix, and many messages have new numbers and text. See "CMS Pipelines" on page 497.
QUERY (in general)	**Incompatible changes:** • See changed commands below. **Upwardly compatible changes:** • Supports new operands.
QUERY ACCESSORS	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
QUERY ALIAS	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
QUERY AUTHORITY	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
QUERY BLOCKS	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*.
QUERY CMSLEVEL	**Incompatible changes:** • In the response, the CMS level value has changed and is now two digits.
QUERY CMSREL	**Incompatible changes:** • The format of the response has changed to include the version. **Upwardly compatible changes:** • In the response, the release level value has changed.
QUERY DIRATTR	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
QUERY ENROLL	**Upwardly compatible changes:** • Added new option for use with BFS: FILESPACE.
QUERY FILEATTR	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*.

Table 67 (Page 8 of 11). CMS Commands Changed since VM/ESA 1.2.0

Command	Changes
QUERY FILEDEF	**Upwardly compatible changes:** • Supports a new optional operand, ATTRIBUT, and its response.
QUERY LIMITS	**Incompatible change:** • File pool administration authority is no longer needed to query limits on a file space other than your own.
QUERY LOCK	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*.
QUERY NAMEDEF	**Upwardly compatible changes:** • Added new operand for use with BFS: *bfsid*.
QUERY RORESPECT	**Upwardly compatible changes:** • Displays the current RORESPECT setting. This setting controls whether or not updates to files using XEDIT and/or COPYFILE in SFS will respect the read-only access of a directory.

QUERY SYSNAMES	**Upwardly compatible changes:**
	• There is a change to the response. The CMSGUI saved system name is displayed.
	An example of the VM/ESA 2.4.0 response:
	`SYSNAMES:  CMSVSAM  CMSAMS   CMSDOS   CMSBAM   CMSGUI`
	`ENTRIES:   CMSVSAM  CMSAMS   CMSDOS   CMSBAM  VMGUILIB`

RDRLIST	**Incompatible changes:** • If you file the file created by RDRLIST, that file might contain new and changed fields (on the far right). **Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Screen supports 4-digit years. • If a date format option is not specified on the RDRLIST command, the CMS DEFAULTS date format setting for RDRLIST will be used. • 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. • TCP/IP origin domain name address used when available and shown on RDRLIST panel origin area within current space and formatting limitations.
READCARD	**Upwardly compatible changes:** • Supports new options: FIFO, LIFO, NOTYPE, STACK, TYPE, MSGALL, MSGSUB.
RECEIVE	**Incompatible changes:** • When you receive DISK DUMP format files, all the files get created on the specified file mode. Entries are made in your NETLOG file for all the files created. In VM/ESA 1.2.0, only the first file was created on the specified file mode and logged in your NETLOG file. The other files, if any, were created on the A-disk and not logged. • When receiving DISK DUMP format files to a file mode other than A, you are not required to have the A file mode accessed as read/write. In VM/ESA 1.2.0, you were required to have an A-disk accessed as read/write. • Added message: DMS1138E. **Upwardly compatible changes:** • New message: DMS3033W. • Supports new options: NOKEEPCC, KEEPCC. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.

Table 67 (Page 9 of 11). CMS Commands Changed since VM/ESA 1.2.0

Command	Changes
RELEASE	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
RELOCATE	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
RENAME	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
REVOKE AUTHORITY	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
RTNLOAD	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
SEGMENT	**Incompatible changes:** • For LOAD or RESERVE, if you are trying to load or reserve a saved segment that is in the process of being saved, you receive error message DMS1083E and the saved segment is not loaded or reserved. In VM/ESA 1.2.0, your load or reserve request was delayed until the save was complete, and then the saved segment was loaded or reserved. • For PURGE and RELEASE, if the segment is a logical saved segment containing minidisk information, the minidisks are released. In VM/ESA 1.2.0, the minidisks were not released.
SENDFILE	**Incompatible changes:** • The SENDFILE screen has changed. A new line is added for the new CLASS option. Also, if your screen size is 24 lines, some input fields may be in different locations on the screen. • When sending a note, if incorrect information is found in the new *class* field of the OPTIONS: line, the note is not sent. In VM/ESA 1.2.0, information found after the NOTEBOOK option field on the OPTIONS: line was ignored. **Upwardly compatible changes:** • Supports a new option: CLASS. • Changed messages: DMS081E, DMS2501E. • TCP/IP domain names accepted as user IDs or as the resolution of nicknames. • New options to specify the transmission method: SMTP, MIME, UFTSYNC, UFTASYNC. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
SET (in general)	**Upwardly compatible changes:** • Supports new operands.
SET AUTODUMP	**Upwardly compatible changes:** • For some severe irrecoverable file system errors detected by CMS, an entire virtual machine dump is generated unless SET AUTODUMP OFF was specified. In VM/ESA 1.2.0, your AUTODUMP setting was always respected.
SET DOS	**Incompatible changes:** • Added message: DMS1101I. • Changed message: DMS333E.
SET DOSPART	**Incompatible changes:** • Added message: DMS1101I. • Changed message: DMS333E.

Table 67 (Page 10 of 11). CMS Commands Changed since VM/ESA 1.2.0

Command	Changes
SET RORESPECT	**Upwardly compatible changes:** • Used to ensure updates to files using XEDIT and/or COPYFILE in SFS will respect the read-only access mode of a directory.
SET SYSNAME	**Upwardly compatible changes:** • Supports new operand: CMSGUI
SYNONYM	**Incompatible changes:** • Removed message: DMS032E. **Upwardly compatible changes:** • Supports a new parameter: *filetype*. In VM/ESA 1.2.0, file type had to be SYNONYM.
TAPE	**Upwardly compatible changes:** • 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. • Added message for Tape Library Dataserver support: DMS2147W.
TELL	**Upwardly compatible changes:** • Accepts a TCP/IP domain name as part of the destination information.
UPDATE	**Incompatible changes:** • Some return codes are changed: – When an update file listed in an AUX file is not found, return code 12 is always returned. In VM/ESA 1.2.0, if at least one update file listed in an AUX file was found and one was not found, return code 12 was returned; but, if no update files listed in an AUX file were found, return code 40 was returned. – When message DMS1259E (File pool *filepoolid* has run out of physical space in the storage group) is issued, return code 31 is returned. In VM/ESA 1.2.0, return code 40 was returned. – In other situations where return code 40 was returned for SFS-related errors, return code 100 is now returned.
VMFPLC2	**Upwardly compatible changes:** • 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. • Added message for Tape Library Dataserver support: DMS2147W.

Table 67 (Page 11 of 11). CMS Commands Changed since VM/ESA 1.2.0

<table>
<tr><th>Command</th><th>Changes</th></tr>
<tr><td>XEDIT</td><td>

Incompatible changes:

- The file mode number of the autosave file may be different:

```
        Edited     Autosave
        File's     File's
Version File Mode  File Mode
Release Number     Number
-----------------------------
1.2.0   anything   0
-----------------------------
2.4.0   0          0
        any other  1
-----------------------------
```

Upwardly compatible changes:

- Added new option for BFS: NAMETYPE If NAMETYPE BFS is used, your profile must be in REXX and it is loaded as a REXX function.
- Added new option for BFS: BFSLINE
- New messages for BFS: DMS033E, DMS512E, DMS2105E, and DMS2134E.

</td></tr>
</table>

CMS File Pool Administration and Operator Commands

Table 68 lists the CMS file pool administration and operator commands that have changed since VM/ESA 1.2.0. See the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book for complete descriptions of file pool administration and operator commands.

Table 68 (Page 1 of 2). CMS File Pool Administration and Operator Commands Changed since VM/ESA 1.2.0

Command	Changes
AUDIT	**Upwardly compatible changes:** • New operands: *fn ft*, REPLACE. • Added messages: DMS024E, DMS1258E, DMS3253I, DMS3254E, DMS3255E. • Changed message: DMS3470W (new text possible).
DATASPACE	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DELETE USER	**Upwardly compatible changes:** • New options: DELAUTH | KEEPAUTH. • Added message: DMS2023E.
ENROLL USER	**Upwardly compatible changes:** • Supports new options: SFS | BFS, *userid*, *gname*. • Added messages for use with BFS: DMS1209E, DMS2023E, DMS2132E.
FILEPOOL BACKUP	**Upwardly compatible changes:** • Supports new BFS parameter: *bfsid*.
FILEPOOL RELOAD	**Upwardly compatible changes:** • New message: DMS3455I.

Table 68 (Page 2 of 2). CMS File Pool Administration and Operator Commands Changed since VM/ESA 1.2.0

Command	Changes
FILEPOOL RENAME	**Incompatible changes:** • The user ID renamed does not have to be enrolled in the file pool. Locks and authorizations are transferred to the new user ID. • New message: DMS2544W (this message replaces DMS1167E when both the administrator's CMS and the SFS server are at the new release level).
FILEPOOL UNLOAD	**Upwardly compatible changes:** • New message: DMS3455I.
FILESERV REORG	**Incompatible changes:** • A new response message, DMS3009R, is issued to recommend that you cancel the processing of the FILESERV REORG command if your control data backup is not current.
FILESERV START	**Incompatible changes:** • The start-up does not continue processing if the DMS3110E message is issued because of a SEGMENT ASSIGN failure.
QUERY DATASPACE	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
QUERY FILEPOOL STATUS	**Incompatible changes:** • The following Counter Information has been added to the output: – Open File CreateMig Requests • The *Wait* column may include the new wait state, *DFSMS/VM_Wait*, if you are exploiting function of DFSMS/VM. • The *Wait* column may include the new wait state, *Storage_Wait* if an agent is waiting for storage to be available on the server. • The colon (:) on the *filepoolid* option is now optional.
QUERY LIMITS	**Incompatible changes:** • File pool administration authority is no longer required.

XEDIT Subcommands

Table 69 lists the XEDIT subcommands that have changed since VM/ESA 1.2.0. For complete descriptions of XEDIT subcommands, see the *VM/ESA: XEDIT Command and Macro Reference*.

Table 69 (Page 1 of 3). XEDIT Subcommands Changed since VM/ESA 1.2.0

Subcommand	Changes
A	**Incompatible changes:** • If you issue the A subcommand in non-display mode: – In VM/ESA 1.2.0, XEDIT returns message DMS529E with RC=3. – In VM/ESA 2.4.0, the subcommand completes successfully with no message and RC=0.
ADD	**Incompatible changes:** • If you issue the ADD subcommand in non-display mode: – In VM/ESA 1.2.0, XEDIT returns message DMS529E with RC=3. – In VM/ESA 2.4.0, the subcommand completes successfully with no message and RC=0.

Table 69 (Page 2 of 3). XEDIT Subcommands Changed since VM/ESA 1.2.0

Subcommand	Changes
CMS	**Upwardly compatible change:** • New message added: DMS642E
COUNT	**Incompatible changes:** • The number of occurrences of the target string is returned regardless of the CP EMSG or XEDIT MSGMODE settings. In VM/ESA 1.2.0, CP EMSG and XEDIT MSGMODE settings were respected.
CP	**Upwardly compatible change:** • New message added: DMS642E
EXTRACT	**Upwardly compatible change:** • New operands added: BFSLine, EPName, NAMetype, PName, and GUI.
FILE	**Upwardly compatible changes:** • Support for the new CMS SET RORESPECT command has been added. FILE will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O. • Support for BFS files has been added. • New messages for BFS support: 024E, 054E, 1184E, 2105E, 2120E, 2131E, and 2134E.
GET	**Upwardly compatible changes:** • Support for BFS files has been added. • New messages for BFS support: 033E, 054E, 512E, 2105E, 2131E, and 2134E.
I	**Incompatible changes:** • If you issue the I subcommand in non-display mode: – In VM/ESA 1.2.0, XEDIT returns message DMS529E with RC=3. – In VM/ESA 2.4.0, the subcommand completes successfully with no message and RC=0.
LOAD	**Upwardly compatible changes:** • Support for writing to BFS files has been added. • Added new options: NAMetype and BFSLine. • New messages for BFS support: 033E, 512E, 2105E, and 2134E.
MODIFY	**Upwardly compatible changes:** • New operands added: BFSLine, NAMetype, and PName.
PRESERVE	**Upwardly compatible changes:** • New operands added: BFSLine, NAMetype, and PName.
PUT	**Incompatible changes:** • Fails when using PUT to a packed file; RC 40 and message DMS743E are issued. In VM/ESA 1.2.0, it did not fail; RC 0 and no message were issued. • The temporary file created (when a *fileid* is not specified) for use by GET is erased when all rings are exited. In VM/ESA 1.2.0, the temporary file was erased when any one of the rings was exited. **Upwardly compatible changes:** • Support for the new CMS SET RORESPECT command has been added. PUT will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O. • Support for writing to BFS files has been added. • New return code for BFS support: 32. • New messages for BFS support: 033E, 054E, 512E, 1184E, 2105E, 2120E, 2131E, and 2134E.

Table 69 (Page 3 of 3). XEDIT Subcommands Changed since VM/ESA 1.2.0

Subcommand	Changes
PUTD	**Incompatible changes:** • Fails when using PUTD to a packed file; RC 40 and message DMS743E are issued. In VM/ESA 1.2.0, it did not fail; RC 0 and no message were issued. • The XEDIT temporary file created when no file name is specified can be used by a GET in the current or in any subsequent rings. In VM/ESA 1.2.0, the XEDIT temporary file was erased if any of the subsequent rings were exited. **Upwardly compatible changes:** • Support for the new CMS SET RORESPECT command has been added. PUTD will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O. • Support for writing to BFS files has been added. • New return code for BFS support: 32. • New messages for BFS support: 033E, 054E, 512E, 1184E, 2105E, 2120E, 2131E, and 2134E.
QUERY	**Upwardly compatible change:** • New operand added.
SAVE	**Upwardly compatible changes:** • Support for the new CMS SET RORESPECT command has been added. SAVE will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O. • Support for writing to BFS files has been added. • New return code for BFS support: 32. • New messages for BFS support: 024E, 054E, 512E, 1184E, 2105E, 2120E, 2131E, and 2134E.
SET	**Upwardly compatible change:** • New operand added.
XEDIT	**Incompatible changes:** • See the entry for XEDIT in Table 67 on page 469 for details on the incompatibilities. • In update mode, the XEDIT subcommand no longer allows the same file in the ring multiple times. When a *fileid* match in the ring is found, that file becomes the current file.

CMS Macros

Table 70 lists the CMS preferred macros that have changed since VM/ESA 1.2.0. Table 71 on page 485 lists the CMS compatibility macros that have changed. Table 72 on page 485 lists the OS simulation macros that have changed. Table 73 lists the simulated OS/MVS supervisor calls that have changed.

Preferred Macros

See the *VM/ESA: CMS Application Development Reference for Assembler* for complete descriptions of these macros.

Table 70 (Page 1 of 3). CMS Preferred Macros Changed since VM/ESA 1.2.0

Macro	Changes
AMODESW	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370

Table 70 (Page 2 of 3). CMS Preferred Macros Changed since VM/ESA 1.2.0

Macro	Changes
CMSCVT	**Incompatible changes:** • The CVTSECT that is generated has changed. The CVTNUCB field is reserved. In VM/ESA 1.2.0, it was the lowest storage address not in the nucleus and was initialized to X'20000'. • Field CVTTZ in CVTSECT is updated to reflect current timezone offset when x'2004' interrupt is processed by CMS. **Upwardly compatible changes:** • Added the fields: CVTECVT, CVTFLAG2.
CMSLEVEL	**Incompatible changes:** • The CMS level has been frozen at X'0F' for CMS Level 12 (VM/ESA Version 2 Release 1.0) and later. Use the new DMSQEFL macro or the DMSQEFL CSL routine instead.
CONSOLE READ	**Upwardly compatible changes:** • Supports a new OPTIONS= option: *reg*.
CONSOLE WRITE	**Upwardly compatible changes:** • Supports a new OPTIONS= option: *reg*. • Supports a new parameter: BRKKEY=.
CSLENTRY	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370
CSLFPI	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370
DIRBUFF	**Upwardly compatible changes:** • The FILE record contains the following new fields: DIRFDAXD, DIRFDAXC, DIRFDAXI, DIRFLV13. • The FILEEXT record contains the following new fields: DIREDAXD, DIREDAXC, DIREDAXI, DIREDRXD, DIREDRXC, DIREDRXI, DIRECDXD, DIRECDXC, DIRECDXI, DIREDCXD, DIREDCXC, DIREDCXI, DIRELV13. • The SEARCHALL and SEARCHAUTH records contain the following new fields: DIRSDAXD, DIRSDAXC, DIRSDAXI, DIRSCEND, DIRSLV13.
ENABLE	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370
EXSBUFF	**Upwardly compatible changes:** • The FILE record contains the following new fields: EXSFDIRL, EXSFDIRD, EXSFCONV, EXSFDAXD, EXSFDAXC, EXSFDAXI, EXSFDRXD, EXSFDRXC, EXSFDRXI, EXSFCDXD, EXSFCDXC, EXSFCDXI, EXSFDCXD, EXSFDCXC, EXSFDCXI, EXSF2000, EXSFRES, EXSFLV13. EXSFTDFL. • The DIR record contains the following new fields: EXSDUNQD, EXSDDCXD, EXSDDCXC, EXSDDCXI, EXSDCDXD, EXSDCDXC, EXSDCDXI, EXSD2000, EXSDRES, EXSDLV13, EXSDTDFL.
FSSTATE	**Upwardly compatible changes:** • 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*nn*, 1=20*nn*, where *nn* is the 2-digit year). In VM/ESA 1.2.0, this bit was not used.
FSTD	**Upwardly compatible changes:** • The FSTFLAGS section contains the new FSTCNTRY field, which is a bit that indicates the century (first two digits of the year) the file was last written or updated (0=19*nn*, 1=20*nn*, where *nn* is the 2-digit year).

Table 70 (Page 3 of 3). CMS Preferred Macros Changed since VM/ESA 1.2.0

Macro	Changes
FSWRITE	**Incompatible changes:** • An FSWRITE to a file in an SFS directory that is in a VM/ESA 1.2.1 or later file pool fails when CMS detects that the file space is full. Your application receives an error return code (RC=13). In VM/ESA 1.2.0, your application did not receive a file space full condition, and you could continue writing. • Changed return code when writing to SFS files: 13. • If the caller's buffer overlaps internal CMS file system buffers, an error code 2 will be returned, or message 1307T will be issued and the system will be abnormally terminated, if it is necessary to avoid file corruption.
HNDINT	**Incompatible changes:** • CP's ATTACH processing has changed so that when a tape drive is attached to an XA, ESA, or XC virtual machine, CP presents, and therefore HNDINT can pass on, an I/O interruption indication that the virtual tape drive is ready. The interruption presented depends on the model of the tape device: – For 3420s: A device end interruption, X'04' – For other tapes: A state change, X'85' (which is attention + device end + unit exception). In VM/ESA 1.2.0, this did not occur in XA, ESA, or XC virtual machines. This did, and still does, occur in 370 virtual machines.
HNDIO	**Incompatible changes:** • CP's ATTACH processing has changed so that when a tape drive is attached to an XA, ESA, or XC virtual machine, CP presents, and therefore HNDIO can pass on, an I/O interruption indication that the virtual tape drive is ready. The interruption presented depends on the model of the tape device: – For 3420s: A device end interruption, X'04' – For other tapes: A state change, X'85' (which is attention + device end + unit exception). In VM/ESA 1.2.0, this did not occur in XA, ESA, or XC virtual machines. This did, and still does, occur in 370 virtual machines. **Upwardly compatible changes:** • Supports new parameter: PERSIST
REXEXIT	**Upwardly compatible changes:** • Supports new parameters: INVOKE and TYPE.
SEGMENT	**Incompatible changes:** • If you are trying to load or find a saved segment that is in the process of being saved, you get return code 44 and the saved segment is not loaded or found. In VM/ESA 1.2.0, your load or find request was delayed until the save was complete, and then the saved segment was loaded or found. • New return code for SEGMENT FIND: 41.
TAPECTL	**Upwardly compatible changes:** • 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.
TVISECT	**Upwardly compatible changes:** • New fields are supported: TVIFLAGS, TVIRECFM, TVILRECL, TVIBLKSI.

Compatibility Macros

See *VM/ESA: Planning and Administration* for information about the DEFNUC macro.

Table 71. CMS Compatibility Macros Changed since VM/ESA 1.2.0

Macro	Changes
DEFNUC	**Incompatible changes:** • The CYLADDR= parameter supports up-to-10 digits. In VM/ESA 1.2.0, it supported up to five. • The format of the default CMS IPL heading (used when the VERSION= parameter is specified without a value in DEFNUC) has changed to include the version. Also, the date in the default IPL heading is presented in ISO format (*yyyy-mm-dd*). • Changed messages: DMS1104R, DMS1105R. **Upwardly compatible changes:** • Supports new parameters: MDBUFSZ. • New message: DMS2104R. • The USEMTSEG and MTSEG parameters no longer have any effect because VMMTLIB is no longer created as a saved segment; it now resides within the CMS nucleus.

OS Simulation Macros and Supervisor Calls

See the *VM/ESA: CMS Application Development Guide for Assembler* for information about the OS macros and supervisor calls that CMS simulates.

Table 72. OS Simulation Macros Changed since VM/ESA 1.2.0

Macro	Changes
CLOSE	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
DCB	**Upwardly compatible changes:** • 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.
GET	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.
OPEN	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
PUT	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.
TIME	**Upwardly compatible changes:** • The 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.

Table 73. OS Simulation Supervisor Calls Changed since VM/ESA 1.2.0

SVC	Changes
SVC 19 (OPEN)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
SVC 20 (CLOSE)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.

Preferred Routines (CSL Routines)

Table 74 lists the CSL routines that have changed since VM/ESA 1.2.0. See the *VM/ESA: CMS Application Development Reference* for complete descriptions of all the routines in this table except QueueOpen. The QueueOpen routine is described in *VM/ESA: CMS Application Multitasking*.

Table 74 (Page 1 of 11). CSL Routines Changed since VM/ESA 1.2.0

Routine	Changes
DMSCATTR	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSCLBLK	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • New reason codes when using BFS: 10220, 65400. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330, 90492, 90495.
DMSCLDBK	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • New reason code when using BFS: 65400. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330, 90495.
DMSCLDIR	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation.

Table 74 (Page 2 of 11). CSL Routines Changed since VM/ESA 1.2.0

Routine	Changes
DMSCLOSE	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330, 90495.
DMSCRALI	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • Supports a new option: UNRESOLVED. • New reason codes: 61620, 98700. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSCRDIR	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSCRFIL	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* and *create_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSCRLOC	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New reason codes when using BFS: 65400, 69200, and 69300. • Supports new BFS parameter: *bfsid*.

Table 74 (Page 3 of 11). CSL Routines Changed since VM/ESA 1.2.0

Routine	Changes
DMSCROB	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* and *create_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSDELOC	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid*.
DMSDEUSR	**Upwardly compatible changes:** • New parameters: DELAUTH, KEEPAUTH, and *length4*. • New reason code: 98700.
DMSDIRAT	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSENUSR	**Upwardly compatible changes:** • New reason codes when using BFS: 10210, 10240, 69000, 90300, 90320, and 98700. • Supports new options: SFS | BFS, *userid*, *gname*. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE, *length7*. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90330, 90495.
DMSERASE	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid*.
DMSERP	**Upwardly compatible changes:** • Supports new functions: GETENV, RES:*envir*, REP:*envir*, EXT:*envir*. • New return code: 125. • Changed return codes: 112, 117, 124. • New information names for Year 2000 support (FILE_DATE_CENTURY, ACT_FILE_DATE_CENTRY (note that U is omitted), and YEAR2000_SUPPORT).

Table 74 (Page 4 of 11). CSL Routines Changed since VM/ESA 1.2.0

Routine	Changes
DMSEXIDI	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.0, it was returned as 8 characters. **Upwardly compatible changes:** • New parameters: *update_date*, *update_time*, *create_date*, *create_time*. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameters: *bfsid*, *filesystemtype* • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *last_change_date* and *create_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330.
DMSEXIFI	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.0, it was returned as 8 characters. **Upwardly compatible changes:** • New parameters: *update_date*, *update_time*. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameters: *bfsid*, *filesystemtype* • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date*, *dateref*, *create_date*, and *last_change_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330.

Table 74 (Page 5 of 11). CSL Routines Changed since VM/ESA 1.2.0

Routine	Changes
DMSEXIST	**Incompatible changes:**

- You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation.

Upwardly compatible changes:

- New parameters: *uniqueid_fpid*, UNIQUEID.
- New reason codes: 90476, 90481, 90486.
- Changed reason code: 90300.
- The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
- Supports new BFS parameter: *bfsid*
- The following offsets have changed in the FILE data record (and the record length has increased to 436 bytes):

OFFSET	**Field Name**	**Description of Change**
1 (X'1')	*file_system_type*	Reserved in VM/ESA 1.2.0
161 (X'A1')	*dirname_len*	Added
162 (X'A2')	*dirname*	Added
315 (X'13B')	*dec_update_date*	Added
318 (X'13E')	Reserved	Added
319 (X'13F')	*dec_update_time*	Added
322 (X'142')	Reserved	Added
323 (X'143')	*update_date*	Added
331 (X'14B')	*update_time*	Added
339 (X'153')	*dec_date_ext*	Added
343 (X'157')	*date_ext*	Added
353 (X'161')	*iso_date_ext*	Added
363 (X'16B')	*dec_dateref_ext*	Added
367 (X'16F')	*dateref_ext*	Added
377 (X'179')	*iso_dateref_ext*	Added
387 (X'183')	*dec_cr_date_ext*	Added
391 (X'187')	*cr_date_ext*	Added
401 (X'191')	*iso_cr_date_ext*	Added
411 (X'19B')	*dec_last_change_date_ext*	Added
415 (X'19F')	*last_change_date_ext*	Added
425 (X'1A9')	*iso_last_change_date_ext*	Added
435 (X'1B3')	Reserved	Added

(continued on next page)

Table 74 (Page 6 of 11). CSL Routines Changed since VM/ESA 1.2.0

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSEXIST - continued</td><td>
<b>Upwardly compatible changes (continued):</b>
<ul><li>The following offsets have changed in the DIRECTORY data record (and the record length has increased to 308 bytes):
<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>1 (X'1')</td><td><i>file_system_type</i></td><td>Reserved in VM/ESA 1.2.0</td></tr>
<tr><td>190 (X'BE')</td><td><i>unique_id</i></td><td>Added</td></tr>
<tr><td>206 (X'CE')</td><td><i>dec_update_date</i></td><td>Added</td></tr>
<tr><td>209 (X'D1')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>210 (X'D2')</td><td><i>dec_update_time</i></td><td>Added</td></tr>
<tr><td>213 (X'D5')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>214 (X'D6')</td><td><i>update_date</i></td><td>Added</td></tr>
<tr><td>222 (X'DE')</td><td><i>update_time</i></td><td>Added</td></tr>
<tr><td>230 (X'E6')</td><td><i>dec_cr_date</i></td><td>Added</td></tr>
<tr><td>233 (X'E9')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>234 (X'EA')</td><td><i>dec_cr_time</i></td><td>Added</td></tr>
<tr><td>237 (X'ED')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>238 (X'EE')</td><td><i>cr_date</i></td><td>Added</td></tr>
<tr><td>246 (X'F6')</td><td><i>cr_time</i></td><td>Added</td></tr>
<tr><td>254 (X'FE')</td><td><i>dec_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>258 (X'102')</td><td><i>last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>268 (X'10C')</td><td><i>iso_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>278 (X'116')</td><td><i>dec_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>282 (X'11A')</td><td><i>cr_date_ext</i></td><td>Added</td></tr>
<tr><td>292 (X'124')</td><td><i>iso_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>302 (X'12E')</td><td>Reserved</td><td>Added</td></tr>
</table>
</li></ul>
</td></tr>
<tr><td>DMSFILEC</td><td>
<b>Incompatible changes:</b>
<ul><li>You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation.</li></ul>
<b>Upwardly compatible changes:</b>
<ul><li>The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.</li></ul>
</td></tr>
<tr><td>DMSGETDA</td><td>
<b>Incompatible changes:</b>
<ul><li>When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.0, it was returned as 8 characters.</li></ul>
<b>Upwardly compatible changes:</b>
<ul><li>New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and <i>length2</i>.</li>
<li>The <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li>
<li>New reason codes: 90310, 90320, 90330.</li></ul>
</td></tr>
<tr><td>DMSGETDF</td><td>
<b>Incompatible changes:</b>
<ul><li>When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.0, it was returned as 8 characters.</li></ul>
<b>Upwardly compatible changes:</b>
<ul><li>New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and <i>length2</i>.</li>
<li>The <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.</li>
<li>New reason codes: 90310, 90320, 90330.</li></ul>
</td></tr>
</table>

Table 74 (Page 7 of 11). CSL Routines Changed since VM/ESA 1.2.0

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSGETDI</td><td><b>Upwardly compatible changes:</b>
<ul>
<li>The following offsets have changed in the FILE data record (and the record length has increased to 112 bytes):
<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>88 (X'58')</td><td><i>dec_date_ext</i></td><td>Added</td></tr>
<tr><td>92 (X'5C')</td><td><i>date_ext</i></td><td>Added</td></tr>
<tr><td>102 (X'66')</td><td><i>iso_date_ext</i></td><td>Added</td></tr>
</table></li>
<li>The following offsets have changed in the FILEEXT data record (and the record length has increased to 284 bytes):
<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>1 (X'1')</td><td><i>file_system_type</i></td><td>Reserved in VM/ESA 1.2.0</td></tr>
<tr><td>161 (X'A1')</td><td><i>dec_update_date</i></td><td>Reserved in VM/ESA 1.2.0</td></tr>
<tr><td>164 (X'A4')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>165 (X'A5')</td><td><i>dec_update_time</i></td><td>Added</td></tr>
<tr><td>168 (X'A8')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>169 (X'A9')</td><td><i>update_date</i></td><td>Added</td></tr>
<tr><td>177 (X'B1')</td><td><i>update_time</i></td><td>Added</td></tr>
<tr><td>185 (X'B9')</td><td><i>dec_date_ext</i></td><td>Added</td></tr>
<tr><td>189 (X'BD')</td><td><i>date_ext</i></td><td>Added</td></tr>
<tr><td>199 (X'C7')</td><td><i>iso_date_ext</i></td><td>Added</td></tr>
<tr><td>209 (X'D1')</td><td><i>dec_dateref_ext</i></td><td>Added</td></tr>
<tr><td>213 (X'D5')</td><td><i>dateref_ext</i></td><td>Added</td></tr>
<tr><td>223 (X'DF')</td><td><i>iso_dateref_ext</i></td><td>Added</td></tr>
<tr><td>233 (X'E9')</td><td><i>dec_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>237 (X'ED')</td><td><i>cr_date_ext</i></td><td>Added</td></tr>
<tr><td>247 (X'F7')</td><td><i>iso_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>257 (X'101')</td><td><i>dec_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>261 (X'105')</td><td><i>last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>271 (X'10F')</td><td><i>iso_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>281 (X'119')</td><td>Reserved</td><td>Added</td></tr>
</table></li>
<li>The following offset has changed in the LOCK data record:
<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>1 (X'1')</td><td><i>file_system_type</i></td><td>Reserved in VM/ESA 1.2.0</td></tr>
</table></li>
<li>The following offsets have changed in the SEARCHALL and SEARCHAUTH data records (and the record length has increased to 252 bytes):
<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>226 (X'E2')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>228 (X'E4')</td><td><i>dec_date_ext</i></td><td>Added</td></tr>
<tr><td>232 (X'E8')</td><td><i>date_ext</i></td><td>Added</td></tr>
<tr><td>242 (X'F2')</td><td><i>iso_date_ext</i></td><td>Added</td></tr>
</table></li>
</ul></td></tr>
<tr><td>DMSGETDK</td><td><b>Upwardly compatible changes:</b>
<ul>
<li>Supports new BFS parameter: <i>filesystemtype</i></li>
</ul></td></tr>
</table>

Table 74 (Page 8 of 11). CSL Routines Changed since VM/ESA 1.2.0

Routine	Changes
DMSGETDS	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.0, it was returned as 8 characters. **Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330.
DMSGETDX	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.0, it was returned as 8 characters. **Upwardly compatible changes** • Supports new parameters: *update_date*, *update_time*. • Supports new BFS parameter: *filesystemtype* • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*. • The *date*, *dateref*, *create_date*, and *update_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330.
DMSGRANT	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSOPBLK	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.0, it was returned as 8 characters. **Upwardly compatible changes:** • Supports a new option: RESOLVE. • Supports new parameters: CREATEMIG, *unique_id*, *userid*, *dateref*, *bfsid* • New reason codes: 30000, 63800, 98700. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New reason codes when using BFS: 10220, 65400, 98700. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date*, *create_date*, and *dateref* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSOPCAT	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid* • Supports new BFS options: FILEATTR SFS|BFS, READEXT • New reason codes when using BFS: 90300, 90350.

Table 74 (Page 9 of 11). CSL Routines Changed since VM/ESA 1.2.0

Routine	Changes
DMSOPDBK	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid* • New reason code when using BFS: 10220. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *create_date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSOPDIR	**Upwardly compatible changes:** • Supports new parameters: *buffersize*, *number*. • New reason codes: 30000, 63800, 98700. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid* • New reason code when using BFS: 65400.
DMSOPEN	**Upwardly compatible changes** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid* • New reason code when using BFS: 10220. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *create_date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSQEFL	**Upwardly compatible changes:** • Supports new values for *cp_level* and *cms_level*
DMSQLIMA	**Incompatible changes:** • File pool administration authority no longer required. **Upwardly compatible changes:** • Supports new BFS parameter: *filesystemtype*
DMSQLIMD	**Upwardly compatible changes:** • Supports new BFS parameter: *filesystemtype*
DMSQLIMU	**Incompatible changes:** • File pool administration authority no longer required. **Upwardly compatible changes:** • Supports new BFS parameter: *filesystemtype*
DMSQSFSL	**Upwardly compatible changes:** • Supports new value for *server_level*
DMSQOBJ	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 74 (Page 10 of 11). CSL Routines Changed since VM/ESA 1.2.0

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSRDCAT</td><td>
<b>Upwardly compatible changes:</b>
<ul>
<li>New ACAT record for unresolved aliases.</li>
<li>New catalogs for BFS support: NAMECAT, NOVCAT.</li>
<li>New OBJECTCAT record for BFS object.</li>
<li>SPACECAT record — the four reserved CHAR(8) fields at the end of the record (following the STORAGEGROUP field) have been restructured for BFS support as follows:
<table>
<tr><th>Field Name</th><th>Field Type/Description</th></tr>
<tr><td>HIGHINO</td><td>INTEGER(4) High OBJECTCAT INO value</td></tr>
<tr><td>HIGHNID</td><td>INTEGER(4) High NAMECAT NID value</td></tr>
<tr><td>FLAGS</td><td>CHAR(1) File space attributes</td></tr>
<tr><td>RESERVED01</td><td>CHAR(7) Reserved</td></tr>
<tr><td>RESERVED02</td><td>CHAR(8) Reserved</td></tr>
<tr><td>RESERVED03</td><td>CHAR(8) Reserved</td></tr>
</table>
</li>
<li>DIRCAT record — new bit settings in the DIRATTS field for BFS support.</li>
<li>OBJECTCAT record for SFS:
<ul>
<li>New bit settings in TYPE and FILEFLAGS fields for BFS support.</li>
<li>New bit settings in FILEFLAGS field to support 4-digit years (century setting for DATE and DATEREF fields).</li>
<li>The fields following the BSCID field are:
<table>
<tr><th>Field Name</th><th>Field Type/Description</th></tr>
<tr><td></td><td>CHAR(8) reserved</td></tr>
<tr><td>CHGDATE_CENTURY</td><td>CHAR(1) Century byte for LAST_CHANGE_DATE</td></tr>
<tr><td>LAST_CHANGE_DATE</td><td>CHAR(3) UTC date of last change</td></tr>
<tr><td></td><td>CHAR(1) reserved</td></tr>
<tr><td>LAST_CHANGE_TIME</td><td>CHAR(3) UTC time of last change</td></tr>
<tr><td>DRA1</td><td>CHAR(8) DRA1</td></tr>
<tr><td>DRA2</td><td>CHAR(8) DRA2</td></tr>
<tr><td>DRA3</td><td>CHAR(8) DRA3</td></tr>
<tr><td></td><td>INTEGER(4) reserved</td></tr>
<tr><td>CREATIONDATE_CENTURY</td><td>CHAR(1) Century byte for CREATIONDATE</td></tr>
<tr><td>CREATIONDATE</td><td>CHAR(3) UTC date of file creation</td></tr>
<tr><td></td><td>CHAR(1) reserved</td></tr>
<tr><td>CREATIONTIME</td><td>CHAR(3) UTC time of file creation</td></tr>
<tr><td>DATEREF</td><td>CHAR(3) UTC date of last reference</td></tr>
</table>
</li>
</ul>
</li>
<li>FQFN record — new H type record for open intent READEXT.</li>
<li>AUTHCAT record — new G type record for open intent READEXT.</li>
<li>EOCAT record — new bit settings in TYPE field for BFS support.</li>
</ul>
</td></tr>
<tr><td>DMSRELOC</td><td>
<b>Upwardly compatible changes:</b>
<ul>
<li>The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.</li>
</ul>
</td></tr>
</table>

Table 74 (Page 11 of 11). CSL Routines Changed since VM/ESA 1.2.0

Routine	Changes
DMSRENAM	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSREVOK	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSTRUNC	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSUDATA	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSVALDT	**Upwardly compatible changes:** • Supports new parameters: *namedef1*, *namedef2*. • New reason codes: 90510, 90530, 96100. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid*.
QueueOpen	**Upwardly compatible changes:** • In VM/ESA 1.2.0, if you tried to use QueueOpen to open a remotely-located network-level queue but forgot to set up the CMS Communication Directory (that is, installed no ComDir entry describing the location of the remote queue), then QueueOpen would fail with reason code *vm_ipc_comm_lost*. In VM/ESA 2.4.0, QueueOpen will proceed anyway, using default values for what it tried to extract from the ComDir. The default values used are described in *VM/ESA: CMS Application Multitasking*.

CSLCNTRL File

The CSL control file has new and revised statements, which are summarized in Table 75.

Table 75. CSLCNTRL File Changes since VM/ESA 1.2.0

Line	Explanation
ROUTINE	**Upwardly compatible changes:** • Supports new options: FILETYPE, COPYTYPE, MAP.
ALIAS	**Upwardly compatible changes:** • Supports new option: COPYTYPE.
TEXT	**Upwardly compatible changes:** • Supports new option: FILETYPE.
INCLUDE	New statement

Compatibility Routines

Table 76 lists the CMS compatibility routines that have changed between VM/ESA 1.2.0 and VM/ESA 2.4.0. See the *VM/ESA: CMS Application Development Guide for Assembler* for more information about the routines in this table.

Table 76. CMS Compatibility Routines Changed since VM/ESA 1.2.0

CMS Routine	Explanation
DMSTVS	**Upwardly compatible changes:** • Supports new LIBSRV plist parameter. • 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.

CMS Pipelines

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.
- 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.

  Note: User-written applications that are sensitive to these changes may require alterations. Published externals (such as PIPGFTXT) have **not** been changed.
- 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. Some specific enhancements that have occurred since VM/ESA 2.1.0 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 *VM/ESA: CMS Pipelines Reference* can be found in the *CMS/TSO Pipelines: Author's Edition*, which is included with the VM/ESA 2.4.0 library.

CMS Messages

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 H, “CMS Pipelines Message Cross-Reference [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]” on page 669.

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/ESA.

The following CMS message does not exist in VM/ESA 2.4.0:

DMS683E

The following CMS messages have changed since VM/ESA 1.2.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CMS messages.

DMS002E
DMS003E
DMS005E
DMS007E
DMS017E
DMS023E
DMS024E
DMS026E
DMS029E
DMS033E
DMS036E
DMS037E
DMS042E
DMS054E
DMS056E
DMS062E
DMS069I
DMS081E
DMS106S
DMS107S
DMS132S
DMS149E
DMS165S
DMS250E
DMS333E
DMS344E
DMS358E
DMS360E
DMS361E
DMS363R
DMS364I
DMS365I
DMS366R
DMS389E
DMS516E
DMS531E
DMS595E
DMS616W
DMS618E
DMS622E
DMS639E
DMS647E
DMS651E
DMS660W
DMS1007E
DMS1096E
DMS1104R
DMS1105E
DMS1127I
DMS1131E
DMS1137E
DMS1138E
DMS1139E
DMS1146E
DMS1151E
DMS1153E
DMS1162E
DMS1176E
DMS1184E
DMS1205I
DMS1227E
DMS1229E
DMS1239E
DMS1258E
DMS1259E
DMS1306T
DMS1307T
DMS1311E
DMS1339S
DMS1341S
DMS1342S
DMS1343S
DMS1344S
DMS2008E
DMS2010E
DMS2013E
DMS2023E
DMS2031E
DMS2040E
DMS2041E
DMS2055I
DMS2153E
DMS2154E
DMS2155E
DMS2501E
DMS2511W
DMS2523E
DMS2727E
DMS2731I
DMS2805E
DMS2883E
DMS2897E
DMS2982E
DMS3000W
DMS3088R
DMS3110E
DMS3208E
DMS3221E
DMS3284E
DMS3431E
DMS3438E
DMS3438I
DMS3453E
DMS3454E
DMS3455I
DMS3470W
DMS3494R
DMS3508E
DMS3514E
DMS3515E
DMS3518E
DMS3521E
DMS3522E
DMS3524E
DMS3526E
DMS3527E
DMS3528E
DMS3529E
DMS3530E
DMS3531E
DMS3532E
DMS3557E
DMS3561E
DMS3562E
DMS3585E
DMS3594R
DMS3608E
DMS3616W
DMS3617I
DMS3618W
DMS3620I
DMS3622E

DMS3623E
DMS3624W
DMS3627E
DMS3628E
DMS3629E
DMS3630W
DMS3631W
DMS3632W
DMS3636E
DMS3641W
DMS3642W
DMS3727E
DMS3728E
DMS3926E

REXX/VM Changes

This section identifies REXX/VM externals that have changed since VM/ESA 1.2.0. It contains the following subsections:

- “REXX/VM Instructions”
- “REXX/VM Functions”
- “External Functions” on page 500

REXX/VM Instructions

Table 77 lists the REXX/VM instructions that have changed since VM/ESA 1.2.0. For complete descriptions of these instructions, see the *VM/ESA: REXX/VM Reference*.

Table 77. REXX/VM Instructions Changed since VM/ESA 1.2.0

Instruction	Changes
CALL	**Upwardly compatible changes:** • Added conditions: OFF NOTREADY, ON NOTREADY.
PARSE	**Upwardly compatible changes:** • Added a new variant: PARSE LINEIN.
SIGNAL	**Upwardly compatible changes:** • Added conditions: OFF NOTREADY, ON NOTREADY.

REXX/VM Functions

Two new functions have been added to REXX that may conflict with functions you have created for yourself. These are B2X and X2B, which translate binary to hexadecimal and hexadecimal to binary. They return a string in character format.

Table 78 lists the REXX/VM functions that have changed since VM/ESA 1.2.0. For complete descriptions of these functions, see the *VM/ESA: REXX/VM Reference*.

Table 78. REXX/VM Functions Changed since VM/ESA 1.2.0

Function	Changes
DATE	**Upwardly compatible changes:** • New parameters allow you to specify a date to be converted to a different format. • New parameters: *output_separator_char*, *input_separator_char*.

External Functions

Table 79 lists external functions which can be used by REXX/VM that have changed since VM/ESA 1.2.0. For complete descriptions of these functions, see the *VM/ESA: REXX/VM Reference*.

Table 79. External Functions Changed since VM/ESA 1.2.0

Function	Changes
CMSFLAG	**Upwardly compatible changes:** • New value for *flag*: YEAR2000.
DIAG DIAGRC	**Upwardly compatible changes:** • New DIAGNOSE code is supported: X'270'.

GCS Changes

This section identifies the GCS externals that have changed since VM/ESA 1.2.0. It contains the following subsection:

- "GCS Commands and Macros"

GCS Commands and Macros

Table 80 lists the GCS commands and macros that have changed since VM/ESA 1.2.0. See *VM/ESA: Group Control System* for complete descriptions of all GCS commands and macros.

Table 80 (Page 1 of 2). GCS Commands and Macros Changed since VM/ESA 1.2.0

Command/Macro	Changes
CVT macro	**Upwardly compatible changes:** • Added the fields: CVTECVT, CVTFLAG2.
GCSLEVEL macro	**Upwardly compatible changes:** • Additional equates for release levels.
GETMAIN macro	**Incompatible changes:** • In VM/ESA 1.2.0, although LOC=RES was documented as the default, the actual default was LOC=BELOW, and all the requested virtual storage was allocated below 16MB. In VM/ESA 2.2.0, the actual default was 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.
ITRACE command	**Upwardly compatible changes:** • New operand: SP.
QUERY ADDRESS command	**Upwardly compatible changes:** • Operand *variable* changed. It now supports either *name* or *address*.
QUERY GCSLEVEL command	**Incompatible changes:** • The format of the response has changed to include the version and the service level: `VM/ESA Version 2 Release 4.0, Service Level 0`

Table 80 (Page 2 of 2). GCS Commands and Macros Changed since VM/ESA 1.2.0

Command/Macro	Changes
QUERY MODDATE command	**Incompatible changes:** • A full 4-digit year is now returned in the date field of the response instead of a 2-digit year.
TIME macro	**Upwardly compatible changes:** • The 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.

Dump Viewing Facility Changes

This identifies the Dump Viewing Facility externals that have changed since VM/ESA 1.2.0. It contains the following subsection:

- “Dump Viewing Facility Commands”

Dump Viewing Facility Commands

Table 81 lists the Dump Viewing Facility commands that have changed between VM/ESA 1.2.0 and VM/ESA 2.4.0. See *VM/ESA: Dump Viewing Facility* for complete descriptions of Dump Viewing Facility commands.

Note: The Dump Viewing Facility in VM/ESA 2.4.0 can process only dumps produced by VM/ESA 2.4.0.

Table 81. Dump Viewing Facility Commands Changed since VM/ESA 1.2.0

Dump Viewing Facility Command	Explanation
DUMPSCAN	**Upwardly compatible changes:** • The FINDUSER subcommand supports a new parameter: AT *address*. • ASID and ACCLIST subcommands support CP abend dumps, snapdumps, and stand-alone dumps.
TRACERED	**Upwardly compatible changes:** • New operands: FILE *fn1 ft1 fm1* ...FILE *fn5 ft5 fm5*. • New operands can cause new messages.

VMSES/E Changes

This section identifies the VMSES/E externals that have changed since VM/ESA 1.2.0. It contains the following subsection:

- “VMSES/E Messages”

VMSES/E Messages

The following VM/ESA 1.2.0 message does not exist in VM/ESA 2.4.0:

VMF2709E

The following VMSES/E messages have changed since VM/ESA 1.2.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of VMSES/E messages.

VMF002E
VMF066E
VMF071E
VMF395E
VMF1015E
VMF1016E
VMF1059E
VMF1068E
VMF1444I
VMF1507E
VMF1821E
VMF1832W
VMF1836E
VMF1851E
VMF1851I
VMF1861E
VMF1861I
VMF1861W
VMF1868W
VMF1905E
VMF1908E
VMF1909I
VMF1944E
VMF1953W
VMF2066E
VMF2070E
VMF2111E
VMF2112I
VMF2114R
VMF2118I
VMF2119I
VMF2120W
VMF2152E
VMF2162E
VMF2173I
VMF2181E
VMF2200E
VMF2201E
VMF2202E
VMF2206W
VMF2214E
VMF2215E
VMF2218I
VMF2225E
VMF2228E
VMF2242E
VMF2500E
VMF2507I
VMF2704E
VMF2707I
VMF2709E
VMF2725I
VMF2740E
VMF2760I
VMF2780E
VMF2786E
VMF2830E
VMF2831E
VMF2835E
VMF2866I
VMF2867W
VMF2930E

Programmable Operator Facility Changes

This section identifies the Programmable Operator Facility externals that have changed since VM/ESA 1.2.0. It contains the following subsection:

- "Programmable Operator Facility Commands and Routing Table Statements"

Also see "Summary of Service Exec Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]" on page 250.

Programmable Operator Facility Commands and Routing Table Statements

Table 82 lists the Programmable Operator Facility commands and routing table statements that have changed between VM/ESA 1.2.0 and VM/ESA 2.4.0. See *VM/ESA: Planning and Administration* for complete descriptions of programmable operator facility commands and routing table statements.

Table 82. Programmable Operator Facility Commands and Routing Table Statements Changed since VM/ESA 1.2.0

Programmable Operator Command or Statement	Explanation
MSGLIMIT	**Upwardly compatible changes:** • New routing table configuration entry statement.

Chapter 19. Compatibility Tables for Converting from VM/ESA 1.2.1

This chapter identifies the VM/ESA externals that have changed between VM/ESA 1.2.1 and VM/ESA 2.4.0. It contains the following major sections:

- "CP Changes"
- "CMS Changes" on page 513
- "REXX/VM Changes" on page 539
- "GCS Changes" on page 540
- "Dump Viewing Facility Changes" on page 541
- "VMSES/E Changes" on page 542
- "Programmable Operator Facility Changes" on page 542

Use the information provided in this chapter to determine if you need to make any changes in the way you use these functions. See "Terminology and Symbols" on page 3 for the meaning of compatibility terms used in the tables.

CP Changes

This section identifies the CP externals that have changed since VM/ESA 1.2.1. It contains the following subsections:

- "CP Commands"
- "SYSTEM CONFIG Statements" on page 509
- "User Directory Control Statements" on page 509
- "CP Utilities" on page 511
- "CP DIAGNOSE Codes" on page 511
- "CP Macros" on page 512
- "CP Messages" on page 513

CP Commands

Table 83 lists the CP commands that have changed since VM/ESA 1.2.1. Refer to the *VM/ESA: CP Command and Utility Reference* for complete descriptions of CP commands.

Table 83 (Page 1 of 7). CP Commands Changed since VM/ESA 1.2.1

Command	Changes
AUTOLOG	**Incompatible changes:** • Users defined with the LBYONLY operand in the password field of the USER directory control statement cannot be logged on by the AUTOLOG command. **Upwardly compatible changes:** • Supports the System Console.
CPLISTFILE	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses include 4-digit years for FULLDATE and ISODATE operands.

Table 83 (Page 2 of 7). CP Commands Changed since VM/ESA 1.2.1

Command	Changes
CPU	**Incompatible changes:** • On nested CPU commands, multiple *cpuaddr*s for virtual CPUs are valid on only one of the commands. VM/ESA 1.2.1 supported multiple CPUs on multiple nested CPU commands. • New message: HCP1460E.
DEFINE (in general)	**Upwardly compatible changes:** • Supports new operand: MSGPROC. • Additional messages: HCP045E, HCP260E, HCP1014E, HCP2800E, HCP2801E, HCP2802E, HCP2803E, HCP2804I, HCP2806E, HCP2811E. • See DEFINE commands below.
DEFINE CRYPTO	**Upwardly compatible changes:** • New response. • New message: HCP1716E.
DETACH (in general)	**Upwardly compatible changes:** • Supports new operand: MSGPROC. • Additional messages: HCP260E, HCP2805E, HCP2807E.
DISCONN	**Upwardly compatible changes:** • Supports the System Console.
DISPLAY Linkage Stack	**Upwardly compatible changes:** • Response indicates the called-space identification (CSID) if the linkage-stack entry type is a program-call state entry with a called-space ID.
DISPLAY Registers	**Upwardly compatible changes:** • Supports new operand: FPC. • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New response when FPC operand is used. • New messages: HCP6153E, HCP6154E.
DUMP Linkage Stack	**Upwardly compatible changes:** • Response indicates the called-space identification (CSID) if the linkage-stack entry type is a program-call state entry with a called-space ID.
IPL	**Incompatible changes:** • A new response is issued if tracing is active when a preferred guest is IPLed. Tracing must be turned off and the guest reIPLed. • In VM/ESA 1.2.1, when you IPL CMS with the PARM operand, CMS initialization attaches a fence of 8 bytes of `X'FF'`s to the end of the PARM data before passing it to the SYSPROF EXEC. In VM/ESA 2.4.0, 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.
LOGOFF or LOGOUT	**Upwardly compatible changes:** • Supports the System Console.

Table 83 (Page 3 of 7). CP Commands Changed since VM/ESA 1.2.1

Command	Changes
LOGON or LOGIN	**Incompatible changes:** • A LOGON of a user with LBYONLY in the password field of the USER directory control statement can only be done by an authorized user using the new BY option of the LOGON command. **Upwardly compatible changes:** • Supports the System Console. • New message: HCP2808E.
QUERY (in general)	**Incompatible changes:** • See QUERY commands below. **Upwardly compatible changes:** • See QUERY commands below.
QUERY ABEND	**Upwardly compatible changes:** • Supports a new operand: SNAPDUMP.
QUERY CACHEFW	**Upwardly compatible changes:** • Response indicates if the cache fast write function is suspended for the subsystem.
QUERY CHPID	**Upwardly compatible changes:** • New operand: TYPE. • New responses if TYPE is specified.
QUERY CPLEVEL	**Incompatible changes:** • The format of the response has changed to include the version. In addition, the service level is returned as *yynn*, where *yy* is the last two digits of the year and *nn* is the sequential number of the RSU tape for that year. **Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • In the response, the release level value has changed. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q CPLEVEL command, the output from Q 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.
QUERY CRYPTO	**Upwardly compatible changes:** • Supports new operand: CAMQS. • Two new responses.
QUERY DASDFW	**Upwardly compatible changes:** • Response indicates if the DASD fast write function is suspended for the subsystem.
QUERY IMG	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q IMG command, the output from Q 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.
QUERY LDEVS	**Upwardly compatible changes:** • The response may include the IP address for TCP/IP.

Table 83 (Page 4 of 7). CP Commands Changed since VM/ESA 1.2.1

Command	Changes

QUERY NAMES	**Incompatible changes:**
	• Response has changed:
	– Can contain new responses if your installation is using System Console support:
	`LOGNSYC   - SYSC`
	`userid   - SYSC`
	Upwardly compatible changes:
	• Supports the EXTended parameter to display the network qualifiers.

QUERY NLS	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q NLS command, the output from Q 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.
QUERY NSS	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q NSS command, the output from Q 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.
QUERY READER / PRINTER / PUNCH	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses include 4-digit years for FULLDATE and ISODATE operands. • 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. • 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 Q RDR|PRT|PUN command, the output from Q 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).
QUERY TIME	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q TIME command, the output from Q TIME uses the new default date format. This causes the date to be expanded to include the 4-digit year.
QUERY TRACE	**Incompatible changes:** • Guest-use of the Subspace-Group Facility is supported only when the machine in which CP is running provides it. In VM/ESA 1.2.1, CP would try to run Subspace-Group Facility whether or not the machine in which CP was running supported it. • Response may have added information.

Table 83 (Page 5 of 7). CP Commands Changed since VM/ESA 1.2.1

Command	Changes
QUERY TRFILES	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q TRFILES command, the output from Q 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.
QUERY TRSOURCE	**Upwardly compatible changes:** • Response may contain additional information. • New message: HCP6163E.
QUERY UCR	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q UCR command, the output from Q 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.
QUERY USERID	**Upwardly compatible changes:** • The response can be different if you are using the System Console: `userid   - SYSC` • Supports the EXTended parameter to display the network qualifiers.
QUERY USERS	**Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL ALL	**Upwardly compatible changes:** • Supports virtual message processors.
QUERY VIRTUAL CONSOLE	**Incompatible changes:** • The response may include a new line containing TCP/IP information. **Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL CRYPTO	**Upwardly compatible changes:** • New response.
QUERY (Virtual Device)	**Upwardly compatible changes:** • Response indicates REAL-MPLF or SIMULATED-MPLF if enabled for the device. • Supports virtual message devices. The device type MSGD appears in the response.
QUERY VIRTUAL GRAF	**Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL LINES	**Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
SET (in general)	**Upwardly compatible changes:** • See SET commands below.
SET ABEND	**Upwardly compatible changes:** • Supports a new operand: SNAPDUMP.

Table 83 (Page 6 of 7). CP Commands Changed since VM/ESA 1.2.1

Command	Changes
SET CPTRACE	**Upwardly compatible changes:** • Output from the ALLcodes operand may include type 3 trace codes. • New trace category and trace codes for QDIO instructions.
SET CRYPTO	**Upwardly compatible changes:** • New operands: MODIFY, ON, OFF. • Changed responses. • Changed messages: HCP1706I, HCP1709E, HCP1710E, HCP1711I. • New messages: HCP1714E, HCP1715E.
SET SHARED	**Upwardly compatible changes:** • New message: HCP2007E.
SPTAPE	**Upwardly compatible changes:** • The first two digits of the 4-digit year are included in hexadecimal format in the SFBLOK dumped to tape.
STORE (Registers)	**Upwardly compatible changes:** • Supports new operand: FPC *hexword.* • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New messages: HCP6153E, HCP6154E.
STORE STATUS	**Upwardly compatible changes:** • 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.
TRACE (in general)	**Incompatible changes:** • New message: HCP1038E. • See TRACE commands below. **Upwardly compatible changes:** • New command responses. • See TRACE commands below.
TRACE BRANCH	**Incompatible changes:** • More information about address spaces may be trapped when address space options are used (PRI, SECO, ANS, AREG, ALET).
TRACE *mnemonic1*	**Upwardly compatible changes:** • Supports new mnemonics: BSA, BSG.
TRACE *mnemonic2*	**Upwardly compatible changes:** • New mnemonic: SIGA.
TRACE STORE	**Incompatible changes:** • More information about address spaces may be trapped when address space options are used (PRI, SECO, ANS, AREG, ALET).
TRSOURCE	**Upwardly compatible changes:** • New messages: HCP6160E, HCP6161E, HCP6162E, HCP6163E.
TRSOURCE ID	**Upwardly compatible changes:** • Supports new conditional operands for system trace type DATA: IF, THEN, ELSE, ENDIF.

Table 83 (Page 7 of 7). CP Commands Changed since VM/ESA 1.2.1

Command	Changes
VARY	**Upwardly compatible changes:** • A new response is displayed if the PATH is OFFLINE because of a request initiated by a control unit.
XAUTOLOG	**Incompatible changes:** • Users defined with the LBYONLY operand in the password field of the USER directory control statement cannot be logged on by the XAUTOLOG command if the password validation is required. **Upwardly compatible changes:** • Supports the System Console.

SYSTEM CONFIG Statements

Table 84 lists SYSTEM CONFIG file statements that have changed since VM/ESA 1.2.1. See *VM/ESA: Planning and Administration* for complete descriptions of SYSTEM CONFIG statements.

Table 84. SYSTEM CONFIG Statements Changed since VM/ESA 1.2.1

Statement	Changes
CHARACTER_DEFAULTS	**Incompatible changes:** • Letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be defined as the system default line edit symbols (line-delete, character-delete, escape, line-end, and tab).
EMERGENCY_MESSAGE_CONSOLES	**Upwardly compatible changes:** • Supports a new operand: SYSTEM_CONSOLE.
OPERATOR_CONSOLES	**Upwardly compatible changes:** • Supports a new operand: SYSTEM_CONSOLE.

User Directory Control Statements

Table 85 lists the user directory control statements that have changed since VM/ESA 1.2.1. See the *VM/ESA: Planning and Administration* book for complete descriptions of user directory control statements.

Note: You can use the VMUDQ macro to query the CP user directory. See the *VM/ESA: CP Programming Services* book for more information on this macro.

Table 85 (Page 1 of 2). User Directory Control Statements Changed since VM/ESA 1.2.1

Statement	Changes
CRYPTO	**Upwardly compatible changes:** • New operand: MODIFY.
DASDOPT	**Upwardly compatible changes:** • Supports a new operand: WRKALLEG. • Supports the NOWRKALLEG option.

Table 85 (Page 2 of 2). User Directory Control Statements Changed since VM/ESA 1.2.1

Statement	Changes
DATEFORMAT	**New** Specifies a user's default date format for commands that provide multiple date formats.
GLOBALDEFS	**New** Signifies the beginning of the global definition section.
LOAD	**Upwardly compatible changes:** • Supports a new G operand for global definitions.
MINIOPT	**Upwardly compatible changes:** • Supports new operands: NOWRKALLEG, WRKALLEG.
OPTION	**Upwardly compatible changes:** • Current LKFAC operand authorizes full-pack minidisks and devices for real MPLF use. For dedicated devices, MPLF channel commands my now success where they used to fail. For full-pack minidisks, the change is transparent until you issue the new SET LKFACR command. • Specifying the TODENABLE operand allows a user to change the virtual machine TOD clock with the new CP SET VTOD command. • Supports new operands: CFVM, CFUSER, DIAG88.
POSIXGLIST	**New** Specifies all POSIX groups of which the user is a member.
POSIXGROUP	**New** Defines a POSIX group.
POSIXINFO	**New** Specifies a user's POSIX information.
POSIXOPT	**New** Specifies option settings related to a user's POSIX capabilities.
SPECIAL	**Upwardly compatible changes:** • Supports new operand: MSGPROC.
STDEVOPT	**Upwardly compatible changes:** • Supports new DASDSYS operands: DATAMOVER and NODATAMOVER.
USER	**Incompatible changes:** • User IDs LOGNSYSC and SYSC are reserved for CP use. • LBYONLY is a new operand that can be specified in the password field of the USER statement. Any user defined with the LBYONLY operand may be restricted from performing operations that require password validation (including LOGON, AUTOLOG, XAUTOLOG, and DIAGNOSE X'84'). • Letters A-Z, numbers 0-9, and bytes `X'OE'` (shift out) and `X'0F'` (shift in) cannot be defined as logical line edit symbols (line-end, line-delete, character-delete, and escape).

CP Utilities

Table 86 lists the utility programs that have changed since VM/ESA 1.2.1. See the *VM/ESA: CP Command and Utility Reference* for details on utility programs.

Table 86. Utility Programs Changed since VM/ESA 1.2.1

Utility	Changes
DIRECTXA	**Incompatible changes:** • The format of the response has changed to include the version. • 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'0F' (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.
SALIPL	**Incompatible changes:** • Message HCP039E deleted, replaced by new message HCP394E with same text. • Date field on the file list panel displays the year with 4 digits.

CP DIAGNOSE Codes

Table 87 lists the DIAGNOSE codes that have changed since VM/ESA 1.2.1. See the *VM/ESA: CP Programming Services* book for complete information on the DIAGNOSE codes discussed in this section.

Table 87 (Page 1 of 2). DIAGNOSE Codes Changes since VM/ESA 1.2.1

Code	Changes
X'00'	Storage extended identification code **Incompatible changes:** • The Environment field has changed from 3 bytes to 2 bytes. • The new Version Information field (formerly the third byte of the Environment field) identifies the version number of the product (for VM/ESA 2.1.0 and later). **Upwardly compatible changes:** • The value in the program product bit map has changed to indicate the new release level. Also, Bit 13 (X'0004000000000000') indicates whether Year 2000 support is present in CP.
X'14'	Input spool file manipulation **Upwardly compatible changes:** • For subcodes X'0004', X'0008', X'0FFE', and X'0FFF', a one-byte century indicator was added to the SFBLOK data area.
X'68'	Virtual Machine Communication Facility (VMCF) **Upwardly compatible changes:** • Supports a new function: SETLIMIT (Subcode X'000C').
X'7C'	Logical Device Support Facility **Upwardly compatible changes:** • 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.

Table 87 (Page 2 of 2). DIAGNOSE Codes Changes since VM/ESA 1.2.1

Code	Changes
X'84'	Directory Update-in-Place **Incompatible changes:** • Users defined with LBYONLY in the password field of the USER directory control statement cannot be the target of DIAGNOSE code X'84' operations unless the virtual machine issuing the diagnose has the D84NOPAS option in its directory entry and the operation specified is not LOGPASS or MDISK. • For EDITCHAR operation, letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be specified as logical line edit symbols (line-end, line-delete, character-delete, and escape). **Upwardly compatible changes:** • Can now replace the following new information: – User's default date format setting • New operation: DATEFMT. • New return codes due to new function (in hex): 122, 123, 124.
X'BC'	Open and Query Spool File Characteristics **Upwardly compatible changes:** • 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.
X'D8'	Read spool file blocks on the system spool file queues **Upwardly compatible changes:** • For subcode X'0000', a one-byte century indicator was added to the SFBLOK data area.

CP Macros

Table 88 lists the IUCV macro functions for use in APPC/VM that have changed since VM/ESA 1.2.1.

The information in this table is based on two assumptions:

1. You have used the parameter lists as documented in the publications (that is, you have not used any undefined parts of the parameter list for your own purposes).
2. When you are converting your APPC/VM applications, your communications partners are in the same environment they were in before your conversion (that is, your communications partners are not exploiting new function).

See the *VM/ESA: CP Programming Services* book for complete descriptions of APPC/VM and IUCV macros.

Table 88. CP Macros Changed since VM/ESA 1.2.1

Macro Function	Changes
IUCV CONNECT	**Upwardly compatible changes:** • New optional parameters supported: TARGET=*address* LOCAL=YES|NO

CP Messages

The following CP messages do not exist in VM/ESA 2.4.0:

HCP543A
HCP1365E
HCP8039S
HCP8611T

The following list identifies the CP messages that have changed since VM/ESA 1.2.1. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CP messages.

Action indicators (for example, the “E” in message HCP006E) in VM/ESA 2.4.0 messages may differ from the corresponding messages in VM/ESA 1.2.1, even though the message number may be the same.

The nonzero return code and accompanying message that VM/ESA 2.4.0 returns for errors may not be the same as the nonzero return code that VM/ESA 1.2.1 returns for corresponding errors.

VM/ESA 2.4.0 suppresses leading zeros in responses more often than VM/ESA 1.2.1 does.

HCP006E
HCP031E
HCP054E
HCP114E
HCP145I
HCP233E
HCP296E
HCP319E
HCP332E
HCP403I
HCP481E
HCP512I
HCP574I
HCP580I
HCP592I
HCP704E
HCP711D
HCP716D
HCP717D
HCP725D
HCP751E
HCP752E
HCP753E
HCP772E
HCP799E
HCP904W
HCP1003E
HCP1006E
HCP1010W
HCP1011E
HCP1016E
HCP1018E
HCP1115E
HCP1116E
HCP1120E
HCP1365E
HCP1512E
HCP1706I
HCP1709E
HCP1710E
HCP1711I
HCP1791E
HCP1879E
HCP2002I
HCP2234E
HCP2252E
HCP2760E
HCP2768E
HCP2779E
HCP6014I
HCP6111I
HCP6283I
HCP6706E
HCP6739E
HCP6743E
HCP6788E
HCP6802E
HCP6804E
HCP8028W
HCP8080I
HCP9020W
HCP9021W
HCP9022W
HCP9036W
HCP9225I
HCP9405E
HCP9408E
HCP9410I
HCP9422E

CMS Changes

This section identifies the CMS externals that have changed since VM/ESA 1.2.1. It contains the following subsections:

- “CMS Commands”
- “CMS File Pool Administration and Operator Commands” on page 522
- “XEDIT Subcommands” on page 523
- “CMS Macros” on page 525
- “Preferred Routines (CSL Routines)” on page 527
- “Compatibility Routines” on page 537

- "CMS Pipelines" on page 538
- "CMS Messages" on page 538

CMS Commands

Table 89 lists the CMS commands that have changed since VM/ESA 1.2.1. See the *VM/ESA: CMS Command Reference* for complete descriptions of CMS commands.

Note: The three character module identifiers have been removed from the messages listed in the *VM/ESA: CMS Command Reference*. For example, a message that used to be listed as DMSAMS136S is now listed as DMS136S.

Table 89 (Page 1 of 9). CMS Commands Changed since VM/ESA 1.2.1

Command	Changes
ACCESS	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added message for use with BFS: DMS2133E.
ALIALIST	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
AUTHLIST	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
COPYFILE	**Incompatible changes:** • New message: DMS516E. **Upwardly compatible changes:** • Support for the new CMS SET RORESPECT command has been added. COPYFILE will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O.
CREATE ALIAS	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
CREATE DIRECTORY	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New message for authorization failure from ESM: DMS1331E.
CREATE FILE	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
CREATE LOCK	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added messages for use with BFS: DMS2040E, DMS2133E. • Added new operand for use with BFS: *bfsid*.
CREATE NAMEDEF	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 89 (Page 2 of 9). CMS Commands Changed since VM/ESA 1.2.1

Command	Changes
CSLLIST	**Upwardly compatible changes:** • 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.
CSLMAP	**Upwardly compatible changes:** • 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.
DATASPACE	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DEFAULTS	**Upwardly compatible changes:** • A new option is supported as a parameter for NETDATA SEND, SENDFILE, and NOTE: CLASS. • A new option is supported for NAMES: VMLINK. • Support for the new command VMLINK. • Supports new options: NOKEEPCC, KEEPCC. • New options are supported as parameters for FILELIST, NETDATA, and RDRLIST: VMDATE, SHORTDATE, FULLDATE, ISODATE.
DELETE LOCK	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*.
DIRATTR	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DIRLIST	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • 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.
DISCARD	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
EDIT	**Incompatible changes:** • The '(OLD' option is no longer supported. You must either use the XEDIT simulation by dropping the '(OLD' option or convert to using the normal XEDIT environment. • Message DMS987E was changed to DMS2520E. The old CMS editor is no longer supported.
ERASE	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*. • New message for authorization failure from ESM: DMS1332E.
FILEATTR	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 89 (Page 3 of 9). CMS Commands Changed since VM/ESA 1.2.1

Command	Changes
FILEDEF	**Upwardly compatible changes:** • New option: LIBSRV. • Allows LRECL definitions up to 65535 bytes for OS variable spanned records (under XLRI processing) and non-OS CMS files. • Allows BLKSIZE definitions up to 65535 bytes for non-OS CMS files.
FILELIST	**Incompatible changes:** • 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 LISTFILE profiles. See Appendix A of the *VM/ESA: CMS Command Reference*. • If you file the file created by FILELIST, that file might contain new and changed fields (on the far right). **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • The BEFORE *date* and AFTER *date* options support 4-digit years. • Screens and responses support 4-digit years. • If a date format option is not specified on the FILELIST command, the CMS DEFAULTS date format setting for FILELIST will be used. • Added message for use with BFS: DMS2133E. • 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.
FILEPOOL CONTROL BACKUP	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
FILEPOOL LIST BACKUP	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
FILEPOOL MINIDISK	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
FILESERV BACKUP	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
FILESERV DEFBACKUP	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
FINIS	**Incompatible changes:** • Will close files opened by the REXX STREAM I/O function.
GLOBAL	**Upwardly compatible changes:** • New message for duplicate library name in input list: DMS045W. The duplicates are ignored.
GRANT AUTHORITY	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 89 (Page 4 of 9). CMS Commands Changed since VM/ESA 1.2.1

Command	Changes
IDENTIFY	**Upwardly compatible changes:** • New options to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses support 4-digit years. • 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. • New option for TCP/IP mail integration: TCPIP.
LISTDIR	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
LISTDS	**Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Responses support 4-digit years. • 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.
LISTFILE	**Incompatible changes:** • Message DMS550E changed to DMS765E. **Upwardly compatible changes:** • New options for ALLDATES option: DTOC, DOLR, DTOLU, DTOLC. • New options to specify date format: SHORTDATE, FULLDATE, ISODATE. • The BEFORE *date* and AFTER *date* options support 4-digit years. • Responses support 4-digit years. • 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.
LKED	**Upwardly compatible changes:** • The defaults for *value1* and *value2* on the SIZE option have been increased to 400K and 100K, respectively.
LOAD	**Upwardly compatible changes:** • New message for insufficient storage above 16MB: DMS891W.
LOADMOD	**Incompatible changes:** • Changed message (new text possible): DMS639E.
MACLIST	**Upwardly compatible changes:** • 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.
MOVEFILE	**Upwardly compatible changes:** • Can process OS variable spanned records (under XLRI processing) and non-OS CMS files with record lengths up to 65535 bytes. • Adjusts output file sizes for compatibility between CMS and OS • Allows greater FILEDEF default flexibility for file attributes (RECFM, LRECL, BLKSIZE) • If default size values are used, fixes record truncation problems when moving data files from fixed to variable format • New message: DMS1116E.

Table 89 (Page 5 of 9). CMS Commands Changed since VM/ESA 1.2.1

Command	Changes
NAMEFIND	**Upwardly compatible changes:** • Supports the new form of the FILE opt, FILE (fn ft fm).
NAMES	**Upwardly compatible changes:** • Supports new option: VMLINK. • On the MAIL panel: – The Userid and Node fields will no longer be uppercased.
NETDATA	**Incompatible changes:** • Different message on empty reader condition: DMS205W (was DMS639E). • New message: DMS1184E • New return code: 20 **Upwardly compatible changes:** • NETDATA SEND supports a new option: CLASS. • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file. • Responses support 4-digit years. • 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.
NOTE	**Incompatible changes:** • In a note, a nonblank line between the 'OPTIONS:' line and the 'Date:' line, called 'USEROPTIONS:', is supported. Any other nonblank lines cause an error. In VM/ESA 1.2.1, you received an error message for any nonblank line between 'OPTIONS:' and 'Date:'. **Upwardly compatible change:** • The OPTIONS: line of a note contains a new option: CLASS. • In the Date field of the note header, the year is now displayed with four digits. • Changed message: – DMS2501E (text change for new function). • TCP/IP domain names accepted as user IDs or as the resolution of nicknames. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
NUCXLOAD	**Incompatible changes:** • Changed message (new text possible): DMS639E.
PARSECMD	**Upwardly compatible changes:** • BFS added new code.n value of PN.
PEEK	**Incompatible changes:** • PEEK profile has changed. If you used the PROFPEEK profile to create your own, PEEK may not work anymore. Use new PROFPEEK. **Upwardly compatible change:** • TCP/IP origin domain name address used when available and shown on PEEK message line for origin within current space and formatting limitations.
PIPE	**Incompatible changes:** • All CMS Pipelines messages have a new prefix, and many messages have new numbers and text. See “CMS Pipelines” on page 538.

Table 89 (Page 6 of 9). CMS Commands Changed since VM/ESA 1.2.1

Command	Changes
QUERY (in general)	**Incompatible changes:** • See changed commands below. **Upwardly compatible changes:** • Supports new operands.
QUERY ACCESSORS	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
QUERY ALIAS	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
QUERY AUTHORITY	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
QUERY BLOCKS	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*.
QUERY CMSLEVEL	**Upwardly compatible changes:** • In the response, the CMS level value has changed.
QUERY CMSREL	**Incompatible changes:** • The format of the response has changed to include the version. **Upwardly compatible changes:** • In the response, the release level value has changed.
QUERY DATASPACE	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
QUERY DIRATTR	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
QUERY ENROLL	**Upwardly compatible changes:** • Added new option for use with BFS: FILESPACE.
QUERY FILEATTR	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*.
QUERY FILEDEF	**Upwardly compatible changes:** • Supports a new optional operand, ATTRIBUT, and its response.
QUERY LIMITS	**Incompatible change:** • File pool administration authority is no longer needed to query limits on a filespace other than your own.

Table 89 (Page 7 of 9). CMS Commands Changed since VM/ESA 1.2.1

Command	Changes
QUERY LOCK	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Added new operand for use with BFS: *bfsid*.
QUERY NAMEDEF	**Upwardly compatible changes:** • Added new operand for use with BFS: *bfsid*.
QUERY RORESPECT	**Upwardly compatible changes:** • Displays the current RORESPECT setting. This setting controls whether or not updates to files using XEDIT and/or COPYFILE in SFS will respect the read-only access of a directory.

QUERY SYSNAMES	**Upwardly compatible changes:**
	• There is a change to the response. The CMSGUI saved system name is displayed.
	An example of the VM/ESA 2.4.0 response:
	`SYSNAMES:  CMSVSAM  CMSAMS   CMSDOS   CMSBAM   CMSGUI`
	`ENTRIES:   CMSVSAM  CMSAMS   CMSDOS   CMSBAM  VMGUILIB`

RDRLIST	**Incompatible changes:** • If you file the file created by RDRLIST, that file might contain new and changed fields (on the far right). **Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Screen supports 4-digit years. • If a date format option is not specified on the RDRLIST command, the CMS DEFAULTS date format setting for RDRLIST will be used. • 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. • TCP/IP origin domain name address used when available and shown on the RDRLIST panel origin area within current space and formatting limitations.
RECEIVE	**Upwardly compatible changes:** • Supports new options: NOKEEPCC, KEEPCC. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
RELEASE	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
RELOCATE	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
RENAME	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
REVOKE AUTHORITY	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
RTNLOAD	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 89 (Page 8 of 9). CMS Commands Changed since VM/ESA 1.2.1

Command	Changes
SENDFILE	**Incompatible changes:** • The SENDFILE screen has changed. A new line is added for the new CLASS option. Also, if your screen size is 24 lines, some input fields may be in different locations on the screen. • When sending a note, if incorrect information is found in the new *class* field of the OPTIONS: line, the note is not sent. In VM/ESA 1.2.1, information found after the NOTEBOOK option field on the OPTIONS: line was ignored. **Upwardly compatible changes:** • Supports a new option: CLASS. • Changed messages: DMS2501E, DMS081E. • TCP/IP domain names accepted as user IDs or as the resolution of nicknames. • New options to specify the transmission method: SMTP, MIME, UFTSYNC, UFTASYNC. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
SET (in general)	**Upwardly compatible changes:** • Supports new operands.
SET DOS	**Incompatible changes:** • Added message: DMS1101I. • Changed message: DMS333E.
SET DOSPART	**Incompatible changes:** • Added message: DMS1101I. • Changed message: DMS333E.
SET RORESPECT	**Upwardly compatible changes:** • Used to ensure updates to files using XEDIT and/or COPYFILE in SFS will respect the read-only access mode of a directory.
SET SYSNAME	**Upwardly compatible changes:** • Supports new operand: CMSGUI
SYNONYM	**Incompatible changes:** • Removed message: DMS032E. **Upwardly compatible changes:** • Supports a new parameter: *filetype*. In VM/ESA 1.2.1, file type had to be SYNONYM.
TAPE	**Upwardly compatible changes:** • 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. • Added message for Tape Library Dataserver support: DMS2147W.
TELL	**Upwardly compatible changes:** • Accepts a TCP/IP domain name as part of the destination information.
VMFPLC2	**Upwardly compatible changes:** • 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. • Added message for Tape Library Dataserver support: DMS2147W.

Table 89 (Page 9 of 9). CMS Commands Changed since VM/ESA 1.2.1

<table>
<tr><th>Command</th><th>Changes</th></tr>
<tr><td>XEDIT</td><td><b>Incompatible changes:</b>
• The file mode number of the autosave file may be different:<pre>
        Edited   Autosave
        File's   File's
Version File Mode File Mode
Release Number   Number

1.2.1   anything   0

2.4.0   0   0
         any other 1

</pre><b>Upwardly compatible changes:</b>
• Added new option for BFS: NAMETYPE If NAMETYPE BFS is used, your profile must be in REXX and it is loaded as a REXX function.
• Added new option for BFS: BFSLINE
• New messages for BFS: DMS033E, DMS512E, DMS2105E, and DMS2134E.</td></tr>
</table>

CMS File Pool Administration and Operator Commands

Table 90 lists the CMS file pool administration and operator commands that have changed since VM/ESA 1.2.1. See the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book for complete descriptions of file pool administration and operator commands.

Table 90 (Page 1 of 2). CMS File Pool Administration and Operator Commands Changed since VM/ESA 1.2.1

Command	Changes
AUDIT	**Upwardly compatible changes:** • New operands: *fn ft*, REPLACE. • Added messages: DMS024E, DMS1258E, DMS3253I, DMS3254E, DMS3255E. • Changed message: DMS3470W (new text possible).
DELETE USER	**Upwardly compatible changes:** • New options: DELAUTH | KEEPAUTH. • Added message: DMS2023E.
ENROLL USER	**Upwardly compatible changes:** • Supports new options: SFS | BFS, *userid*, *gname*. • Added messages for use with BFS: DMS1209E, DMS2023E, DMS2132E.
FILEPOOL BACKUP	**Upwardly compatible changes:** • Supports new BFS parameter: *bfsid*.
FILEPOOL RELOAD	**Upwardly compatible changes:** • New message: DMS3455I.
FILEPOOL RENAME	**Incompatible changes:** • The user ID renamed does not have to be enrolled in the file pool. Locks and authorizations are transferred to the new user ID. • New message: DMS2544W (this message replaces DMS1167E when both the administrator's CMS and the SFS server are at the new release level).

Table 90 (Page 2 of 2). CMS File Pool Administration and Operator Commands Changed since VM/ESA 1.2.1

Command	Changes
FILEPOOL UNLOAD	**Upwardly compatible changes:** • New message: DMS3455I.
FILESERV REORG	**Incompatible changes:** • A new response message, DMS3009R, is issued to recommend that you cancel the processing of the FILESERV REORG command if your control data backup is not current.
FILESERV START	**Incompatible changes:** • The start-up does not continue processing if the DMS3110E message is issued because of a SEGMENT ASSIGN failure.
QUERY FILEPOOL STATUS	**Incompatible changes:** • The following Counter Information has been added to the output: – Open File CreateMig Requests
QUERY LIMITS	**Incompatible changes:** • File pool administration authority is no longer required.

XEDIT Subcommands

Table 91 lists the XEDIT subcommands that have changed since VM/ESA 1.2.1. For complete descriptions of XEDIT subcommands, see the *VM/ESA: XEDIT Command and Macro Reference*.

Table 91 (Page 1 of 2). XEDIT Subcommands Changed since VM/ESA 1.2.1

Subcommand	Changes
A	**Incompatible changes:** • If you issue the A subcommand in non-display mode: – In VM/ESA 1.2.1, XEDIT returns message DMS529E with RC=3. – In VM/ESA 2.4.0, the subcommand completes successfully with no message and RC=0.
ADD	**Incompatible changes:** • If you issue the ADD subcommand in non-display mode: – In VM/ESA 1.2.1, XEDIT returns message DMS529E with RC=3. – In VM/ESA 2.4.0, the subcommand completes successfully with no message and RC=0.
CMS	**Upwardly compatible change:** • New message added: DMS642E
CP	**Upwardly compatible change:** • New message added: DMS642E
EXTRACT	**Upwardly compatible change:** • New operands added: BFSLine, EPName, NAMetype, PName, and GUI.
FILE	**Upwardly compatible changes:** • Support for the new CMS SET RORESPECT command has been added. FILE will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O. • Support for BFS files has been added. • New messages for BFS support: 024E, 054E, 1184E, 2105E, 2120E, 2131E, and 2134E.

Table 91 (Page 2 of 2). XEDIT Subcommands Changed since VM/ESA 1.2.1

Subcommand	Changes
GET	**Upwardly compatible changes:** • Support for BFS files has been added. • New messages for BFS support: 033E, 054E, 512E, 2105E, 2131E, and 2134E.
I	**Incompatible changes:** • If you issue the I subcommand in non-display mode: – In VM/ESA 1.2.1, XEDIT returns message DMS529E with RC=3. – In VM/ESA 2.4.0, the subcommand completes successfully with no message and RC=0.
LOAD	**Upwardly compatible changes:** • Support for writing to BFS files has been added. • Added new options: NAMetype and BFSLine. • New messages for BFS support: 033E, 512E, 2105E, and 2134E.
MODIFY	**Upwardly compatible changes:** • New operands added: BFSLine, NAMetype, and PName.
PRESERVE	**Upwardly compatible changes:** • New operands added: BFSLine, NAMetype, and PName.
PUT	**Upwardly compatible changes:** • Support for the new CMS SET RORESPECT command has been added. PUT will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O. • Support for writing to BFS files has been added. • New return code for BFS support: 32. • New messages for BFS support: 033E, 054E, 512E, 1184E, 2105E, 2120E, 2131E, and 2134E.
PUTD	**Upwardly compatible changes:** • Support for the new CMS SET RORESPECT command has been added. PUTD will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O. • Support for writing to BFS files has been added. • New return code for BFS support: 32. • New messages for BFS support: 033E, 054E, 512E, 1184E, 2105E, 2120E, 2131E, and 2134E.
QUERY	**Upwardly compatible change:** • New operand added.
SAVE	**Upwardly compatible changes:** • Support for the new CMS SET RORESPECT command has been added. SAVE will fail if SET RORESPECT ON has been issued and the target filemode is an SFS file control directory accessed R/O. • Support for writing to BFS files has been added. • New return code for BFS support: 32. • New messages for BFS support: 024E, 054E, 512E, 1184E, 2105E, 2120E, 2131E, and 2134E.
SET	**Upwardly compatible change:** • New operand added.
XEDIT	**Incompatible changes:** • See the entry for XEDIT in Table 89 on page 514 for details on the incompatibilities. • In update mode, the XEDIT subcommand no longer allows the same file in the ring multiple times. When a *fileid* match in the ring is found, that file becomes the current file.

CMS Macros

Table 92 lists the CMS preferred macros that have changed since VM/ESA 1.2.1. Table 93 lists the CMS compatibility macros that have changed. Table 94 lists the OS simulation macros that have changed. Table 95 lists the simulated OS/MVS supervisor calls that have changed.

Preferred Macros

See the *VM/ESA: CMS Application Development Reference for Assembler* for complete descriptions of these macros.

Table 92 (Page 1 of 2). CMS Preferred Macros Changed since VM/ESA 1.2.0

Macro	Changes
AMODESW	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370
CMSCVT	**Incompatible changes:** • Field CVTTZ in CVTSECT is updated to reflect current timezone offset when x'2004' interrupt is processed by CMS. **Upwardly compatible changes:** • Added the fields: CVTECVT, CVTFLAG2.
CMSLEVEL	**Incompatible changes:** • The CMS level has been frozen at X'0F' for CMS Level 12 (VM/ESA Version 2 Release 1.0) and later. Use the new DMSQEFL macro or the DMSQEFL CSL routine instead.
CSLENTRY	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370
CSLFPI	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370
DIRBUFF	**Upwardly compatible changes:** • The FILE record contains the following new fields: DIRFDAXD, DIRFDAXC, DIRFDAXI, DIRFLV13. • The FILEEXT record contains the following new fields: DIREDAXD, DIREDAXC, DIREDAXI, DIREDRXD, DIREDRXC, DIREDRXI, DIRECDXD, DIRECDXC, DIRECDXI, DIREDCXD, DIREDCXC, DIREDCXI, DIRELV13. • The SEARCHALL and SEARCHAUTH records contain the following new fields: DIRSDAXD, DIRSDAXC, DIRSDAXI, DIRSCEND, DIRSLV13.
ENABLE	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370
EXSBUFF	**Upwardly compatible changes:** • The FILE record contains the following new fields: EXSFDAXD, EXSFDAXC, EXSFDAXI, EXSFDRXD, EXSFDRXC, EXSFDRXI, EXSFCDXD, EXSFCDXC, EXSFCDXI, EXSFDCXD, EXSFDCXC, EXSFDCXI, EXSF2000, EXSFLV13. • In the FILE record, the following field has changed: EXSFRES. • The DIR record contains the following new fields: EXSDDCXD, EXSDDCXC, EXSDDCXI, EXSDCDXD, EXSDCDXC, EXSDCDXI, EXSD2000, EXSDLV13. • In the DIR record, the following field has changed: EXSDRES.
FSSTATE	**Upwardly compatible changes:** • 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*nn*, 1=20*nn*, where *nn* is the 2-digit year). In VM/ESA 1.2.1, this bit was not used.

Table 92 (Page 2 of 2). CMS Preferred Macros Changed since VM/ESA 1.2.0

Macro	Changes
FSTD	**Upwardly compatible changes:** • The FSTFLAGS section contains the new FSTCNTRY field, which is a bit that indicates the century (first two digits of the year) the file was last written or updated (0=19*nn*, 1=20*nn*, where *nn* is the 2-digit year).
FSWRITE	**Incompatible changes:** • If the caller's buffer overlaps internal CMS file system buffers, an error code 2 will be returned, or message 1307T will be issued and the system will be abnormally terminated, if it is necessary to avoid file corruption.
HNDIO	**Upwardly compatible changes:** • Supports new parameter: PERSIST
SEGMENT	**Incompatible changes:** • New return code for SEGMENT FIND: 41.
TAPECTL	**Upwardly compatible changes:** • 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.

Compatibility Macros

See *VM/ESA: Planning and Administration* for information about the DEFNUC macro.

Table 93. CMS Compatibility Macros Changed since VM/ESA 1.2.1

Macro	Changes
DEFNUC	**Incompatible changes:** • The format of the default CMS IPL heading (used when the VERSION= parameter is specified without a value in DEFNUC) has changed to include the version. Also, the date in the default IPL heading is presented in ISO format (*yyyy-mm-dd*). **Upwardly compatible changes:** • The USEMTSEG and MTSEG parameters no longer have any effect because VMMTLIB is no longer created as a saved segment; it now resides within the CMS nucleus.

OS Simulation Macros and Supervisor Calls

See the *VM/ESA: CMS Application Development Guide for Assembler* for information about the OS macros and supervisor calls that CMS simulates.

Table 94 (Page 1 of 2). OS Simulation Macros Changed since VM/ESA 1.2.1

Macro	Changes
CLOSE	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.

Table 94 (Page 2 of 2). OS Simulation Macros Changed since VM/ESA 1.2.1

Macro	Changes
DCB	**Upwardly compatible changes:** • 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.
GET	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.
OPEN	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
PUT	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.
TIME	**Upwardly compatible changes:** • The 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.

Table 95. OS Simulation Supervisor Calls Changed since VM/ESA 1.2.1

SVC	Changes
SVC 19 (OPEN)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
SVC 20 (CLOSE)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.

Preferred Routines (CSL Routines)

Table 96 lists the CSL routines that have changed since VM/ESA 1.2.1. See the *VM/ESA: CMS Application Development Reference* for complete descriptions of all the routines in this table except QueueOpen. The description of QueueOpen is in *VM/ESA: CMS Application Multitasking*.

Table 96 (Page 1 of 11). CSL Routines Changed since VM/ESA 1.2.1

Routine	Changes
DMSCATTR	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 96 (Page 2 of 11). CSL Routines Changed since VM/ESA 1.2.1

Routine	Changes
DMSCLBLK	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • New reason codes when using BFS: 10220, 65400. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330, 90492, 90495.
DMSCLDBK	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • New reason code when using BFS: 65400. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330, 90495.
DMSCLDIR	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation.
DMSCLOSE	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330, 90495.
DMSCRALI	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • Supports a new option: UNRESOLVED. • New reason codes: 61620, 98700. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSCRDIR	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.

Table 96 (Page 3 of 11). CSL Routines Changed since VM/ESA 1.2.1

Routine	Changes
DMSCRFIL	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* and *create_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSCRLOC	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New reason codes when using BFS: 65400, 69200, and 69300. • Supports new BFS parameter: *bfsid*.
DMSCROB	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* and *create_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSDELOC	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid*.
DMSDEUSR	**Upwardly compatible changes:** • New parameters: DELAUTH, KEEPAUTH, and *length4*. • New reason code: 98700.
DMSDIRAT	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSENUSR	**Upwardly compatible changes:** • New reason codes when using BFS: 10210, 10240, 69000, 90300, 90320, and 98700. • Supports new options: SFS | BFS, *userid*, *gname*. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE, *length7*. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90330, 90495.

Table 96 (Page 4 of 11). CSL Routines Changed since VM/ESA 1.2.1

Routine	Changes
DMSERASE	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid*.
DMSERP	**Upwardly compatible changes:** • Supports new functions: GETENV, RES:*envir*, REP:*envir*, EXT:*envir*. • New return code: 125. • Changed return codes: 112, 117, 124. • New information names for Year 2000 support (FILE_DATE_CENTURY, ACT_FILE_DATE_CENTRY (note that U is omitted), and YEAR2000_SUPPORT).
DMSEXIDI	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.1, it was returned as 8 characters. **Upwardly compatible changes:** • New parameters: *update_date*, *update_time*, *create_date*, *create_time*. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameters: *bfsid*, *filesystemtype* • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *last_change_date* and *create_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330.
DMSEXIFI	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.1, it was returned as 8 characters. **Upwardly compatible changes:** • New parameters: *update_date*, *update_time*. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameters: *bfsid*, *filesystemtype* • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date*, *dateref*, *create_date*, and *last_change_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330.

Table 96 (Page 5 of 11). CSL Routines Changed since VM/ESA 1.2.1

Routine	Changes
DMSEXIST	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid* • The following offsets have changed in the FILE data record (and the record length has increased to 436 bytes):

OFFSET	Field Name	Description of Change
1 (X'1')	*file_system_type*	Reserved in VM/ESA 1.2.1
315 (X'13B')	*dec_update_date*	Reserved in VM/ESA 1.2.1
318 (X'13E')	Reserved	Added
319 (X'13F')	*dec_update_time*	Added
322 (X'142')	Reserved	Added
323 (X'143')	*update_date*	Added
331 (X'14B')	*update_time*	Added
339 (X'153')	*dec_date_ext*	Added
343 (X'157')	*date_ext*	Added
353 (X'161')	*iso_date_ext*	Added
363 (X'16B')	*dec_dateref_ext*	Added
367 (X'16F')	*dateref_ext*	Added
377 (X'179')	*iso_dateref_ext*	Added
387 (X'183')	*dec_cr_date_ext*	Added
391 (X'187')	*cr_date_ext*	Added
401 (X'191')	*iso_cr_date_ext*	Added
411 (X'19B')	*dec_last_change_date_ext*	Added
415 (X'19F')	*last_change_date_ext*	Added
425 (X'1A9')	*iso_last_change_date_ext*	Added
435 (X'1B3')	Reserved	Added

(continued on next page)

Table 96 (Page 6 of 11). CSL Routines Changed since VM/ESA 1.2.1

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSEXIST - continued</td><td><b>Upwardly compatible changes (continued):</b>
• The following offsets have changed in the DIRECTORY data record (and the record length has increased to 308 bytes):
<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>1 (X'1')</td><td><i>file_system_type</i></td><td>Reserved in VM/ESA 1.2.1</td></tr>
<tr><td>206 (X'CE')</td><td><i>dec_update_date</i></td><td>Reserved in VM/ESA 1.2.1</td></tr>
<tr><td>209 (X'D1')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>210 (X'D2')</td><td><i>dec_update_time</i></td><td>Added</td></tr>
<tr><td>213 (X'D5')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>214 (X'D6')</td><td><i>update_date</i></td><td>Added</td></tr>
<tr><td>222 (X'DE')</td><td><i>update_time</i></td><td>Added</td></tr>
<tr><td>230 (X'E6')</td><td><i>dec_cr_date</i></td><td>Added</td></tr>
<tr><td>233 (X'E9')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>234 (X'EA')</td><td><i>dec_cr_time</i></td><td>Added</td></tr>
<tr><td>237 (X'ED')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>238 (X'EE')</td><td><i>cr_date</i></td><td>Added</td></tr>
<tr><td>246 (X'F6')</td><td><i>cr_time</i></td><td>Added</td></tr>
<tr><td>254 (X'FE')</td><td><i>dec_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>258 (X'102')</td><td><i>last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>268 (X'10C')</td><td><i>iso_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>278 (X'116')</td><td><i>dec_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>282 (X'11A')</td><td><i>cr_date_ext</i></td><td>Added</td></tr>
<tr><td>292 (X'124')</td><td><i>iso_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>302 (X'12E')</td><td>Reserved</td><td>Added</td></tr>
</table></td></tr>
<tr><td>DMSFILEC</td><td><b>Incompatible changes:</b>
• You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation.
<b>Upwardly compatible changes:</b>
• The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.</td></tr>
<tr><td>DMSGETDA</td><td><b>Incompatible changes:</b>
• When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.1, it was returned as 8 characters.
<b>Upwardly compatible changes:</b>
• New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and <i>length2</i>.
• The <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.
• New reason codes: 90310, 90320, 90330.</td></tr>
<tr><td>DMSGETDF</td><td><b>Incompatible changes:</b>
• When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.1, it was returned as 8 characters.
<b>Upwardly compatible changes:</b>
• New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and <i>length2</i>.
• The <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.
• New reason codes: 90310, 90320, 90330.</td></tr>
</table>

Table 96 (Page 7 of 11). CSL Routines Changed since VM/ESA 1.2.1

Routine	Changes
DMSGETDI	**Upwardly compatible changes:**

- The following offsets have changed in the FILE data record (and the record length has increased to 112 bytes):

OFFSET	Field Name	Description of Change
88 (X'58')	*dec_date_ext*	Added
92 (X'5C')	*date_ext*	Added
102 (X'66')	*iso_date_ext*	Added

- The following offsets have changed in the FILEEXT data record (and the record length has increased to 284 bytes):

OFFSET	Field Name	Description of Change
1 (X'1')	*file_system_type*	Reserved in VM/ESA 1.2.1
161 (X'A1')	*dec_update_date*	Reserved in VM/ESA 1.2.1
164 (X'A4')	Reserved	Added
165 (X'A5')	*dec_update_time*	Added
168 (X'A8')	Reserved	Added
169 (X'A9')	*update_date*	Added
177 (X'B1')	*update_time*	Added
185 (X'B9')	*dec_date_ext*	Added
189 (X'BD')	*date_ext*	Added
199 (X'C7')	*iso_date_ext*	Added
209 (X'D1')	*dec_dateref_ext*	Added
213 (X'D5')	*dateref_ext*	Added
223 (X'DF')	*iso_dateref_ext*	Added
233 (X'E9')	*dec_cr_date_ext*	Added
237 (X'ED')	*cr_date_ext*	Added
247 (X'F7')	*iso_cr_date_ext*	Added
257 (X'101')	*dec_last_change_date_ext*	Added
261 (X'105')	*last_change_date_ext*	Added
271 (X'10F')	*iso_last_change_date_ext*	Added
281 (X'119')	Reserved	Added

- The following offsets have changed in the LOCK data record:

OFFSET	Field Name	Description of Change
1 (X'1')	*file_system_type*	Reserved in VM/ESA 1.2.1

- The following offsets have changed in the SEARCHALL and SEARCHAUTH data records (and the record length has increased to 252 bytes):

OFFSET	Field Name	Description of Change
226 (X'E2')	Reserved	Added
228 (X'E4')	*dec_date_ext*	Added
232 (X'E8')	*date_ext*	Added
242 (X'F2')	*iso_date_ext*	Added

Routine	Changes
DMSGETDK	**Upwardly compatible changes:**

- Supports new BFS parameter: *filesystemtype*

Table 96 (Page 8 of 11). CSL Routines Changed since VM/ESA 1.2.1

Routine	Changes
DMSGETDS	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.1, it was returned as 8 characters. **Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330.
DMSGETDX	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.1, it was returned as 8 characters. **Upwardly compatible changes** • Supports new parameters: *update_date*, *update_time*. • Supports new BFS parameter: *filesystemtype* • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*. • The *date*, *dateref*, *create_date*, and *update_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330.
DMSGRANT	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSOPBLK	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.1, it was returned as 8 characters. **Upwardly compatible changes:** • Supports a new option: RESOLVE. • Supports new parameters: CREATEMIG, *unique_id*, *userid*, *dateref*, *bfsid* • New reason codes: 30000, 63800, 98700. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New reason codes when using BFS: 10220, 65400, 98700. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date*, *create_date*, and *dateref* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSOPCAT	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid* • Supports new BFS options: FILEATTR SFS|BFS, READEXT • New reason codes when using BFS: 90300, 90350.

Table 96 (Page 9 of 11). CSL Routines Changed since VM/ESA 1.2.1

Routine	Changes
DMSOPDBK	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid* • New reason code when using BFS: 10220. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *create_date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSOPDIR	**Upwardly compatible changes:** • Supports new parameters: *buffersize*, *number*. • New reason codes: 30000, 63800, 98700. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid* • New reason code when using BFS: 65400.
DMSOPEN	**Upwardly compatible changes** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid* • New reason code when using BFS: 10220. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *create_date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSQEFL	**Upwardly compatible changes:** • Supports new values for *cp_level* and *cms_level*
DMSQLIMA	**Incompatible changes:** • File pool administration authority no longer required. **Upwardly compatible changes:** • Supports new BFS parameter: *filesystemtype*
DMSQLIMD	**Upwardly compatible changes:** • Supports new BFS parameter: *filesystemtype*
DMSQLIMU	**Incompatible changes:** • File pool administration authority no longer required. **Upwardly compatible changes:** • Supports new BFS parameter: *filesystemtype*
DMSQSFSL	**Upwardly compatible changes:** • Supports new value for *server_level*
DMSQOBJ	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.

Table 96 (Page 10 of 11). CSL Routines Changed since VM/ESA 1.2.1

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSRDCAT</td><td><b>Upwardly compatible changes:</b>
• New ACAT record for unresolved aliases.
• New catalogs for BFS support: NAMECAT, NOVCAT.
• New OBJECTCAT record for BFS object.
• SPACECAT record — the four reserved CHAR(8) fields at the end of the record (following the STORAGEGROUP field) have been restructured for BFS support as follows:
<table>
<tr><th>Field Name</th><th>Field Type/Description</th></tr>
<tr><td>HIGHINO</td><td>INTEGER(4) High OBJECTCAT INO value</td></tr>
<tr><td>HIGHNID</td><td>INTEGER(4) High NAMECAT NID value</td></tr>
<tr><td>FLAGS</td><td>CHAR(1) File space attributes</td></tr>
<tr><td>RESERVED01</td><td>CHAR(7) Reserved</td></tr>
<tr><td>RESERVED02</td><td>CHAR(8) Reserved</td></tr>
<tr><td>RESERVED03</td><td>CHAR(8) Reserved</td></tr>
</table>
• DIRCAT record — new bit settings in the DIRATTS field for BFS support.
• OBJECTCAT record for SFS:
– New bit settings in TYPE and FILEFLAGS fields for BFS support.
– New bit settings in FILEFLAGS field to support 4-digit years (century setting for DATE and DATEREF fields).
– The fields following the BSCID field are:
<table>
<tr><th>Field Name</th><th>Field Type/Description</th></tr>
<tr><td></td><td>CHAR(8) reserved</td></tr>
<tr><td>CHGDATE_CENTURY</td><td>CHAR(1) Century byte for LAST_CHANGE_DATE</td></tr>
<tr><td>LAST_CHANGE_DATE</td><td>CHAR(3) UTC date of last change</td></tr>
<tr><td></td><td>CHAR(1) reserved</td></tr>
<tr><td>LAST_CHANGE_TIME</td><td>CHAR(3) UTC time of last change</td></tr>
<tr><td>DRA1</td><td>CHAR(8) DRA1</td></tr>
<tr><td>DRA2</td><td>CHAR(8) DRA2</td></tr>
<tr><td>DRA3</td><td>CHAR(8) DRA3</td></tr>
<tr><td></td><td>INTEGER(4) reserved</td></tr>
<tr><td>CREATIONDATE_CENTURY</td><td>CHAR(1) Century byte for CREATIONDATE</td></tr>
<tr><td>CREATIONDATE</td><td>CHAR(3) UTC date of file creation</td></tr>
<tr><td></td><td>CHAR(1) reserved</td></tr>
<tr><td>CREATIONTIME</td><td>CHAR(3) UTC time of file creation</td></tr>
<tr><td>DATEREF</td><td>CHAR(3) UTC date of last reference</td></tr>
</table>
• FQFN record — new H type record for open intent READEXT.
• AUTHCAT record — new G type record for open intent READEXT.
• EOCAT record — new bit settings in TYPE field for BFS support.</td></tr>
<tr><td>DMSRELOC</td><td><b>Upwardly compatible changes:</b>
• The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.</td></tr>
</table>

Table 96 (Page 11 of 11). CSL Routines Changed since VM/ESA 1.2.1

Routine	Changes
DMSRENAM	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSREVOK	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSTRUNC	**Incompatible changes:** • You receive a new reason code, 50700, when there is not enough room in the file space to complete the operation. **Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSUDATA	**Upwardly compatible changes:** • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine.
DMSVALDT	**Upwardly compatible changes:** • Supports new parameters: *namedef1*, *namedef2*. • New reason codes: 90510, 90530, 96100. • The default changed for the file space name, if not specified. It defaults first to the file space ID set with the SET FILESPACE command, and then to the user ID calling the routine. • Supports new BFS parameter: *bfsid*.
QueueOpen	**Upwardly compatible changes:** • In VM/ESA 1.2.1, if you tried to use QueueOpen to open a remotely-located network-level queue but forgot to set up the CMS Communication Directory (that is, installed no ComDir entry describing the location of the remote queue), then QueueOpen would fail with reason code *vm_ipc_comm_lost*. In VM/ESA 2.4.0, QueueOpen will proceed anyway, using default values for what it tried to extract from the ComDir. The default values used are described in *VM/ESA: CMS Application Multitasking*.

Compatibility Routines

Table 97 lists the CMS compatibility routines that have changed between VM/ESA 1.2.1 and VM/ESA 2.4.0. See the *VM/ESA: CMS Application Development Guide for Assembler* for more information about the routines in this table.

Table 97. CMS Compatibility Routines Changed since VM/ESA 1.2.1

CMS Routine	Explanation
DMSTVS	**Upwardly compatible changes:** • Supports new LIBSRV plist parameter. • 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.

CMS Pipelines

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.
- 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.

  Note: User-written applications that are sensitive to these changes may require alterations. Published externals (such as PIPGFTXT) have **not** been changed.
- 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. Some specific enhancements that have occurred since VM/ESA 2.1.0 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 *VM/ESA: CMS Pipelines Reference* can be found in the *CMS/TSO Pipelines: Author's Edition*, which is included with the VM/ESA 2.4.0 library.

CMS Messages

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 H, “CMS Pipelines Message Cross-Reference [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]” on page 669.

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/ESA.

The following CMS message does not exist in VM/ESA 2.4.0:

DMS683E

The following CMS messages have changed since VM/ESA 1.2.1. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CMS messages.

DMS002E
DMS003E
DMS005E
DMS007E
DMS017E
DMS023E
DMS024E
DMS026E
DMS029E
DMS033E
DMS036E
DMS037E
DMS042E
DMS054E
DMS056E
DMS062E
DMS069I
DMS081E
DMS106S
DMS132S
DMS149E
DMS250E
DMS333E
DMS344E
DMS358E
DMS360E
DMS361E
DMS363R
DMS364I
DMS365I
DMS366R
DMS516E
DMS531E
DMS595E
DMS616W
DMS618E
DMS622E
DMS639E
DMS647E
DMS651E
DMS660W
DMS1007E
DMS1127I
DMS1131E
DMS1137E
DMS1138E
DMS1139E
DMS1146E
DMS1151E
DMS1153E
DMS1162E
DMS1176E
DMS1184E
DMS1205I
DMS1227E
DMS1229E
DMS1239E
DMS1258E
DMS1259E
DMS1306T
DMS1307T
DMS1311E
DMS1339S
DMS1341S
DMS1342S
DMS1343S
DMS1344S
DMS2008E
DMS2010E
DMS2013E
DMS2023E
DMS2031E
DMS2040E
DMS2041E
DMS2153E
DMS2154E
DMS2155E
DMS2501E
DMS2511W
DMS2523E
DMS2727E
DMS2731I
DMS2805E
DMS2982E
DMS3000W
DMS3088R
DMS3110E
DMS3208E
DMS3284E
DMS3431E
DMS3438E
DMS3438I
DMS3453E
DMS3454E
DMS3455I
DMS3470W
DMS3494R
DMS3508E
DMS3514E
DMS3515E
DMS3518E
DMS3521E
DMS3522E
DMS3526E
DMS3527E
DMS3528E
DMS3529E
DMS3530E
DMS3531E
DMS3532E
DMS3557E
DMS3561E
DMS3562E
DMS3585E
DMS3594R
DMS3616W
DMS3617I
DMS3618W
DMS3620I
DMS3622E
DMS3623E
DMS3624W
DMS3627E
DMS3628E
DMS3629E
DMS3630W
DMS3631W
DMS3632W
DMS3636E
DMS3641W
DMS3642W
DMS3727E
DMS3728E
DMS3926E

REXX/VM Changes

This section identifies the REXX/VM externals that have changed since VM/ESA 1.2.1. This section contains the following subsections:

- “REXX/VM Functions”
- “External Functions” on page 540

REXX/VM Functions

Table 98 lists the REXX/VM functions that have changed since VM/ESA 1.2.1. For complete descriptions of these functions, see the *VM/ESA: REXX/VM Reference*.

Table 98. REXX/VM Functions Changed since VM/ESA 1.2.1

Function	Changes
DATE	**Upwardly compatible changes:** • New parameters allow you to specify a date to be converted to a different format. • New parameters: *output_separator_char*, *input_separator_char*.
STREAM	**Upwardly compatible changes:** • Supports a new option: RECFM.

External Functions

Table 99 lists external functions which can be used by REXX/VM that have changed since VM/ESA 1.2.1. For complete descriptions of these functions, see the *VM/ESA: REXX/VM Reference*.

Table 99. External Functions Changed since VM/ESA 1.2.1

Function	Changes
CMSFLAG	**Upwardly compatible changes:** • New value for *flag*: YEAR2000.
DIAG DIAGRC	**Upwardly compatible changes:** • New DIAGNOSE code is supported: X'270'.

GCS Changes

This section identifies the GCS externals that have changed since VM/ESA 1.2.1. It contains the following subsection:

- "GCS Commands and Macros"

GCS Commands and Macros

Table 100 lists the GCS commands and macros that have changed since VM/ESA 1.2.1. See *VM/ESA: Group Control System* for complete descriptions of all GCS commands and macros.

Table 100 (Page 1 of 2). GCS Commands and Macros Changed since VM/ESA 1.2.1

Command/Macro	Changes
CVT macro	**Upwardly compatible changes:** • Added the fields: CVTECVT, CVTFLAG2.
GCSLEVEL macro	**Upwardly compatible changes:** • Additional equates for release levels.

Table 100 (Page 2 of 2). GCS Commands and Macros Changed since VM/ESA 1.2.1

Command/Macro	Changes
GETMAIN macro	**Incompatible changes:** • In VM/ESA 1.2.1, although LOC=RES was documented as the default, the actual default was LOC=BELOW, and all the requested virtual storage was allocated below 16MB. In VM/ESA 2.2.0, the actual default was 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.
ITRACE command	**Upwardly compatible changes:** • New operand: SP.
QUERY ADDRESS command	**Upwardly compatible changes:** • Operand *variable* changed. It now supports either *name* or *address*.
QUERY GCSLEVEL command	**Incompatible changes:** • The format of the response has changed to include the version and the service level: `VM/ESA Version 2 Release 4.0, Service Level 0`
QUERY MODDATE command	**Incompatible changes:** • A full 4-digit year is now returned in the date field of the response instead of a 2-digit year.
TIME macro	**Upwardly compatible changes:** • The 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.

Dump Viewing Facility Changes

This section identifies the Dump Viewing Facility externals that have changed since VM/ESA 1.2.1. It contains the following subsection:

- “Dump Viewing Facility Commands”

Dump Viewing Facility Commands

Table 101 lists the Dump Viewing Facility commands that have changed between VM/ESA 1.2.1 and VM/ESA 2.4.0. See *VM/ESA: Dump Viewing Facility* for complete descriptions of Dump Viewing Facility commands.

Note: The Dump Viewing Facility in VM/ESA 2.4.0 can process only dumps produced by VM/ESA 2.4.0.

Table 101. Dump Viewing Facility Commands Changed since VM/ESA 1.2.1

Dump Viewing Facility Command	Explanation
DUMPSCAN	**Upwardly compatible changes:** • ASID and ACCLIST subcommands support CP abend dumps, snapdumps, and stand-alone dumps.

VMSES/E Changes

This section identifies the VMSES/E externals that have changed since VM/ESA 1.2.1. It contains the following subsection:

- "VMSES/E Messages"

VMSES/E Messages

The following VM/ESA 1.2.1 message does not exist in VM/ESA 2.4.0:

VMF2709E

The following VMSES/E messages have changed since VM/ESA 1.2.1. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of VMSES/E messages.

VMF002E
VMF066E
VMF071E
VMF395E
VMF1015E
VMF1016E
VMF1059E
VMF1068E
VMF1444I
VMF1507E
VMF1821E
VMF1832W
VMF1851E
VMF1851I
VMF1861E
VMF1861I
VMF1861W
VMF1868W
VMF1905E
VMF1908E
VMF1944E
VMF1953W
VMF2066E
VMF2070E
VMF2111E
VMF2112I
VMF2114R
VMF2118I
VMF2119I
VMF2120W
VMF2152E
VMF2162E
VMF2173I
VMF2181E
VMF2200E
VMF2201E
VMF2202E
VMF2206E
VMF2206W
VMF2215E
VMF2225E
VMF2228E
VMF2242E
VMF2500E
VMF2507I
VMF2509I
VMF2602R
VMF2704E
VMF2740E
VMF2760I
VMF2780E
VMF2786E
VMF2830E
VMF2831E
VMF2835E
VMF2866I
VMF2867W
VMF2930E

Programmable Operator Facility Changes

This section identifies the Programmable Operator Facility externals that have changed since VM/ESA 1.2.1. It contains the following subsection:

- "Programmable Operator Facility Commands and Routing Table Statements"

Also see "Summary of Service Exec Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]" on page 250.

Programmable Operator Facility Commands and Routing Table Statements

Table 102 lists the Programmable Operator Facility commands and routing table statements that have changed between VM/ESA 1.2.1 and VM/ESA 2.4.0. See *VM/ESA: Planning and Administration* for complete descriptions of programmable operator facility commands and routing table statements.

Table 102. Programmable Operator Facility Commands and Routing Table Statements Changed since VM/ESA 1.2.1

Programmable Operator Command or Statement	Explanation
MSGLIMIT	**Upwardly compatible changes:** • New routing table configuration entry statement.

Chapter 20. Compatibility Tables for Converting from VM/ESA 1.2.2

This chapter identifies the VM/ESA externals that have changed between VM/ESA 1.2.2 and VM/ESA 2.4.0. It contains the following major sections:

- "CP Changes"
- "CMS Changes" on page 553
- "REXX/VM Changes" on page 571
- "GCS Changes" on page 571
- "VMSES/E Changes" on page 572
- "Programmable Operator Facility Changes" on page 573

Use the information provided in this chapter to determine if you need to make any changes in the way you use these functions. See "Terminology and Symbols" on page 3 for the meaning of compatibility terms used in the tables.

CP Changes

This section identifies the CP externals that have changed since VM/ESA 1.2.2. It contains the following subsections:

- "CP Commands"
- "SYSTEM CONFIG Statements" on page 549
- "User Directory Control Statements" on page 550
- "CP Utilities" on page 551
- "CP DIAGNOSE Codes" on page 551
- "CP Messages" on page 552

CP Commands

Table 103 lists the CP commands that have changed since VM/ESA 1.2.2. Refer to the *VM/ESA: CP Command and Utility Reference* for complete descriptions of CP commands.

Table 103 (Page 1 of 5). CP Commands Changed since VM/ESA 1.2.2

Command	Changes
CPLISTFILE	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses include 4-digit years for FULLDATE and ISODATE operands.
DEFINE (in general)	**Upwardly compatible changes:** • Supports new operand: MSGPROC. • Additional messages: HCP045E, HCP260E, HCP1014E, HCP2800E, HCP2801E, HCP2802E, HCP2803E, HCP2804I, HCP2806E, HCP2811E. • See DEFINE commands below.
DEFINE CRYPTO	**Upwardly compatible changes:** • New response. • New message: HCP1716E.
DETACH (in general)	**Upwardly compatible changes:** • Supports new operand: MSGPROC. • Additional messages: HCP260E, HCP2805E, HCP2807E.

Table 103 (Page 2 of 5). CP Commands Changed since VM/ESA 1.2.2

Command	Changes
DISPLAY Linkage Stack	**Upwardly compatible changes:** • Response indicates the called-space identification (CSID) if the linkage-stack entry type is a program-call state entry with a called-space ID.
DISPLAY Registers	**Upwardly compatible changes:** • Supports new operand: FPC. • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New response when FPC operand is used. • New messages: HCP6153E, HCP6154E.
DUMP Linkage Stack	**Upwardly compatible changes:** • Response indicates the called-space identification (CSID) if the linkage-stack entry type is a program-call state entry with a called-space ID.
IPL	**Incompatible changes:** • A new response is issued if tracing is active when a preferred guest is IPLed. Tracing must be turned off and the guest reIPLed. • In VM/ESA 1.2.2, when you IPL CMS with the PARM operand, CMS initialization attaches a fence of 8 bytes of X'FF's to the end of the PARM data before passing it to the SYSPROF EXEC. In VM/ESA 2.4.0, 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.
LOGON or LOGIN	**Upwardly compatible changes:** • New message: HCP2808E.
QUERY (in general)	**Incompatible changes:** • See QUERY commands below. **Upwardly compatible changes:** • See QUERY commands below.
QUERY CACHEFW	**Upwardly compatible changes:** • Response indicates if the cache fast write function is suspended for the subsystem.
QUERY CHPID	**Upwardly compatible changes:** • New operand: TYPE. • New responses if TYPE is specified.
QUERY CPLEVEL	**Incompatible changes:** • The format of the response has changed to include the version. **Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • In the response, the release level value has changed. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q CPLEVEL command, the output from Q 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.
QUERY DASDFW	**Upwardly compatible changes:** • Response indicates if the DASD fast write function is suspended for the subsystem.

Table 103 (Page 3 of 5). CP Commands Changed since VM/ESA 1.2.2

Command	Changes
QUERY IMG	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q IMG command, the output from Q 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.
QUERY LDEVS	**Upwardly compatible changes:** • The response may include the IP address for TCP/IP.
QUERY NAMES	**Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY NLS	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q NLS command, the output from Q 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.
QUERY NSS	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q NSS command, the output from Q 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.
QUERY READER / PRINTER / PUNCH	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses include 4-digit years for FULLDATE and ISODATE operands. • 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. • 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 Q RDR|PRT|PUN command, the output from Q 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).
QUERY TIME	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q TIME command, the output from Q TIME uses the new default date format. This causes the date to be expanded to include the 4-digit year.

Table 103 (Page 4 of 5). CP Commands Changed since VM/ESA 1.2.2

Command	Changes
QUERY TRFILES	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q TRFILES command, the output from Q 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.
QUERY UCR	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q UCR command, the output from Q 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.
QUERY USERID	**Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY USERS	**Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL ALL	**Upwardly compatible changes:** • Supports virtual message processors.
QUERY VIRTUAL CONSOLE	**Incompatible changes:** • The response may include a new line containing TCP/IP information. **Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL CRYPTO	**Upwardly compatible changes:** • New response.
QUERY (Virtual Device)	**Upwardly compatible changes:** • Response indicates REAL-MPLF or SIMULATED-MPLF if enabled for the device. • Supports virtual message devices. The device type MSGD appears in the response.
QUERY VIRTUAL GRAF	**Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
QUERY VIRTUAL LINES	**Upwardly compatible changes:** • Supports the EXTended parameter to display the network qualifiers.
SET (in general)	**Upwardly compatible changes:** • See SET commands below.
SET CPTRACE	**Upwardly compatible changes:** • New trace category and trace codes for QDIO instructions.
SET CRYPTO	**Upwardly compatible changes:** • New operands: MODIFY, ON, OFF. • Changed responses. • Changed messages: HCP1706I, HCP1709E, HCP1710E, HCP1711I. • New messages: HCP1714E, HCP1715E.

Table 103 (Page 5 of 5). CP Commands Changed since VM/ESA 1.2.2

Command	Changes
SPTAPE	**Upwardly compatible changes:** • The first two digits of the 4-digit year are included in hexadecimal format in the SFBLOK dumped to tape.
SPXTAPE	**Upwardly compatible changes:** • Supports new option: PROGress_interval
SPXTAPE DUMP	**Upwardly compatible changes:** • Supports new option: PROGress_interval
SPXTAPE LOAD	**Upwardly compatible changes:** • Supports new option: PROGress_interval
SPXTAPE SCAN	**Upwardly compatible changes:** • Supports new option: PROGress_interval
STORE (Registers)	**Upwardly compatible changes:** • Supports new operand: FPC *hexword*. • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New messages: HCP6153E, HCP6154E.
STORE STATUS	**Upwardly compatible changes:** • 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.
TRACE (in general)	**Incompatible changes:** • New message: HCP1038E. **Upwardly compatible changes:** • New command responses. • See TRACE commands below.
TRACE *mnemonic1*	**Upwardly compatible changes:** • Supports new mnemonics: BSA.
TRACE *mnemonic2*	**Upwardly compatible changes:** • New mnemonic: SIGA.

SYSTEM CONFIG Statements

Table 104 lists SYSTEM CONFIG file statements that have changed since VM/ESA 1.2.2. See *VM/ESA: Planning and Administration* for complete descriptions of SYSTEM CONFIG statements.

Table 104. SYSTEM CONFIG Statements Changed since VM/ESA 1.2.2

Statement	Changes
CHARACTER_DEFAULTS	**Incompatible changes:** • Letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be defined as the system default line edit symbols (line-delete, character-delete, escape, line-end, and tab).

User Directory Control Statements

Table 105 lists the user directory control statements that have changed since VM/ESA 1.2.2. See the *VM/ESA: Planning and Administration* book for complete descriptions of user directory control statements.

Note: You can use the VMUDQ macro to query the CP user directory. See the *VM/ESA: CP Programming Services* book for more information on this macro.

Table 105 (Page 1 of 2). User Directory Control Statements Changed since VM/ESA 1.2.2

Statement	Changes
CRYPTO	**Upwardly compatible changes:** • New operand: MODIFY.
DASDOPT	**Upwardly compatible changes:** • Supports the NOWRKALLEG option.
DATEFORMAT	**New** Specifies a user's default date format for commands that provide multiple date formats.
GLOBALDEFS	**New** Signifies the beginning of the global definition section.
GLOBALOPTS	**New** Used to define global settings to be used while processing user definitions.
LOAD	**Upwardly compatible changes:** • Supports a new G operand for global definitions.
OPTION	**Upwardly compatible changes:** • 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. • Specifying the TODENABLE operand allows a user to change the virtual machine TOD clock with the new CP SET VTOD command. • Supports new operands: CFVM, CFUSER, DIAG88.
POSIXGLIST	**New** Specifies all POSIX groups of which the user is a member.
POSIXGROUP	**New** Defines a POSIX group.
POSIXINFO	**New** Specifies a user's POSIX information.
POSIXOPT	**New** Specifies option settings related to a user's POSIX capabilities.
SPECIAL	**Upwardly compatible changes:** • Supports new operand: MSGPROC.
STDEVOPT	**Upwardly compatible changes:** • Supports new DASDSYS operands: DATAMOVER and NODATAMOVER.

Table 105 (Page 2 of 2). User Directory Control Statements Changed since VM/ESA 1.2.2

Statement	Changes
USER	**Incompatible changes:** • Letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be defined as logical line edit symbols (line-end, line-delete, character-delete, and escape).

CP Utilities

Table 106 lists the utility programs that have changed since VM/ESA 1.2.2. See the *VM/ESA: CP Command and Utility Reference* for details on utility programs.

Table 106. Utility Programs Changed since VM/ESA 1.2.2

Utility	Changes
DIRECTXA	**Incompatible changes:** • The format of the response has changed to include the version. • 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'0F' (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.
SALIPL	**Incompatible changes:** • Message HCP039E deleted, replaced by new message HCP394E with same text. • Date field on the file list panel displays the year with 4 digits.

CP DIAGNOSE Codes

Table 107 lists the DIAGNOSE codes that have changed since VM/ESA 1.2.2. See the *VM/ESA: CP Programming Services* book for complete information on the DIAGNOSE codes discussed in this section.

Table 107 (Page 1 of 2). DIAGNOSE Codes Changed since VM/ESA 1.2.2

Code	Changes
X'00'	Storage extended identification code **Incompatible changes:** • The Environment field has changed from 3 bytes to 2 bytes. • The new Version Information field (formerly the third byte of the Environment field) identifies the version number of the product (for VM/ESA 2.1.0 and later). **Upwardly compatible changes:** • The value in the program product bit map has changed to indicate the new release level. Also, Bit 13 (X'0004000000000000') indicates whether Year 2000 support is present in CP.
X'14'	Input spool file manipulation **Upwardly compatible changes:** • For subcodes X'0004', X'0008', X'0FFE', and X'0FFF', a one-byte century indicator was added to the SFBLOK data area.
X'68'	Virtual Machine Communication Facility (VMCF) **Upwardly compatible changes:** • Supports a new function: SETLIMIT (Subcode X'000C').

Table 107 (Page 2 of 2). DIAGNOSE Codes Changed since VM/ESA 1.2.2

Code	Changes
X'7C'	Logical Device Support Facility **Upwardly compatible changes:** • 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.
X'84'	Directory Update-in-Place **Incompatible changes:** • For EDITCHAR operation, letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be specified as logical line edit symbols (line-end, line-delete, character-delete, and escape). **Upwardly compatible changes:** • Can now replace the following new information: – User's default date format setting • New operation: DATEFMT. • New return codes due to new function (in hex): 122, 123, 124.
X'BC'	Open and Query Spool File Characteristics **Upwardly compatible changes:** • 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.
X'D8'	Read spool file blocks on the system spool file queues **Upwardly compatible changes:** • For subcode X'0000', a one-byte century indicator was added to the SFBLOK data area.

CP Messages

The following CP messages do not exist in VM/ESA 2.4.0:

HCP543A
HCP1365E
HCP8039S
HCP8611T

The following list identifies the CP messages that have changed since VM/ESA 1.2.2. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CP messages.

HCP006E
HCP031E
HCP114E
HCP145I
HCP233E
HCP296E
HCP319E
HCP332E
HCP403I
HCP481E
HCP512I
HCP580I
HCP592I
HCP704E
HCP711D
HCP716D
HCP717D
HCP725D
HCP751E
HCP752E
HCP753E
HCP772E
HCP799E
HCP1003E
HCP1006E
HCP1010W
HCP1016E
HCP1018E
HCP1115E
HCP1116E
HCP1120E
HCP1365E
HCP1512E
HCP1706I
HCP1709E
HCP1710E
HCP1711I
HCP1791E
HCP1879E
HCP2002I
HCP2234E
HCP2252E
HCP2760E
HCP2768E
HCP2779E
HCP6014I
HCP6111I
HCP6283I

HCP6706E
HCP6739E
HCP6743E
HCP6788E
HCP6802E
HCP6804E
HCP8028W
HCP8080I
HCP9020W
HCP9021W
HCP9022W
HCP9036W
HCP9225I
HCP9405E
HCP9408E
HCP9410I
HCP9422E

CMS Changes

This section identifies the CMS externals that have changed since VM/ESA 1.2.2. It contains the following subsections:

- “CMS Commands”
- “CMS File Pool Administration and Operator Commands” on page 558
- “XEDIT Subcommands” on page 559
- “CMS Macros” on page 560
- “Preferred Routines (CSL Routines)” on page 562
- “Compatibility Routines” on page 569
- “CMS Pipelines” on page 569
- “CMS Messages” on page 570

CMS Commands

Table 108 lists the CMS commands that have changed since VM/ESA 1.2.2. See the *VM/ESA: CMS Command Reference* for complete descriptions of CMS commands.

Note: The three character module identifiers have been removed from the messages listed in the *VM/ESA: CMS Command Reference*. For example, a message that used to be listed as DMSAMS136S is now listed as DMS136S.

Table 108 (Page 1 of 6). CMS Commands Changed since VM/ESA 1.2.2

Command	Changes
ACCESS	**Upwardly compatible changes:** • Added message for use with BFS: DMS2133E.
COPYFILE	**Incompatible changes:** • New message: DMS516E.
CREATE DIRECTORY	**Upwardly compatible changes:** • New message for authorization failure from ESM: DMS1331E.
CREATE LOCK	**Upwardly compatible changes:** • Added messages for use with BFS: DMS2040E, DMS2133E. • Added new operand for use with BFS: *bfsid*.
CSLLIST	**Upwardly compatible changes:** • 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.
CSLMAP	**Upwardly compatible changes:** • 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.

Table 108 (Page 2 of 6). CMS Commands Changed since VM/ESA 1.2.2

Command	Changes
DEFAULTS	**Upwardly compatible changes:** • Supports new options: NOKEEPCC, KEEPCC. • New options are supported as parameters for FILELIST, NETDATA, and RDRLIST: VMDATE, SHORTDATE, FULLDATE, ISODATE.
DELETE LOCK	**Upwardly compatible changes:** • Added new operand for use with BFS: *bfsid*.
DIRLIST	**Upwardly compatible changes:** • 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.
EDIT	**Incompatible changes:** • The '(OLD' option is no longer supported. You must either use the XEDIT simulation by dropping the '(OLD' option or convert to using the normal XEDIT environment. • Message DMS987E was changed to DMS2520E. The old CMS editor is no longer supported.
ERASE	**Upwardly compatible changes:** • Added new operand for use with BFS: *bfsid*. • New message for authorization failure from ESM: DMS1332E.
FILEDEF	**Upwardly compatible changes:** • New option: LIBSRV. • Allows LRECL definitions up to 65535 bytes for OS variable spanned records (under XLRI processing) and non-OS CMS files. • Allows BLKSIZE definitions up to 65535 bytes for non-OS CMS files.
FILELIST	**Incompatible changes:** • 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 LISTFILE profiles. See Appendix A of the *VM/ESA: CMS Command Reference*. • If you file the file created by FILELIST, that file might contain new and changed fields (on the far right). **Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • The BEFORE *date* and AFTER *date* options support 4-digit years. • Screens and responses support 4-digit years. • If a date format option is not specified on the FILELIST command, the CMS DEFAULTS date format setting for FILELIST will be used. • Added message for use with BFS: DMS2133E. • 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.
FINIS	**Incompatible changes:** • Will close files opened by the REXX STREAM I/O function.
GLOBAL	**Upwardly compatible changes:** • New message for duplicate library name in input list: DMS045W. The duplicates are ignored.

Table 108 (Page 3 of 6). CMS Commands Changed since VM/ESA 1.2.2

Command	Changes
IDENTIFY	**Upwardly compatible changes:** • New options to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses support 4-digit years. • 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. • New option for TCP/IP mail integration: TCPIP.
LISTDS	**Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Responses support 4-digit years. • 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.
LISTFILE	**Incompatible changes:** • Message DMS550E changed to DMS765E. **Upwardly compatible changes:** • New options to specify date format: SHORTDATE, FULLDATE, ISODATE. • The BEFORE *date* and AFTER *date* options support 4-digit years. • Responses support 4-digit years. • 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.
LKED	**Upwardly compatible changes:** • The defaults for *value1* and *value2* on the SIZE option have been increased to 400K and 100K, respectively.
LOAD	**Upwardly compatible changes:** • New message for insufficient storage above 16MB: DMS891W.
LOADMOD	**Incompatible changes:** • Changed message (new text possible): DMS639E.
MACLIST	**Upwardly compatible changes:** • 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.
MOVEFILE	**Upwardly compatible changes:** • Can process OS variable spanned records (under XLRI processing) and non-OS CMS files with record lengths up to 65535 bytes. • Adjusts output file sizes for compatibility between CMS and OS • Allows greater FILEDEF default flexibility for file attributes (RECFM, LRECL, BLKSIZE) • If default size values are used, fixes record truncation problems when moving data files from fixed to variable format • New message: DMS1116E.

Table 108 (Page 4 of 6). CMS Commands Changed since VM/ESA 1.2.2

Command	Changes
NETDATA	**Incompatible changes:** • Different message on empty reader condition: DMS205W (was DMS639E). **Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file. • Responses support 4-digit years. • 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.
NOTE	**Upwardly compatible changes:** • In the Date field of the note header, the year is now displayed with four digits. • TCP/IP domain names accepted as user IDs or as the resolution of nicknames. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
NUCXLOAD	**Incompatible changes:** • Changed message (new text possible): DMS639E.
PARSECMD	**Upwardly compatible changes:** • BFS added new code.n value of PN.
PEEK	**Upwardly compatible changes:** • TCP/IP origin domain name address used when available and shown on PEEK message line for origin within current space and formatting limitations.
PIPE	**Incompatible changes:** • All CMS Pipelines messages have a new prefix, and many messages have new numbers and text. See “CMS Pipelines” on page 569.
QUERY (in general)	**Incompatible changes:** • See changed commands below. **Upwardly compatible changes:** • Supports new operands.
QUERY BLOCKS	**Upwardly compatible changes:** • Added new operand for use with BFS: *bfsid*.
QUERY CMSLEVEL	**Upwardly compatible changes:** • In the response, the CMS level value has changed.
QUERY CMSREL	**Incompatible changes:** • The format of the response has changed to include the version. **Upwardly compatible changes:** • In the response, the release level value has changed.
QUERY ENROLL	**Upwardly compatible changes:** • Added new option for use with BFS: FILESPACE.
QUERY FILEATTR	**Upwardly compatible changes:** • Added new operand for use with BFS: *bfsid*.
QUERY FILEDEF	**Upwardly compatible changes:** • Supports a new optional operand, ATTRIBUT, and its response.

Table 108 (Page 5 of 6). CMS Commands Changed since VM/ESA 1.2.2

Command	Changes
QUERY LIMITS	**Incompatible change:** • File pool administration authority is no longer needed to query limits on a filespace other than your own.
QUERY LOCK	**Upwardly compatible changes:** • Added new operand for use with BFS: *bfsid*.

QUERY SYSNAMES	**Upwardly compatible changes:**
	• There is a change to the response. The CMSGUI saved system name is displayed.
	An example of the VM/ESA 2.4.0 response:
	`SYSNAMES:  CMSVSAM  CMSAMS   CMSDOS   CMSBAM   CMSGUI`
	`ENTRIES:   CMSVSAM  CMSAMS   CMSDOS   CMSBAM  VMGUILIB`

QUERY NAMEDEF	**Upwardly compatible changes:** • Added new operand for use with BFS: *bfsid*.
RDRLIST	**Incompatible changes:** • If you file the file created by RDRLIST, that file might contain new and changed fields (on the far right). **Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Screen supports 4-digit years. • If a date format option is not specified on the RDRLIST command, the CMS DEFAULTS date format setting for RDRLIST will be used. • 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. • TCP/IP origin domain name address used when available and shown on RDRLIST panel origin area within current space and formatting limitations.
RECEIVE	**Upwardly compatible changes:** • Supports new options: NOKEEPCC, KEEPCC. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
SENDFILE	**Upwardly compatible changes:** • TCP/IP domain names accepted as user IDs or as the resolution of nicknames. • New options to specify the transmission method: SMTP, MIME, UFTSYNC, UFTASYNC. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
SET (in general)	**Upwardly compatible changes:** • Supports new operands.
SET SYSNAME	**Upwardly compatible changes:** • Supports new operand: CMSGUI
TAPE	**Upwardly compatible changes:** • 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. • Added message for Tape Library Dataserver support: DMS2147W.
TELL	**Upwardly compatible changes:** • Accepts a TCP/IP domain name as part of the destination information.

Table 108 (Page 6 of 6). CMS Commands Changed since VM/ESA 1.2.2

Command	Changes
VMFPLC2	**Upwardly compatible changes:** • 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. • Added message for Tape Library Dataserver support: DMS2147W.
VMLINK	**Upwardly compatible changes:** • Added operand, LIST, to the AUTOLINK option. • 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.
XEDIT	**Upwardly compatible changes:** • Added new option for BFS: NAMETYPE If NAMETYPE BFS is used, your profile must be in REXX and it is loaded as a REXX function. • Added new option for BFS: BFSLINE • New messages for BFS: DMS033E, DMS512E, DMS2105E, and DMS2134E.

CMS File Pool Administration and Operator Commands

Table 109 lists the CMS file pool administration and operator commands that have changed since VM/ESA 1.2.2. See the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book for complete descriptions of file pool administration and operator commands.

Table 109 (Page 1 of 2). CMS File Pool Administration and Operator Commands Changed since VM/ESA 1.2.2

Command	Changes
AUDIT	**Upwardly compatible changes:** • New operands: *fn ft*, REPLACE. • Added messages: DMS024E, DMS1258E, DMS3253I, DMS3254E, DMS3255E. • Changed message: DMS3470W (new text possible).
DELETE USER	**Upwardly compatible changes:** • New options: DELAUTH | KEEPAUTH. • Added message: DMS2023E.
ENROLL USER	**Upwardly compatible changes:** • Supports new options: SFS | BFS, *userid*, *gname*. • Added messages for use with BFS: DMS1209E, DMS2023E, DMS2132E.
FILEPOOL BACKUP	**Upwardly compatible changes:** • Supports new BFS parameter: *bfsid*.
FILEPOOL RELOAD	**Upwardly compatible changes:** • New message: DMS3455I.
FILEPOOL UNLOAD	**Upwardly compatible changes:** • New message: DMS3455I.
FILESERV START	**Incompatible changes:** • The start-up does not continue processing if the DMS3110E message is issued because of a SEGMENT ASSIGN failure.

Table 109 (Page 2 of 2). CMS File Pool Administration and Operator Commands Changed since VM/ESA 1.2.2

Command	Changes
QUERY LIMITS	**Incompatible changes:** • File pool administration authority is no longer required.

XEDIT Subcommands

Table 110 lists the XEDIT subcommands that have changed since VM/ESA 1.2.2. For complete descriptions of XEDIT subcommands, see the *VM/ESA: XEDIT Command and Macro Reference*.

Table 110 (Page 1 of 2). XEDIT Subcommands Changed since VM/ESA 1.2.2

Subcommand	Changes
CMS	**Upwardly compatible change:** • New message added: DMS642E
CP	**Upwardly compatible change:** • New message added: DMS642E
EXTRACT	**Upwardly compatible change:** • New operands added: BFSLine, EPName, NAMetype, PName, and GUI.
FILE	**Upwardly compatible changes:** • Support for BFS files has been added. • New messages for BFS support: 024E, 054E, 1184E, 2105E, 2120E, 2131E, and 2134E.
GET	**Upwardly compatible changes:** • Support for BFS files has been added. • New messages for BFS support: 033E, 054E, 512E, 2105E, 2131E, and 2134E.
LOAD	**Upwardly compatible changes:** • Support for writing to BFS files has been added. • Added new options: NAMetype and BFSLine. • New messages for BFS support: 033E, 512E, 2105E, and 2134E.
MODIFY	**Upwardly compatible changes:** • New operands added: BFSLine, NAMetype, and PName.
PRESERVE	**Upwardly compatible changes:** • New operands added: BFSLine, NAMetype, and PName.
PUT	**Upwardly compatible changes:** • Support for writing to BFS files has been added. • New return code for BFS support: 32. • New messages for BFS support: 033E, 054E, 512E, 1184E, 2105E, 2120E, 2131E, and 2134E.
PUTD	**Upwardly compatible changes:** • Support for writing to BFS files has been added. • New return code for BFS support: 32. • New messages for BFS support: 033E, 054E, 512E, 1184E, 2105E, 2120E, 2131E, and 2134E.
QUERY	**Upwardly compatible change:** • New operand added.

Table 110 (Page 2 of 2). XEDIT Subcommands Changed since VM/ESA 1.2.2

Subcommand	Changes
SAVE	**Upwardly compatible changes:** • Support for writing to BFS files has been added. • New return code for BFS support: 32. • New messages for BFS support: 024E, 054E, 512E, 1184E, 2105E, 2120E, 2131E, and 2134E.
SET	**Upwardly compatible change:** • New operand added.

CMS Macros

Table 111 lists the CMS preferred macros that have changed since VM/ESA 1.2.2. Table 112 lists the CMS compatibility macros that have changed. Table 113 lists the OS simulation macros that have changed. Table 114 lists the simulated OS/MVS supervisor calls that have changed.

Preferred Macros

See the *VM/ESA: CMS Application Development Reference for Assembler* for complete descriptions of these macros.

Table 111 (Page 1 of 2). CMS Preferred Macros Changed since VM/ESA 1.2.2

Macro	Changes
AMODESW	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370
CMSCVT	**Upwardly compatible changes:** • Added the fields: CVTECVT, CVTFLAG2.
CMSLEVEL	**Incompatible changes:** • The CMS level has been frozen at X'0F' for CMS Level 12 (VM/ESA Version 2 Release 1.0) and later. Use the new DMSQEFL macro or the DMSQEFL CSL routine instead.
CSLENTRY	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370
CSLFPI	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370
DIRBUFF	**Upwardly compatible changes:** • The FILE record contains the following new fields: DIRFDAXD, DIRFDAXC, DIRFDAXI, DIRFLV13. • The FILEEXT record contains the following new fields: DIREDAXD, DIREDAXC, DIREDAXI, DIREDRXD, DIREDRXC, DIREDRXI, DIRECDXD, DIRECDXC, DIRECDXI, DIREDCXD, DIREDCXC, DIREDCXI, DIRELV13. • The SEARCHALL and SEARCHAUTH records contain the following new fields: DIRSDAXD, DIRSDAXC, DIRSDAXI, DIRSCEND, DIRSLV13.
ENABLE	**Upwardly compatible changes:** • Supports new parameter: MODE=NO370

Table 111 (Page 2 of 2). CMS Preferred Macros Changed since VM/ESA 1.2.2

Macro	Changes
EXSBUFF	**Upwardly compatible changes:** • The FILE record contains the following new fields: EXSFDAXD, EXSFDAXC, EXSFDAXI, EXSFDRXD, EXSFDRXC, EXSFDRXI, EXSFCDXD, EXSFCDXC, EXSFCDXI, EXSFDCXD, EXSFDCXC, EXSFDCXI, EXSF2000, EXSFLV13. • In the FILE record, the following field has changed: EXSFRES. • The DIR record contains the following new fields: EXSDDCXD, EXSDDCXC, EXSDDCXI, EXSDCDXD, EXSDCDXC, EXSDCDXI, EXSD2000, EXSDLV13. • In the DIR record, the following field has changed: EXSDRES.
FSSTATE	**Upwardly compatible changes:** • 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*nn*, 1=20*nn*, where *nn* is the 2-digit year). In VM/ESA 1.2.2, this bit was not used.
FSTD	**Upwardly compatible changes:** • The FSTFLAGS section contains the new FSTCNTRY field, which is a bit that indicates the century (first two digits of the year) the file was last written or updated (0=19*nn*, 1=20*nn*, where *nn* is the 2-digit year).
TAPECTL	**Upwardly compatible changes:** • 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.

Compatibility Macros

See *VM/ESA: Planning and Administration* for information about the DEFNUC macro.

Table 112. CMS Compatibility Macros Changed since VM/ESA 1.2.2

Macro	Changes
DEFNUC	**Incompatible changes:** • The format of the default CMS IPL heading (used when the VERSION= parameter is specified without a value in DEFNUC) has changed to include the version. Also, the date in the default IPL heading is presented in ISO format (*yyyy-mm-dd*). **Upwardly compatible changes:** • The USEMTSEG and MTSEG parameters no longer have any effect because VMMTLIB is no longer created as a saved segment; it now resides within the CMS nucleus.

OS Simulation Macros and Supervisor Calls

See the *VM/ESA: CMS Application Development Guide for Assembler* for information about the OS macros and supervisor calls that CMS simulates.

Table 113 (Page 1 of 2). OS Simulation Macros Changed since VM/ESA 1.2.2

Macro	Changes
CLOSE	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.

Table 113 (Page 2 of 2). OS Simulation Macros Changed since VM/ESA 1.2.2

Macro	Changes
DCB	**Upwardly compatible changes:** • 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.
GET	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.
OPEN	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
PUT	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.
TIME	**Upwardly compatible changes:** • The 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.

Table 114. OS Simulation Supervisor Calls Changed since VM/ESA 1.2.2

SVC	Changes
SVC 19 (OPEN)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
SVC 20 (CLOSE)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.

Preferred Routines (CSL Routines)

Table 115 lists the CSL routines that have changed since VM/ESA 1.2.2. See the *VM/ESA: CMS Application Development Reference* for complete descriptions of all the routines in this table except QueueOpen. The description of QueueOpen is in *VM/ESA: CMS Application Multitasking*.

Table 115 (Page 1 of 8). CSL Routines Changed since VM/ESA 1.2.2

Routine	Changes
DMSCLBLK	**Upwardly compatible changes:** • New reason codes when using BFS: 10220, 65400. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330, 90492, 90495.

Table 115 (Page 2 of 8). CSL Routines Changed since VM/ESA 1.2.2

Routine	Changes
DMSCLDBK	**Upwardly compatible changes:** • New reason code when using BFS: 65400. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330, 90495.
DMSCLOSE	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330, 90495.
DMSCRDIR	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSCRFIL	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* and *create_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSCRLOC	**Upwardly compatible changes:** • New reason codes when using BFS: 65400, 69200, and 69300. • Supports new BFS parameter: *bfsid*.
DMSCROB	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* and *create_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSDELOC	**Upwardly compatible changes:** • Supports new BFS parameter: *bfsid*.
DMSDEUSR	**Upwardly compatible changes:** • New parameters: DELAUTH, KEEPAUTH, and *length4*. • New reason code: 98700.
DMSENUSR	**Upwardly compatible changes:** • New reason codes when using BFS: 10210, 10240, 69000, 90300, 90320, and 98700. • Supports new options: SFS | BFS, *userid*, *gname*. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE, *length7*. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90330, 90495.
DMSERASE	**Upwardly compatible changes:** • Supports new BFS parameter: *bfsid*.
DMSERP	**Upwardly compatible changes:** • New information names for Year 2000 support (FILE_DATE_CENTURY, ACT_FILE_DATE_CENTRY (note that U is omitted), and YEAR2000_SUPPORT).

Table 115 (Page 3 of 8). CSL Routines Changed since VM/ESA 1.2.2

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSEXIDI</td><td><b>Incompatible changes:</b>
• When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.2, it was returned as 8 characters.
<b>Upwardly compatible changes:</b>
• Supports new BFS parameters: <i>bfsid</i>, <i>filesystemtype</i>
• New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.
• The <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.
• New reason codes: 90320, 90330.</td></tr>
<tr><td>DMSEXIFI</td><td><b>Incompatible changes:</b>
• When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.2, it was returned as 8 characters.
<b>Upwardly compatible changes:</b>
• Supports new BFS parameters: <i>bfsid</i>, <i>filesystemtype</i>
• New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.
• The <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.
• New reason codes: 90320, 90330.</td></tr>
<tr><td>DMSEXIST</td><td><b>Upwardly compatible changes:</b>
• Supports new BFS parameter: <i>bfsid</i>
• The following offsets have changed in the FILE data record (and the record length has increased to 436 bytes):
<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>1 (X'1')</td><td><i>file_system_type</i></td><td>Reserved in VM/ESA 1.2.2</td></tr>
<tr><td>339 (X'153')</td><td><i>dec_date_ext</i></td><td>Reserved in VM/ESA 1.2.2</td></tr>
<tr><td>343 (X'157')</td><td><i>date_ext</i></td><td>Added</td></tr>
<tr><td>353 (X'161')</td><td><i>iso_date_ext</i></td><td>Added</td></tr>
<tr><td>363 (X'16B')</td><td><i>dec_dateref_ext</i></td><td>Added</td></tr>
<tr><td>367 (X'16F')</td><td><i>dateref_ext</i></td><td>Added</td></tr>
<tr><td>377 (X'179')</td><td><i>iso_dateref_ext</i></td><td>Added</td></tr>
<tr><td>387 (X'183')</td><td><i>dec_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>391 (X'187')</td><td><i>cr_date_ext</i></td><td>Added</td></tr>
<tr><td>401 (X'191')</td><td><i>iso_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>411 (X'19B')</td><td><i>dec_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>415 (X'19F')</td><td><i>last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>425 (X'1A9')</td><td><i>iso_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>435 (X'1B3')</td><td>Reserved</td><td>Added</td></tr>
</table>
• The following offset has changed in the DIRECTORY data record (and the record length has increased to 308 bytes):
<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>1 (X'1')</td><td><i>file_system_type</i></td><td>Reserved in VM/ESA 1.2.2</td></tr>
<tr><td>254 (X'FE')</td><td><i>dec_last_change_date_ext</i></td><td>Reserved in VM/ESA 1.2.2</td></tr>
<tr><td>258 (X'102')</td><td><i>last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>268 (X'10C')</td><td><i>iso_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>278 (X'116')</td><td><i>dec_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>282 (X'11A')</td><td><i>cr_date_ext</i></td><td>Added</td></tr>
<tr><td>292 (X'124')</td><td><i>iso_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>302 (X'12E')</td><td>Reserved</td><td>Added</td></tr>
</table></td></tr>
</table>

Table 115 (Page 4 of 8). CSL Routines Changed since VM/ESA 1.2.2

Routine	Changes
DMSGETDA	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.2, it was returned as 8 characters. **Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330.
DMSGETDF	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.2, it was returned as 8 characters. **Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330.

Table 115 (Page 5 of 8). CSL Routines Changed since VM/ESA 1.2.2

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSGETDI</td><td>

Upwardly compatible changes:

- The following offsets have changed in the FILE data record (and the record length has increased to 112 bytes):

<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>88 (X'58')</td><td><i>dec_date_ext</i></td><td>Added</td></tr>
<tr><td>92 (X'5C')</td><td><i>date_ext</i></td><td>Added</td></tr>
<tr><td>102 (X'66')</td><td><i>iso_date_ext</i></td><td>Added</td></tr>
</table>

- The following offsets have changed in the FILEEXT data record (and the record length has increased to 284 bytes):

<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>1 (X'1')</td><td><i>file_system_type</i></td><td>Reserved in VM/ESA 1.2.2</td></tr>
<tr><td>185 (X'B9')</td><td><i>dec_date_ext</i></td><td>Reserved in VM/ESA 1.2.2</td></tr>
<tr><td>189 (X'BD')</td><td><i>date_ext</i></td><td>Added</td></tr>
<tr><td>199 (X'C7')</td><td><i>iso_date_ext</i></td><td>Added</td></tr>
<tr><td>209 (X'D1')</td><td><i>dec_dateref_ext</i></td><td>Added</td></tr>
<tr><td>213 (X'D5')</td><td><i>dateref_ext</i></td><td>Added</td></tr>
<tr><td>223 (X'DF')</td><td><i>iso_dateref_ext</i></td><td>Added</td></tr>
<tr><td>233 (X'E9')</td><td><i>dec_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>237 (X'ED')</td><td><i>cr_date_ext</i></td><td>Added</td></tr>
<tr><td>247 (X'F7')</td><td><i>iso_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>257 (X'101')</td><td><i>dec_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>261 (X'105')</td><td><i>last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>271 (X'10F')</td><td><i>iso_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>281 (X'119')</td><td>Reserved</td><td>Added</td></tr>
</table>

- The following offset has changed in the LOCK data record:

<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>1 (X'1')</td><td><i>file_system_type</i></td><td>Reserved in VM/ESA 1.2.2</td></tr>
</table>

- The following offsets have changed in the SEARCHALL and SEARCHAUTH data records (and the record length has increased to 252 bytes):

<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>226 (X'E2')</td><td>Reserved</td><td>Added</td></tr>
<tr><td>228 (X'E4')</td><td><i>dec_date_ext</i></td><td>Added</td></tr>
<tr><td>232 (X'E8')</td><td><i>date_ext</i></td><td>Added</td></tr>
<tr><td>242 (X'F2')</td><td><i>iso_date_ext</i></td><td>Added</td></tr>
</table>

</td></tr>
<tr><td>DMSGETDK</td><td>

Upwardly compatible changes:

- Supports new BFS parameter: *filesystemtype*

</td></tr>
<tr><td>DMSGETDS</td><td>

Incompatible changes:

- When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.2, it was returned as 8 characters.

Upwardly compatible changes:

- New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*.
- The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.
- New reason codes: 90310, 90320, 90330.

</td></tr>
</table>

Table 115 (Page 6 of 8). CSL Routines Changed since VM/ESA 1.2.2

Routine	Changes
DMSGETDX	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.2, it was returned as 8 characters. **Upwardly compatible changes** • Supports new BFS parameter: *filesystemtype* • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*. • The *date*, *dateref*, *create_date*, and *update_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330.
DMSOPBLK	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 1.2.2, it was returned as 8 characters. **Upwardly compatible changes:** • Supports new parameters: *dateref*, *bfsid* • New reason codes when using BFS: 10220, 65400, 98700. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date*, *create_date*, and *dateref* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSOPCAT	**Upwardly compatible changes:** • Supports new BFS parameter: *bfsid* • Supports new BFS options: FILEATTR SFS|BFS,READEXT • New reason codes when using BFS: 90300, 90350.
DMSOPDBK	**Upwardly compatible changes:** • Supports new BFS parameter: *bfsid* • New reason code when using BFS: 10220. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *create_date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSOPDIR	**Upwardly compatible changes:** • Supports new BFS parameter: *bfsid* • New reason code when using BFS: 65400.
DMSOPEN	**Upwardly compatible changes** • Supports new BFS parameter: *bfsid* • New reason code when using BFS: 10220. • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *create_date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSQEFL	**Upwardly compatible changes:** • Supports new values for *cp_level* and *cms_level*
DMSQLIMA	**Incompatible changes:** • File pool administration authority no longer required. **Upwardly compatible changes:** • Supports new BFS parameter: *filesystemtype*

Table 115 (Page 7 of 8). CSL Routines Changed since VM/ESA 1.2.2

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSQLIMD</td><td><b>Upwardly compatible changes:</b>
• Supports new BFS parameter: <i>filesystemtype</i></td></tr>
<tr><td>DMSQLIMU</td><td><b>Incompatible changes:</b>
• File pool administration authority no longer required.
<b>Upwardly compatible changes:</b>
• Supports new BFS parameter: <i>filesystemtype</i></td></tr>
<tr><td>DMSQSFSL</td><td><b>Upwardly compatible changes:</b>
• Supports new value for <i>server_level</i></td></tr>
<tr><td>DMSRDCAT</td><td><b>Upwardly compatible changes:</b>
• New catalogs for BFS support: NAMECAT, NOVCAT.
• New OBJECTCAT record for BFS object.
• SPACECAT record — the four reserved CHAR(8) fields at the end of the record (following the STORAGEGROUP field) have been restructured for BFS support as follows:
<table>
<tr><th>Field Name</th><th>Field Type/Description</th></tr>
<tr><td>HIGHINO</td><td>INTEGER(4) High OBJECTCAT INO value</td></tr>
<tr><td>HIGHNID</td><td>INTEGER(4) High NAMECAT NID value</td></tr>
<tr><td>FLAGS</td><td>CHAR(1) File space attributes</td></tr>
<tr><td>RESERVED01</td><td>CHAR(7) Reserved</td></tr>
<tr><td>RESERVED02</td><td>CHAR(8) Reserved</td></tr>
<tr><td>RESERVED03</td><td>CHAR(8) Reserved</td></tr>
</table>
• DIRCAT record — new bit settings in the DIRATTS field for BFS support.
• OBJECTCAT record for SFS:
– New bit settings in TYPE and FILEFLAGS fields for BFS support.
– New bit settings in FILEFLAGS field to support 4-digit years (century setting for the DATE and DATEREF fields).
– The reserved CHAR(1) field preceding the LAST_CHANGE_DATE field has changed to:
<table>
<tr><th>Field Name</th><th>Field Type/Description</th></tr>
<tr><td>CHGDATE_CENTURY</td><td>CHAR(1) Century byte for LAST_CHANGE_DATE</td></tr>
</table>
– The reserved CHAR(1) field preceding the CREATIONDATE field has changed to:
<table>
<tr><th>Field Name</th><th>Field Type/Description</th></tr>
<tr><td>CREATIONDATE_CENTURY</td><td>CHAR(1) Century byte for CREATIONDATE</td></tr>
</table>
• FQFN record — new H type record for open intent READEXT.
• AUTHCAT record — new G type record for open intent READEXT.
• EOCAT record — new bit settings in TYPE field for BFS support.</td></tr>
<tr><td>DMSTRUNC</td><td><b>Upwardly compatible changes:</b>
• New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.
• The <i>date</i> parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.
• New reason code: 90495.</td></tr>
<tr><td>DMSVALDT</td><td><b>Upwardly compatible changes:</b>
• Supports new BFS parameter: <i>bfsid</i>.</td></tr>
</table>

Table 115 (Page 8 of 8). CSL Routines Changed since VM/ESA 1.2.2

Routine	Changes
QueueOpen	**Upwardly compatible changes:** • In VM/ESA 1.2.2, if you tried to use QueueOpen to open a remotely-located network-level queue but forgot to set up the CMS Communication Directory (that is, installed no ComDir entry describing the location of the remote queue), then QueueOpen would fail with reason code *vm_ipc_comm_lost*. In VM/ESA 2.4.0, QueueOpen will proceed anyway, using default values for what it tried to extract from the ComDir. The default values used are described in *VM/ESA: CMS Application Multitasking*.

Compatibility Routines

Table 116 lists the CMS compatibility routines that have changed between VM/ESA 1.2.2 and VM/ESA 2.4.0. See the *VM/ESA: CMS Application Development Guide for Assembler* for more information about the routines in this table.

Table 116. CMS Compatibility Routines Changed since VM/ESA 1.2.2

CMS Routine	Explanation
DMSTVS	**Upwardly compatible changes:** • Supports new LIBSRV plist parameter. • 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.

CMS Pipelines

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.
- 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.

  Note: User-written applications that are sensitive to these changes may require alterations. Published externals (such as PIPGFTXT) have **not** been changed.
- 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. Some specific enhancements that have occurred since VM/ESA 2.1.0 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 *VM/ESA: CMS Pipelines Reference* can be found in the *CMS/TSO Pipelines: Author's Edition*, which is included with the VM/ESA 2.4.0 library.

CMS Messages

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 H, “CMS Pipelines Message Cross-Reference [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]” on page 669.

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/ESA.

The following CMS message does not exist in VM/ESA 2.4.0:

DMS683E

The following CMS messages have changed since VM/ESA 1.2.2. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CMS messages.

DMS002E
DMS003E
DMS005E
DMS007E
DMS017E
DMS023E
DMS024E
DMS026E
DMS029E
DMS033E
DMS036E
DMS037E
DMS042E
DMS054E
DMS056E
DMS062E
DMS069I
DMS106S
DMS132S
DMS149E
DMS250E
DMS344E
DMS358E
DMS360E
DMS361E
DMS363R
DMS364I
DMS365I
DMS366R
DMS516E
DMS531E
DMS595E
DMS616W
DMS618E
DMS622E
DMS639E
DMS651E
DMS1127I
DMS1131E
DMS1137E
DMS1138E
DMS1139E
DMS1151E
DMS1153E
DMS1162E
DMS1176E
DMS1184E
DMS1205I
DMS1229E
DMS1239E
DMS1258E
DMS1259E
DMS1306T
DMS1311E
DMS1339S
DMS1341S
DMS1342S
DMS1343S
DMS1344S
DMS2008E
DMS2010E
DMS2013E
DMS2023E
DMS2031E
DMS2040E
DMS2041E
DMS2153E
DMS2154E
DMS2155E
DMS2523E
DMS2731I
DMS2982E
DMS3000W
DMS3009R
DMS3088R
DMS3110E
DMS3208E
DMS3284E
DMS3431E
DMS3438E
DMS3438I
DMS3453E
DMS3454E
DMS3455I
DMS3470W
DMS3494R
DMS3508E
DMS3514E
DMS3515E
DMS3518E
DMS3521E
DMS3522E
DMS3526E
DMS3527E
DMS3528E
DMS3529E
DMS3530E
DMS3531E
DMS3532E
DMS3557E
DMS3561E
DMS3562E
DMS3585E
DMS3594R
DMS3616W
DMS3617I
DMS3618W
DMS3620I
DMS3622E
DMS3623E
DMS3624W
DMS3627E
DMS3628E
DMS3629E
DMS3630W
DMS3631W
DMS3632W
DMS3636E
DMS3641W
DMS3642W
DMS3727E
DMS3728E
DMS3926E

REXX/VM Changes

This section identifies the REXX/VM externals that have changed since VM/ESA 1.2.2. It contains the following subsections:

- "REXX/VM Functions"
- "External Functions"

REXX/VM Functions

Table 117 lists the REXX/VM functions that have changed since VM/ESA 1.2.2. For complete descriptions of these functions, see the *VM/ESA: REXX/VM Reference*.

Table 117. REXX/VM Functions Changed since VM/ESA 1.2.2

Function	Changes
DATE	**Upwardly compatible changes:** • New parameters allow you to specify a date to be converted to a different format. • New parameters: *output_separator_char*, *input_separator_char*.

External Functions

Table 118 lists external functions which can be used by REXX/VM that have changed since VM/ESA 1.2.2. For complete descriptions of these functions, see the *VM/ESA: REXX/VM Reference*.

Table 118. External Functions Changed since VM/ESA 1.2.2

Function	Changes
CMSFLAG	**Upwardly compatible changes:** • New value for *flag*: YEAR2000.
DIAG DIAGRC	**Upwardly compatible changes:** • New DIAGNOSE code is supported: X'270'.

GCS Changes

This section identifies the GCS externals that have changed since VM/ESA 1.2.2. It contains the following subsection:

- "GCS Commands and Macros"

GCS Commands and Macros

Table 119 lists the GCS commands and macros that have changed since VM/ESA 1.2.2. See *VM/ESA: Group Control System* for complete descriptions of all GCS commands and macros.

Table 119 (Page 1 of 2). GCS Commands and Macros Changed since VM/ESA 1.2.2

Command/Macro	Changes
CVT macro	**Upwardly compatible changes:** • Added the fields: CVTECVT, CVTFLAG2.

Table 119 (Page 2 of 2). GCS Commands and Macros Changed since VM/ESA 1.2.2

Command/Macro	Changes
GCSLEVEL macro	**Upwardly compatible changes:** • Additional equates for new release levels.
GETMAIN macro	**Incompatible changes:** • In VM/ESA 1.2.2, although LOC=RES was documented as the default, the actual default was LOC=BELOW, and all the requested virtual storage was allocated below 16MB. In VM/ESA 2.2.0, the actual default was 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.
ITRACE command	**Upwardly compatible changes:** • New operand: SP.
QUERY GCSLEVEL command	**Incompatible changes:** • The format of the response has changed to include the version: `VM/ESA Version 2 Release 4.0, Service Level 0`
QUERY MODDATE command	**Incompatible changes:** • A full 4-digit year is now returned in the date field of the response instead of a 2-digit year.
TIME macro	**Upwardly compatible changes:** • The 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.

VMSES/E Changes

This section identifies the VMSES/E externals that have changed since VM/ESA 1.2.2. It contains the following subsection:

- "VMSES/E Messages"

VMSES/E Messages

The following VMSES/E messages have changed since VM/ESA 1.2.2. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of VMSES/E messages.

VMF002E
VMF1821E
VMF2066E
VMF2114R
VMF2118I
VMF2119I
VMF2120W
VMF2152E
VMF2162E
VMF2173I
VMF2200E
VMF2201E
VMF2202E
VMF2206E
VMF2206W
VMF2215E
VMF2225E
VMF2228E
VMF2507I
VMF2509I
VMF2760I
VMF2767I
VMF2780E
VMF2786E
VMF2867W

Programmable Operator Facility Changes

This section identifies the Programmable Operator Facility externals that have changed since VM/ESA 1.2.2. This section contains the following subsection:

- "Programmable Operator Facility Commands and Routing Table Statements"

Also see "Summary of Service Exec Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]" on page 250.

Programmable Operator Facility Commands and Routing Table Statements

Table 120 lists the Programmable Operator Facility commands and routing table statements that have changed between VM/ESA 1.2.2 and VM/ESA 2.4.0. See *VM/ESA: Planning and Administration* for complete descriptions of programmable operator facility commands and routing table statements.

Table 120. Programmable Operator Facility Commands and Routing Table Statements Changed since VM/ESA 1.2.2

Programmable Operator Command or Statement	**Explanation**
MSGLIMIT	**Upwardly compatible changes:** • New routing table configuration entry statement.

Chapter 21. Compatibility Tables for Converting from VM/ESA 2.1.0

This chapter identifies the VM/ESA externals that have changed between VM/ESA 2.1.0 and VM/ESA 2.4.0. It contains the following major sections:

- “CP Changes”
- “CMS Changes” on page 583
- “REXX/VM Changes” on page 595
- “GCS Changes” on page 596
- “VMSES/E Changes” on page 597

Use the information provided in this chapter to determine if you need to make any changes in the way you use these functions. See “Terminology and Symbols” on page 3 for the meaning of compatibility terms used in the tables.

CP Changes

This section identifies the CP externals that have changed since VM/ESA 2.1.0. It contains the following subsections:

- “CP Commands”
- “SYSTEM CONFIG Statements” on page 580
- “User Directory Control Statements” on page 580
- “CP Utilities” on page 581
- “CP DIAGNOSE Codes” on page 581
- “CP Macros” on page 582
- “CP Messages” on page 582

CP Commands

Table 121 lists the CP commands that have changed since VM/ESA 2.1.0. Refer to the *VM/ESA: CP Command and Utility Reference* for complete descriptions of CP commands.

Table 121 (Page 1 of 6). CP Commands Changed since VM/ESA 2.1.0

Command	Changes
CPLISTFILE	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses include 4-digit years for FULLDATE and ISODATE operands.
DEFINE (in general)	**Upwardly compatible changes:** • Supports new operand: MSGPROC. • Additional messages: HCP045E, HCP260E, HCP1014E, HCP2800E, HCP2801E, HCP2802E, HCP2803E, HCP2804I, HCP2806E, HCP2811E. • See DEFINE commands below.
DEFINE CHPID / PATH	**Upwardly compatible changes:** • New operands: INTEGRATED_SYSTEM_DEVICE, ISD, CLUSTer_bus_sender_channel, CBS, FICON, FC, FICON_CONVerter, FCV, OSA_Direct_express, OSD, OSA_Express, OSE. • New return codes for message HCP6806E.

Table 121 (Page 2 of 6). CP Commands Changed since VM/ESA 2.1.0

Command	Changes
DEFINE CRYPTO	**Upwardly compatible changes:** • New response. • New message: HCP1716E.
DEFINE CU / CNTLUNIT	**Upwardly compatible changes:** • The 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. • The range of permitted values for CU_LOGICAL_ADDRESS has been increased.
DEFINE DEVICE / IODEVICE	**Upwardly compatible changes:** • The 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.
DELETE CHPID / PATH	**Upwardly compatible changes:** • New return codes for message HCP6806E.
DELETE CU / CNTLUNIT	**Upwardly compatible changes:** • The 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.
DELETE DEVICE / IODEVICE	**Upwardly compatible changes:** • The 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.
DETACH (in general)	**Upwardly compatible changes:** • Supports new operand: MSGPROC. • Additional messages: HCP260E, HCP2805E, HCP2807E.
DISPLAY (in general)	**Upwardly compatible changes:** • See DISPLAY commands below.
DISPLAY Linkage Stack	**Upwardly compatible changes:** • Response indicates the called-space identification (CSID) if the linkage-stack entry type is a program-call state entry with a called-space ID.
DISPLAY Registers	**Upwardly compatible changes:** • Supports new operand: FPC. • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New response when FPC operand is used. • New messages: HCP6153E, HCP6154E.
DUMP Linkage Stack	**Upwardly compatible changes:** • Response indicates the called-space identification (CSID) if the linkage-stack entry type is a program-call state entry with a called-space ID.

Table 121 (Page 3 of 6). CP Commands Changed since VM/ESA 2.1.0

Command	Changes
IPL	**Incompatible changes:** • A new response is issued if tracing is active when a preferred guest is IPLed. Tracing must be turned off and the guest reIPLed. • In VM/ESA 2.1.0, when you IPL CMS with the PARM operand, CMS initialization attaches a fence of 8 bytes of X'FF's to the end of the PARM data before passing it to the SYSPROF EXEC. In VM/ESA 2.4.0, 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.
LOGON or LOGIN	**Upwardly compatible changes:** • New message: HCP2808E.
MODIFY CHPID / PATH	**Upwardly compatible changes:** • New return codes for message HCP6806E.
MODIFY CU / CNTLUNIT	**Upwardly compatible changes:** • New operands: TYPE CF. • The range of permitted values for CU_LOGICAL_ADDRESS has been increased.
QUERY (in general)	**Incompatible changes:** • See QUERY commands below. **Upwardly compatible changes:** • See QUERY commands below.
QUERY CACHEFW	**Upwardly compatible changes:** • Response indicates if the cache fast write function is suspended for the subsystem.
QUERY CHPID	**Upwardly compatible changes:** • New operand: TYPE. • New responses if TYPE is specified.
QUERY CPLEVEL	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • In the response, the release level value has changed. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q CPLEVEL command, the output from Q 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.
QUERY CRYPTO	**Upwardly compatible changes:** • Supports new operand: CAMQS. • Two new responses.
QUERY DASDFW	**Upwardly compatible changes:** • Response indicates if the DASD fast write function is suspended for the subsystem.
QUERY EXITS	**Upwardly compatible changes:** • Additional response information provided for a dynamic CP exit.

Table 121 (Page 4 of 6). CP Commands Changed since VM/ESA 2.1.0

Command	Changes
QUERY IMG	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q IMG command, the output from Q 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.
QUERY LDEVS	**Upwardly compatible changes:** • The response may include the IP address for TCP/IP.
QUERY NLS	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q NLS command, the output from Q 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.
QUERY NSS	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q NSS command, the output from Q 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.
QUERY READER / PRINTER / PUNCH	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses include 4-digit years for FULLDATE and ISODATE operands. • 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. • 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 Q RDR|PRT|PUN command, the output from Q 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).
QUERY TIME	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q TIME command, the output from Q TIME uses the new default date format. This causes the date to be expanded to include the 4-digit year.
QUERY TRFILES	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q TRFILES command, the output from Q 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.

Table 121 (Page 5 of 6). CP Commands Changed since VM/ESA 2.1.0

Command	Changes
QUERY UCR	**Upwardly compatible changes:** • New operands to specify date format: SHORTDATE, FULLDATE, ISODATE. • Response includes 4-digit years for FULLDATE and ISODATE operands. • 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 Q UCR command, the output from Q 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.
QUERY VIRTUAL ALL	**Upwardly compatible changes:** • Supports virtual message processors.
QUERY VIRTUAL CONSOLE	**Incompatible changes:** • The response may include a new line containing TCP/IP information.
QUERY VIRTUAL CRYPTO	**Upwardly compatible changes:** • New response.
QUERY (Virtual Device)	**Upwardly compatible changes:** • Response indicates REAL-MPLF or SIMULATED-MPLF if enabled for the device. • Supports virtual message devices. The device type MSGD appears in the response.
QUERY VIRTUAL OSA	**Upwardly compatible changes:** • New lines in the response for OSA devices that use the Queued-Direct-I/O (QDIO) Facility.
SET (in general)	**Upwardly compatible changes:** • See SET commands below.
SET CPTRACE	**Upwardly compatible changes:** • New trace category and trace codes for QDIO instructions.
SET CRYPTO	**Upwardly compatible changes:** • New operands: MODIFY, ON, OFF. • Changed responses. • Changed messages: HCP1706I, HCP1709E, HCP1710E, HCP1711I. • New messages: HCP1714E, HCP1715E.
SPTAPE	**Upwardly compatible changes:** • The first two digits of the 4-digit year are included in hexadecimal format in the SFBLOK dumped to tape.
STORE (Registers)	**Upwardly compatible changes:** • Supports new operand: FPC *hexword*. • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New messages: HCP6153E, HCP6154E.
STORE STATUS	**Upwardly compatible changes:** • 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.
TRACE (in general)	**Incompatible changes:** • New message: HCP1038E. **Upwardly compatible changes:** • New command responses. • See TRACE commands below.

Table 121 (Page 6 of 6). CP Commands Changed since VM/ESA 2.1.0

Command	Changes
TRACE *mnemonic1*	**Upwardly compatible changes:** • Supports new mnemonics: BSA.
TRACE *mnemonic2*	**Upwardly compatible changes:** • New mnemonic: SIGA.

SYSTEM CONFIG Statements

Table 122 lists SYSTEM CONFIG file statements that have changed since VM/ESA 2.1.0. See *VM/ESA: Planning and Administration* for complete descriptions of SYSTEM CONFIG statements.

Table 122. SYSTEM CONFIG Statements Changed since VM/ESA 2.1.0

Statement	Changes
CHARACTER_DEFAULTS	**Incompatible changes:** • Letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be defined as the system default line edit symbols (line-delete, character-delete, escape, line-end, and tab).

User Directory Control Statements

Table 123 lists the user directory control statements that have changed since VM/ESA 2.1.0. See the *VM/ESA: Planning and Administration* book for complete descriptions of user directory control statements.

Note: You can use the VMUDQ macro to query the CP user directory. See the *VM/ESA: CP Programming Services* book for more information on this macro.

Table 123 (Page 1 of 2). User Directory Control Statements Changed since VM/ESA 2.1.0

Statement	Changes
CRYPTO	**Upwardly compatible changes:** • New operand: MODIFY.
DATEFORMAT	**New** Specifies a user's default date format for commands that provide multiple date formats.
OPTION	**Upwardly compatible changes:** • 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. • Specifying the TODENABLE operand allows a user to change the virtual machine TOD clock with the new CP SET VTOD command. • Supports new operands: CFVM, CFUSER, DIAG88.
SPECIAL	**Upwardly compatible changes:** • Supports new operand: MSGPROC.

Table 123 (Page 2 of 2). User Directory Control Statements Changed since VM/ESA 2.1.0

Statement	Changes
USER	**Incompatible changes:** • Letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be defined as logical line edit symbols (line-end, line-delete, character-delete, and escape).

CP Utilities

Table 124 lists the utility programs that have changed since VM/ESA 2.1.0. See the *VM/ESA: CP Command and Utility Reference* for details on utility programs.

Table 124. Utility Programs Changed since VM/ESA 2.1.0

Utility	Changes
DIRECTXA	**Incompatible changes:** • 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'0F' (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. **Upwardly compatible changes:** • In the response, the release level value has changed.
SALIPL	**Incompatible changes:** • Message HCP039E deleted, replaced by new message HCP394E with same text. • Date field on the file list panel displays the year with 4 digits.

CP DIAGNOSE Codes

Table 125 lists the DIAGNOSE codes that have changed since VM/ESA 2.1.0. See the *VM/ESA: CP Programming Services* book for complete information on the DIAGNOSE codes discussed in this section.

Table 125 (Page 1 of 2). DIAGNOSE Codes Changed since VM/ESA 2.1.0

Code	Changes
X'00'	Storage extended identification code **Upwardly compatible changes:** • The value in the program product bit map has changed to indicate the new release level. Also, Bit 13 (X'0004000000000000') indicates whether Year 2000 support is present in CP.
X'14'	Input spool file manipulation **Upwardly compatible changes:** • For subcodes X'0004', X'0008', X'0FFE', and X'0FFF', a one-byte century indicator was added to the SFBLOK data area.
X'68'	Virtual Machine Communication Facility (VMCF) **Upwardly compatible changes:** • Supports a new function: SETLIMIT (Subcode X'000C').

Table 125 (Page 2 of 2). DIAGNOSE Codes Changed since VM/ESA 2.1.0

Code	Changes
X'7C'	Logical Device Support Facility **Upwardly compatible changes:** • 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.
X'84'	Directory Update-in-Place **Incompatible changes:** • For EDITCHAR operation, letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be specified as logical line edit symbols (line-end, line-delete, character-delete, and escape). **Upwardly compatible changes:** • Can now replace the following new information: – User's default date format setting • New operation: DATEFMT. • New return codes due to new function (in hex): 122, 123, 124.
X'BC'	Open and Query Spool File Characteristics **Upwardly compatible changes:** • 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.
X'D8'	Read spool file blocks on the system spool file queues **Upwardly compatible changes:** • For subcode X'0000', a one-byte century indicator was added to the SFBLOK data area.

CP Macros

Table 126 lists the CP macros that have changed since VM/ESA 2.1.0.

Table 126. CP Macros Changed since VM/ESA 2.1.0

Macro	Changes
HCPTKDEF	**Upwardly compatible changes:** • Supports new conversion type: INSTRUCT.

CP Messages

The following CP messages do not exist in VM/ESA 2.4.0:

HCP543A
HCP1161I
HCP1365E
HCP8039S
HCP8611T

The following CP messages have changed since VM/ESA 2.1.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CP messages.

HCP1003E
HCP1706I
HCP1709E
HCP1710E
HCP1711I
HCP2234E
HCP2252E
HCP2768E
HCP6111I
HCP6706E
HCP6789E

CMS Changes

This section identifies the CMS externals that have changed since VM/ESA 2.1.0. It contains the following subsections:

- “CMS Commands”
- “CMS File Pool Administration and Operator Commands” on page 587
- “CMS Macros” on page 587
- “Preferred Routines (CSL Routines)” on page 589
- “Compatibility Routines” on page 594
- “CMS Pipelines” on page 594
- “CMS Messages” on page 595

CMS Commands

Table 127 lists the CMS commands that have changed since VM/ESA 2.1.0. See the *VM/ESA: CMS Command Reference* for complete descriptions of CMS commands.

Table 127 (Page 1 of 5). CMS Commands Changed since VM/ESA 2.1.0

Command	Changes
CMSDESK	**Incompatible changes:** • Message DMS2302E replaced by new format of DMS622E, same return code. **Upwardly compatible changes:** • Supports three date formats: short date, full date, ISO date. • The display of the contents of the File Manager application has been changed. • Displays a toolbar on the applications. • New and changed menu items for some applications.
COPYFILE	**Incompatible changes:** • New message: DMS516E.
CREATE DIRECTORY	**Upwardly compatible changes:** • New message for authorization failure from ESM: DMS1331E.
CSLLIST	**Upwardly compatible changes:** • 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.
CSLMAP	**Upwardly compatible changes:** • 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.
DEFAULTS	**Upwardly compatible changes:** • New options are supported as parameters for FILELIST, NETDATA, and RDRLIST: VMDATE, SHORTDATE, FULLDATE, ISODATE.
DIRLIST	**Upwardly compatible changes:** • 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.

Table 127 (Page 2 of 5). CMS Commands Changed since VM/ESA 2.1.0

Command	Changes
ERASE	**Upwardly compatible changes:** • New message for authorization failure from ESM: DMS1332E.
FILEDEF	**Upwardly compatible changes:** • New option: LIBSRV. • Allows LRECL definitions up to 65535 bytes for OS variable spanned records (under XLRI processing) and non-OS CMS files. • Allows BLKSIZE definitions up to 65535 bytes for non-OS CMS files.
FILELIST	**Incompatible changes:** • 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 LISTFILE profiles. See Appendix A of the *VM/ESA: CMS Command Reference*. • If you file the file created by FILELIST, that file might contain new and changed fields (on the far right). **Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • The BEFORE *date* and AFTER *date* options support 4-digit years. • Screens and responses support 4-digit years. • If a date format option is not specified on the FILELIST command, the CMS DEFAULTS date format setting for FILELIST will be used. • 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.
GLOBAL	**Upwardly compatible changes:** • New message for duplicate library name in input list: DMS045W. The duplicates are ignored.
IDENTIFY	**Upwardly compatible changes:** • New options to specify date format: SHORTDATE, FULLDATE, ISODATE. • Responses support 4-digit years. • 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. • New option for TCP/IP mail integration: TCPIP.
LISTDS	**Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Responses support 4-digit years. • 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.

Table 127 (Page 3 of 5). CMS Commands Changed since VM/ESA 2.1.0

Command	Changes
LISTFILE	**Incompatible changes:** • Message DMS550E changed to DMS765E. **Upwardly compatible changes:** • New options to specify date format: SHORTDATE, FULLDATE, ISODATE. • The BEFORE *date* and AFTER *date* options support 4-digit years. • Responses support 4-digit years. • 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.
LKED	**Upwardly compatible changes:** • The defaults for *value1* and *value2* on the SIZE option have been increased to 400K and 100K, respectively.
LOAD	**Upwardly compatible changes:** • New message for insufficient storage above 16MB: DMS891W.
LOADMOD	**Incompatible changes:** • Changed message (new text possible): DMS639E.
MACLIST	**Upwardly compatible changes:** • 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.
MOVEFILE	**Upwardly compatible changes:** • Can process OS variable spanned records (under XLRI processing) and non-OS CMS files with record lengths up to 65535 bytes. • Adjusts output file sizes for compatibility between CMS and OS • Allows greater FILEDEF default flexibility for file attributes (RECFM, LRECL, BLKSIZE) • If default size values are used, fixes record truncation problems when moving data files from fixed to variable format • New message: DMS1116E.
NETDATA	**Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file. • Responses support 4-digit years. • 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.
NOTE	**Upwardly compatible changes:** • In the Date field of the note header, the year is now displayed with four digits. • TCP/IP domain names accepted as user IDs or as the resolution of nicknames. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
NUCXLOAD	**Incompatible changes:** • Changed message (new text possible): DMS639E.
OPENVM RUN	**Incompatible changes:** • Changed message (new text possible): DMS639E.

Table 127 (Page 4 of 5). CMS Commands Changed since VM/ESA 2.1.0

Command	Changes
PEEK	**Upwardly compatible changes:** • TCP/IP origin domain name address used when available and shown on PEEK message line for origin within current space and formatting limitations.
PIPE	**Incompatible changes:** • All CMS Pipelines messages have a new prefix, and many messages have new numbers and text. See "CMS Pipelines" on page 594.
QUERY CMSLEVEL	**Upwardly compatible changes:** • In the response, the CMS level value has changed.
QUERY CMSREL	**Upwardly compatible changes:** • In the response, the release level value has changed.
QUERY FILEDEF	**Upwardly compatible changes:** • Supports a new optional operand, ATTRIBUT, and its response.
RDRLIST	**Incompatible changes:** • If you file the file created by RDRLIST, that file might contain new and changed fields (on the far right). **Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Screen supports 4-digit years. • If a date format option is not specified on the RDRLIST command, the CMS DEFAULTS date format setting for RDRLIST will be used. • 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. • TCP/IP origin domain name address used when available and shown on the RDRLIST panel origin area within current space and formatting limitations.
RECEIVE	**Upwardly compatible changes:** • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
SENDFILE	**Upwardly compatible changes:** • TCP/IP domain names accepted as user IDs or as the resolution of nicknames. • New options to specify the transmission method: SMTP, MIME, UFTSYNC, UFTASYNC. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
TAPE	**Upwardly compatible changes:** • 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. • Added message for Tape Library Dataserver support: DMS2147W.
TELL	**Upwardly compatible changes:** • Accepts a TCP/IP domain name as part of the destination information.
VMFPLC2	**Upwardly compatible changes:** • 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. • Added message for Tape Library Dataserver support: DMS2147W.

Table 127 (Page 5 of 5). CMS Commands Changed since VM/ESA 2.1.0

Command	Changes
VMLINK	**Upwardly compatible changes:** • 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.

CMS File Pool Administration and Operator Commands

Table 128 lists the CMS file pool administration and operator commands that have changed since VM/ESA 2.1.0. See the *VM/ESA: CMS File Pool Planning, Administration, and Operation* book for complete descriptions of file pool administration and operator commands.

Table 128. CMS File Pool Administration and Operator Commands Changed since VM/ESA 2.1.0

Command	Changes
AUDIT	**Upwardly compatible changes:** • New operands: *fn ft*, REPLACE. • Added messages: DMS024E, DMS1258E, DMS3253I, DMS3254E, DMS3255E. • Changed message: DMS3470W (new text possible).
DELETE USER	**Upwardly compatible changes:** • New options: DELAUTH | KEEPAUTH. • Added message: DMS2023E.
FILEPOOL RELOAD	**Upwardly compatible changes:** • New message: DMS3455I.
FILEPOOL UNLOAD	**Upwardly compatible changes:** • New message: DMS3455I.

CMS Macros

Table 129 lists the CMS preferred macros that have changed since VM/ESA 2.1.0. Table 130 lists the CMS compatibility macros that have changed. Table 131 lists the OS simulation macros that have changed. Table 132 lists the simulated OS/MVS supervisor calls that have changed.

Preferred Macros

See the *VM/ESA: CMS Application Development Reference for Assembler* for complete descriptions of these macros.

Table 129 (Page 1 of 2). CMS Preferred Macros Changed since VM/ESA 2.1.0

Macro	Changes
CMSLEVEL	**Incompatible changes:** • The CMS level has been frozen at X'0F' for CMS Level 12 (VM/ESA Version 2 Release 1.0) and later. Use the new DMSQEFL macro or the DMSQEFL CSL routine instead.

Table 129 (Page 2 of 2). CMS Preferred Macros Changed since VM/ESA 2.1.0

Macro	Changes
DIRBUFF	**Upwardly compatible changes:** • The FILE record contains the following new fields: DIRFDAXD, DIRFDAXC, DIRFDAXI, DIRFLV13. • The FILEEXT record contains the following new fields: DIREDAXD, DIREDAXC, DIREDAXI, DIREDRXD, DIREDRXC, DIREDRXI, DIRECDXD, DIRECDXC, DIRECDXI, DIREDCXD, DIREDCXC, DIREDCXI, DIRELV13. • The SEARCHALL and SEARCHAUTH records contain the following new fields: DIRSDAXD, DIRSDAXC, DIRSDAXI, DIRSCEND, DIRSLV13.
EXSBUFF	**Upwardly compatible changes:** • The FILE record contains the following new fields: EXSFDAXD, EXSFDAXC, EXSFDAXI, EXSFDRXD, EXSFDRXC, EXSFDRXI, EXSFCDXD, EXSFCDXC, EXSFCDXI, EXSFDCXD, EXSFDCXC, EXSFDCXI, EXSF2000, EXSFLV13. • In the FILE record, the following field has changed: EXSFRES. • The DIR record contains the following new fields: EXSDDCXD, EXSDDCXC, EXSDDCXI, EXSDCDXD, EXSDCDXC, EXSDCDXI, EXSD2000, EXSDLV13. • In the DIR record, the following field has changed: EXSDRES.
FSSTATE	**Upwardly compatible changes:** • 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*nn*, 1=20*nn*, where *nn* is the 2-digit year). In VM/ESA 2.1.0, this bit was not used.
FSTD	**Upwardly compatible changes:** • The FSTFLAGS section contains the new FSTCNTRY field, which is a bit that indicates the century (first two digits of the year) the file was last written or updated (0=19*nn*, 1=20*nn*, where *nn* is the 2-digit year).
TAPECTL	**Upwardly compatible changes:** • 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.

Compatibility Macros

See *VM/ESA: Planning and Administration* for information about the DEFNUC macro.

Table 130. CMS Compatibility Macros Changed since VM/ESA 2.1.0

Macro	Changes
DEFNUC	**Incompatible changes:** • 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*). **Upwardly compatible changes:** • The default IPL heading has been updated with the new release level.

OS Simulation Macros and Supervisor Calls

See the *VM/ESA: CMS Application Development Guide for Assembler* for information about the OS macros and supervisor calls that CMS simulates.

Table 131. OS Simulation Macros Changed since VM/ESA 2.1.0

Macro	Changes
CLOSE	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
DCB	**Upwardly compatible changes:** • 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.
GET	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.
OPEN	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
PUT	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.
TIME	**Upwardly compatible changes:** • The 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.

Table 132. OS Simulation Supervisor Calls Changed since VM/ESA 2.1.0

SVC	Changes
SVC 19 (OPEN)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
SVC 20 (CLOSE)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.

Preferred Routines (CSL Routines)

Table 133 lists the CSL routines that have changed since VM/ESA 2.1.0. See the *VM/ESA: CMS Application Development Reference* for complete descriptions of the routines in this table.

Table 133 (Page 1 of 5). CSL Routines Changed since VM/ESA 2.1.0

Routine	Changes
DMSCLBLK	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330, 90492, 90495.
DMSCLDBK	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330, 90495.
DMSCLOSE	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90320, 90330, 90495.
DMSCRDIR	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSCRFIL	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* and *create_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSCROB	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* and *create_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSDEUSR	**Upwardly compatible changes:** • New parameters: DELAUTH, KEEPAUTH, and *length4*. • New reason code: 98700.
DMSENUSR	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE, *length7*. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90330, 90495.
DMSERP	**Upwardly compatible changes:** • New information names for Year 2000 support (FILE_DATE_CENTURY, ACT_FILE_DATE_CENTRY (note that U is omitted), and YEAR2000_SUPPORT).

Table 133 (Page 2 of 5). CSL Routines Changed since VM/ESA 2.1.0

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSEXIDI</td><td><b>Incompatible changes:</b>
• When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 2.1.0, it was returned as 8 characters.
<b>Upwardly compatible changes:</b>
• New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.
• The <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.
• New reason codes: 90320, 90330.</td></tr>
<tr><td>DMSEXIFI</td><td><b>Incompatible changes:</b>
• When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 2.1.0, it was returned as 8 characters.
<b>Upwardly compatible changes:</b>
• New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE.
• The <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.
• New reason codes: 90320, 90330.</td></tr>
<tr><td>DMSEXIST</td><td><b>Upwardly compatible changes:</b>
• The following offsets have changed in the FILE data record (and the record length has increased to 436 bytes):
<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>339 (X'153')</td><td><i>dec_date_ext</i></td><td>Reserved in VM/ESA 2.1.0</td></tr>
<tr><td>343 (X'157')</td><td><i>date_ext</i></td><td>Added</td></tr>
<tr><td>353 (X'161')</td><td><i>iso_date_ext</i></td><td>Added</td></tr>
<tr><td>363 (X'16B')</td><td><i>dec_dateref_ext</i></td><td>Added</td></tr>
<tr><td>367 (X'16F')</td><td><i>dateref_ext</i></td><td>Added</td></tr>
<tr><td>377 (X'179')</td><td><i>iso_dateref_ext</i></td><td>Added</td></tr>
<tr><td>387 (X'183')</td><td><i>dec_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>391 (X'187')</td><td><i>cr_date_ext</i></td><td>Added</td></tr>
<tr><td>401 (X'191')</td><td><i>iso_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>411 (X'19B')</td><td><i>dec_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>415 (X'19F')</td><td><i>last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>425 (X'1A9')</td><td><i>iso_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>435 (X'1B3')</td><td>Reserved</td><td>Added</td></tr>
</table>
• The following offsets have changed in the DIRECTORY data record (and the record length has increased to 308 bytes):
<table>
<tr><th>OFFSET</th><th>Field Name</th><th>Description of Change</th></tr>
<tr><td>254 (X'FE')</td><td><i>dec_last_change_date_ext</i></td><td>Reserved in VM/ESA 2.1.0</td></tr>
<tr><td>258 (X'102')</td><td><i>last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>268 (X'10C')</td><td><i>iso_last_change_date_ext</i></td><td>Added</td></tr>
<tr><td>278 (X'116')</td><td><i>dec_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>282 (X'11A')</td><td><i>cr_date_ext</i></td><td>Added</td></tr>
<tr><td>292 (X'124')</td><td><i>iso_cr_date_ext</i></td><td>Added</td></tr>
<tr><td>302 (X'12E')</td><td>Reserved</td><td>Added</td></tr>
</table></td></tr>
</table>

Table 133 (Page 3 of 5). CSL Routines Changed since VM/ESA 2.1.0

<table>
<tr><th>Routine</th><th>Changes</th></tr>
<tr><td>DMSGETDA</td><td>

Incompatible changes:

- When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 2.1.0, it was returned as 8 characters.

Upwardly compatible changes:

- New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*.
- The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.
- New reason codes: 90310, 90320, 90330.

</td></tr>
<tr><td>DMSGETDF</td><td>

Incompatible changes:

- When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 2.1.0, it was returned as 8 characters.

Upwardly compatible changes:

- New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*.
- The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters.
- New reason codes: 90310, 90320, 90330.

</td></tr>
<tr><td>DMSGETDI</td><td>

Upwardly compatible changes:

- The following offsets have changed in the FILE data record (and the record length has increased to 112 bytes):

OFFSET	Field Name	Description of Change
88 (X'58')	*dec_date_ext*	Added
92 (X'5C')	*date_ext*	Added
102 (X'66')	*iso_date_ext*	Added

- The following offsets have changed in the FILEEXT data record (and the record length has increased to 284 bytes):

OFFSET	Field Name	Description of Change
185 (X'B9')	*dec_date_ext*	Reserved in VM/ESA 2.1.0
189 (X'BD')	*date_ext*	Added
199 (X'C7')	*iso_date_ext*	Added
209 (X'D1')	*dec_dateref_ext*	Added
213 (X'D5')	*dateref_ext*	Added
223 (X'DF')	*iso_dateref_ext*	Added
233 (X'E9')	*dec_cr_date_ext*	Added
237 (X'ED')	*cr_date_ext*	Added
247 (X'F7')	*iso_cr_date_ext*	Added
257 (X'101')	*dec_last_change_date_ext*	Added
261 (X'105')	*last_change_date_ext*	Added
271 (X'10F')	*iso_last_change_date_ext*	Added
281 (X'119')	Reserved	Added

- The following offsets have changed in the SEARCHALL and SEARCHAUTH data records (and the record length has increased to 252 bytes):

OFFSET	Field Name	Description of Change
226 (X'E2')	Reserved	Added
228 (X'E4')	*dec_date_ext*	Added
232 (X'E8')	*date_ext*	Added
242 (X'F2')	*iso_date_ext*	Added

</td></tr>
</table>

Table 133 (Page 4 of 5). CSL Routines Changed since VM/ESA 2.1.0

Routine	Changes
DMSGETDS	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 2.1.0, it was returned as 8 characters. **Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330.
DMSGETDX	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 2.1.0, it was returned as 8 characters. **Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE and *length2*. • The *date*, *dateref*, *create_date*, and *update_date* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason codes: 90310, 90320, 90330.
DMSOPBLK	**Incompatible changes:** • When this routine is called from a REXX program, the date field is returned as 10 characters. In VM/ESA 2.1.0, it was returned as 8 characters. **Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date*, *create_date*, and *dateref* parameters support 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSOPDBK	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *create_date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSOPEN	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *create_date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.
DMSQEFL	**Upwardly compatible changes:** • Supports new values for *cp_level* and *cms_level*
DMSQSFSL	**Upwardly compatible changes:** • Supports new value for *server_level*

Table 133 (Page 5 of 5). CSL Routines Changed since VM/ESA 2.1.0

Routine	Changes
DMSRDCAT	**Upwardly compatible changes:** • OBJECTCAT record for SFS: – New bit settings in FILEFLAGS field to support 4-digit years (century setting for DATE and DATEREF fields). – The reserved CHAR(1) field preceding the LAST_CHANGE_DATE field has changed to: **Field Name** — **Field Type/Description** CHGDATE_CENTURY — CHAR(1) Century byte for LAST_CHANGE_DATE – The reserved CHAR(1) field preceding the CREATIONDATE field has changed to: **Field Name** — **Field Type/Description** CREATIONDATE_CENTURY — CHAR(1) Century byte for CREATIONDATE
DMSTRUNC	**Upwardly compatible changes:** • New parameters to specify date format: SHORTDATE, FULLDATE, ISODATE. • The *date* parameter supports 4-digit years (10-character dates) when used with the FULLDATE and ISODATE parameters. • New reason code: 90495.

Compatibility Routines

Table 134 lists the CMS compatibility routines that have changed between VM/ESA 2.1.0 and VM/ESA 2.4.0. See the *VM/ESA: CMS Application Development Guide for Assembler* for more information about the routines in this table.

Table 134. CMS Compatibility Routines Changed since VM/ESA 2.1.0

CMS Routine	Explanation
DMSTVS	**Upwardly compatible changes:** • Supports new LIBSRV plist parameter. • 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.

CMS Pipelines

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.

- 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.

  Note: User-written applications that are sensitive to these changes may require alterations. Published externals (such as PIPGFTXT) have **not** been changed.

- 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. Some specific enhancements that have occurred since VM/ESA 2.1.0 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 *VM/ESA: CMS Pipelines Reference* can be found in the *CMS/TSO Pipelines: Author's Edition*, which is included with the VM/ESA 2.4.0 library.

CMS Messages

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 H, "CMS Pipelines Message Cross-Reference [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]" on page 669.

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/ESA.

The following CMS message does not exist in VM/ESA 2.4.0:

DMS683E

The following CMS messages have changed since VM/ESA 2.1.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CMS messages.

DMS516E
DMS622E
DMS639E
DMS1229E
DMS2010E
DMS3009R
DMS3455I
DMS3470W

REXX/VM Changes

This section identifies the REXX/VM externals that have changed since VM/ESA 2.1.0. It contains the following subsections:

- "REXX/VM Functions"
- "External Functions" on page 596

REXX/VM Functions

Table 135 lists the REXX/VM functions that have changed since VM/ESA 2.1.0. For complete descriptions of these functions, see the *VM/ESA: REXX/VM Reference*.

Table 135. REXX/VM Functions Changed since VM/ESA 2.1.0

Function	Changes
DATE	**Upwardly compatible changes:** • New parameters allow you to specify a date to be converted to a different format. • New parameters: *output_separator_char*, *input_separator_char*.

External Functions

Table 136 lists external functions which can be used by REXX/VM that have changed since VM/ESA 2.1.0. For complete descriptions of these functions, see the *VM/ESA: REXX/VM Reference*.

Table 136. External Functions Changed since VM/ESA 2.1.0

Function	Changes
CMSFLAG	**Upwardly compatible changes:** • New value for *flag*: YEAR2000.
DIAG DIAGRC	**Upwardly compatible changes:** • New DIAGNOSE code is supported: X'270'.

GCS Changes

This section identifies the GCS externals that have changed since VM/ESA 2.1.0. It contains the following subsection:

- "GCS Commands and Macros"

GCS Commands and Macros

Table 137 lists the GCS commands and macros that have changed since VM/ESA 2.1.0. See *VM/ESA: Group Control System* for complete descriptions of all GCS commands and macros.

Table 137. GCS Commands and Macros Changed since VM/ESA 2.1.0

Command/Macro	Changes
GCSLEVEL macro	**Upwardly compatible changes:** • Additional equates for new release levels.
GETMAIN macro	**Incompatible changes:** • In VM/ESA 2.1.0, although LOC=RES was documented as the default, the actual default was LOC=BELOW, and all the requested virtual storage was allocated below 16MB. In VM/ESA 2.2.0, the actual default was 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.
QUERY GCSLEVEL command	**Upwardly compatible changes:** • Response contains new values.
QUERY MODDATE command	**Incompatible changes:** • A full 4-digit year is now returned in the date field of the response instead of a 2-digit year.
TIME macro	**Upwardly compatible changes:** • The 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.

VMSES/E Changes

This section identifies the VMSES/E externals that have changed since VM/ESA 2.1.0. It contains the following subsection:

- “VMSES/E Messages”

Also see “Summary of Service Exec Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]” on page 250.

VMSES/E Messages

The following VMSES/E messages have changed since VM/ESA 2.1.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of VMSES/E messages.

VMF1078I
VMF1821E
VMF2066E
VMF2114R
VMF2118I
VMF2119I
VMF2120W
VMF2152E
VMF2206E
VMF2206W
VMF2225E
VMF2507I
VMF2509I
VMF2760I
VMF2767I

Chapter 22. Compatibility Tables for Converting from VM/ESA 2.2.0

This chapter identifies the VM/ESA externals that have changed between VM/ESA 2.2.0 and VM/ESA 2.4.0. It contains the following sections:

- "CP Changes"
- "CMS Changes" on page 604
- "REXX/VM Changes" on page 609
- "GCS Changes" on page 610
- "VMSES/E Changes" on page 610

Use the information provided in this chapter to determine if you need to make any changes in the way you use these functions. See "Terminology and Symbols" on page 3 for the meaning of compatibility terms used in the tables.

CP Changes

This section identifies the CP externals that have changed since VM/ESA 2.2.0. It contains the following subsections:

- "CP Commands"
- "SYSTEM CONFIG Statements" on page 602
- "User Directory Control Statements" on page 602
- "CP Utilities" on page 603
- "CP DIAGNOSE Codes" on page 603
- "CP Macros" on page 604
- "CP Messages" on page 604

CP Commands

Table 138 lists the CP commands that have changed since VM/ESA 2.2.0. Refer to the *VM/ESA: CP Command and Utility Reference* for complete descriptions of CP commands.

Table 138 (Page 1 of 4). CP Commands Changed since VM/ESA 2.2.0

Command	Changes
DEFINE (in general)	**Upwardly compatible changes:** • Supports new operand: MSGPROC. • Additional messages: HCP045E, HCP260E, HCP1014E, HCP2800E, HCP2801E, HCP2802E, HCP2803E, HCP2804I, HCP2806E, HCP2811E. • See DEFINE commands below.
DEFINE CHPID / PATH	**Upwardly compatible changes:** • Supports new operands: CLUSTer_bus_sender_channel, CBS, FICON, FC, FICON_CONVerter, FCV, OSA_Direct_express, OSD, OSA_Express, OSE. • New return codes for message HCP6806E.
DEFINE CRYPTO	**Upwardly compatible changes:** • New response. • New message: HCP1716E.

Table 138 (Page 2 of 4). CP Commands Changed since VM/ESA 2.2.0

Command	Changes
DEFINE CU / CNTLUNIT	**Upwardly compatible changes:** • The 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. • The range of permitted values for CU_LOGICAL_ADDRESS has been increased.
DEFINE DEVICE / IODEVICE	**Upwardly compatible changes:** • The 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.
DELETE CHPID / PATH	**Upwardly compatible changes:** • New return codes for message HCP6806E.
DELETE CU / CNTLUNIT	**Upwardly compatible changes:** • The 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.
DELETE DEVICE / IODEVICE	**Upwardly compatible changes:** • The 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.
DETACH (in general)	**Upwardly compatible changes:** • Supports new operand: MSGPROC. • Additional messages: HCP260E, HCP2805E, HCP2807E.
DISPLAY (in general)	**Upwardly compatible changes:** • See DISPLAY commands below.
DISPLAY Registers	**Upwardly compatible changes:** • Supports new operand: FPC. • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New response when FPC operand is used. • New messages: HCP6153E, HCP6154E.
IPL	**Incompatible changes:** • In VM/ESA 2.2.0, when you IPL CMS with the PARM operand, CMS initialization attaches a fence of 8 bytes of X'FF's to the end of the PARM data before passing it to the SYSPROF EXEC. In VM/ESA 2.4.0, 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.
LOGON or LOGIN	**Upwardly compatible changes:** • New message: HCP2808E.
MODIFY CHPID / PATH	**Upwardly compatible changes:** • New return codes for message HCP6806E.

Table 138 (Page 3 of 4). CP Commands Changed since VM/ESA 2.2.0

Command	Changes
MODIFY CU / CNTLUNIT	**Upwardly compatible changes:** • New operands: TYPE CF. • The range of permitted values for CU_LOGICAL_ADDRESS has been increased.
QUERY (in general)	**Incompatible changes:** • See QUERY commands below. **Upwardly compatible changes:** • See QUERY commands below.
QUERY CHPID / PATH	**Upwardly compatible changes:** • New operand: TYPE. • New responses if TYPE is specified.
QUERY CPLEVEL	**Upwardly compatible changes:** • In the response, the release level value has changed.
QUERY CRYPTO	**Upwardly compatible changes:** • Supports new operand: CAMQS. • Two new responses.
QUERY EXITS	**Upwardly compatible changes:** • Additional response information provided for a dynamic CP exit.
QUERY LDEVS	**Upwardly compatible changes:** • The response may include the IP address for TCP/IP.
QUERY VIRTUAL ALL	**Upwardly compatible changes:** • Supports virtual message processors.
QUERY VIRTUAL CONSOLE	**Incompatible changes:** • The response may include a new line containing TCP/IP information.
QUERY VIRTUAL CRYPTO	**Upwardly compatible changes:** • New response.
QUERY (Virtual Device)	**Upwardly compatible changes:** • Supports virtual message devices. The device type MSGD appears in the response.
QUERY VIRTUAL OSA	**Upwardly compatible changes:** • New lines in the response for OSA devices that use the Queued-Direct-I/O (QDIO) Facility.
SET (in general)	**Upwardly compatible changes:** • See SET commands below.
SET CPTRACE	**Upwardly compatible changes:** • New trace category and trace codes for QDIO instructions.
SET CRYPTO	**Upwardly compatible changes:** • New operands: MODIFY, ON, OFF. • Changed responses. • Changed messages: HCP1706I, HCP1709E, HCP1710E, HCP1711I. • New messages: HCP1714E, HCP1715E.

Table 138 (Page 4 of 4). CP Commands Changed since VM/ESA 2.2.0

Command	Changes
STORE (Registers)	**Upwardly compatible changes:** • Supports new operand: FPC *hexword*. • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New messages: HCP6153E, HCP6154E.
STORE STATUS	**Upwardly compatible changes:** • 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.
TRACE (in general)	**Upwardly compatible changes:** • New command responses. • See TRACE commands below.
TRACE *mnemonic1*	**Upwardly compatible changes:** • Supports new mnemonics: BSA.
TRACE *mnemonic2*	**Upwardly compatible changes:** • New mnemonic: SIGA.

SYSTEM CONFIG Statements

Table 139 lists SYSTEM CONFIG file statements that have changed since VM/ESA 2.2.0. See *VM/ESA: Planning and Administration* for complete descriptions of SYSTEM CONFIG statements.

Table 139. SYSTEM CONFIG Statements Changed since VM/ESA 2.2.0

Statement	Changes
CHARACTER_DEFAULTS	**Incompatible changes:** • Letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be defined as the system default line edit symbols (line-delete, character-delete, escape, line-end, and tab).

User Directory Control Statements

Table 140 lists the user directory control statements that have changed since VM/ESA 2.2.0. See the *VM/ESA: Planning and Administration* book for complete descriptions of user directory control statements.

Note: You can use the VMUDQ macro to query the CP user directory. See the *VM/ESA: CP Programming Services* book for more information on this macro.

Table 140 (Page 1 of 2). User Directory Control Statements Changed since VM/ESA 2.2.0

Statement	Changes
CRYPTO	**Upwardly compatible changes:** • New operand: MODIFY.

Table 140 (Page 2 of 2). User Directory Control Statements Changed since VM/ESA 2.2.0

Statement	Changes
OPTION	**Upwardly compatible changes:** • Specifying the TODENABLE operand allows a user to change the virtual machine TOD clock with the new CP SET VTOD command. • Supports new operands: CFVM, CFUSER, DIAG88.
SPECIAL	**Upwardly compatible changes:** • Supports new operand: MSGPROC.
USER	**Incompatible changes:** • Letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be defined as logical line edit symbols (line-end, line-delete, character-delete, and escape).

CP Utilities

Table 141 lists the utility programs that have changed since VM/ESA 2.2.0. See the *VM/ESA: CP Command and Utility Reference* for details on utility programs.

Table 141. Utility Programs Changed since VM/ESA 2.2.0

Utility	Changes
DIRECTXA	**Incompatible changes:** • 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'0F' (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. **Upwardly compatible changes:** • In the response, the release level value has changed.

CP DIAGNOSE Codes

Table 142 lists the DIAGNOSE codes that have changed since VM/ESA 2.2.0. See the *VM/ESA: CP Programming Services* book for complete information on the DIAGNOSE codes discussed in this section.

Table 142 (Page 1 of 2). DIAGNOSE Codes Changed since VM/ESA 2.2.0

Code	Changes
X'00'	Storage extended identification code **Upwardly compatible changes:** • The value in the program product bit map has changed to indicate the new release level.
X'68'	Virtual Machine Communication Facility (VMCF) **Upwardly compatible changes:** • Supports a new function: SETLIMIT (Subcode X'000C').
X'7C'	Logical Device Support Facility **Upwardly compatible changes:** • 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.

Table 142 (Page 2 of 2). DIAGNOSE Codes Changed since VM/ESA 2.2.0

Code	Changes
X'84'	Directory Update-in-Place **Incompatible changes:** • For EDITCHAR operation, letters A-Z, numbers 0-9, and bytes X'OE' (shift out) and X'0F' (shift in) cannot be specified as logical line edit symbols (line-end, line-delete, character-delete, and escape).
X'270'	Pseudo Timer Extended **Upwardly compatible changes:** • Output has been expanded to include the version of DIAGNOSE X'270', the user's default date format, and the system default date format.

CP Macros

Table 143 lists the CP macros that have changed since VM/ESA 2.2.0.

Table 143. CP Macros Changed since VM/ESA 2.2.0

Macro	Changes
HCPTKDEF	**Upwardly compatible changes:** • Supports new conversion type: INSTRUCT.

CP Messages

The following CP messages do not exist in VM/ESA 2.4.0:

HCP543A
HCP1365E
HCP8039S
HCP8611T

The following CP messages have changed since VM/ESA 2.2.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CP messages.

HCP580I
HCP1003E
HCP1016E
HCP1512E
HCP1706I
HCP1709E
HCP1710E
HCP1711I
HCP2234E
HCP2768E
HCP6111I
HCP6706E
HCP6789E

CMS Changes

This section identifies the CMS externals that have changed since VM/ESA 2.2.0. It contains the following subsections:

- "CMS Commands"
- "CMS Macros" on page 607
- "Preferred Routines (CSL Routines)" on page 608
- "CMS Pipelines" on page 608
- "CMS Messages" on page 609

CMS Commands

Table 144 lists the CMS commands that have changed since VM/ESA 2.2.0. See the *VM/ESA: CMS Command Reference* for complete descriptions of CMS commands.

Table 144 (Page 1 of 3). CMS Commands Changed since VM/ESA 2.2.0

Command	Changes
CSLLIST	**Upwardly compatible changes:** • 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.
CSLMAP	**Upwardly compatible changes:** • 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.
DEFAULTS	**Upwardly compatible changes:** • New options are supported as parameters for NETDATA: VMDATE, SHORTDATE, FULLDATE, ISODATE.
DIRLIST	**Upwardly compatible changes:** • 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.
FILEDEF	**Upwardly compatible changes:** • Allows LRECL definitions up to 65535 bytes for OS variable spanned records (under XLRI processing) and non-OS CMS files. • Allows BLKSIZE definitions up to 65535 bytes for non-OS CMS files.
FILELIST	**Upwardly compatible changes:** • 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.
IDENTIFY	**Upwardly compatible changes:** • New option for TCP/IP mail integration: TCPIP.
LISTDS	**Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Responses support 4-digit years. • 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.
LISTFILE	**Incompatible changes:** • Message DMS550E changed to DMS765E.
LKED	**Upwardly compatible changes:** • The defaults for *value1* and *value2* on the SIZE option have been increased to 400K and 100K, respectively.
MACLIST	**Upwardly compatible changes:** • 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.

Table 144 (Page 2 of 3). CMS Commands Changed since VM/ESA 2.2.0

Command	Changes
MOVEFILE	**Upwardly compatible changes:** • Can process OS variable spanned records (under XLRI processing) and non-OS CMS files with record lengths up to 65535 bytes. • Adjusts output file sizes for compatibility between CMS and OS • Allows greater FILEDEF default flexibility for file attributes (RECFM, LRECL, BLKSIZE) • If default size values are used, fixes record truncation problems when moving data files from fixed to variable format • New message: DMS1116E.
NETDATA	**Upwardly compatible changes:** • New options to specify date format: VMDATE, SHORTDATE, FULLDATE, ISODATE. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file. • Responses support 4-digit years. • 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.
NOTE	**Upwardly compatible changes:** • TCP/IP domain names accepted as user IDs or as the resolution of nicknames. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
PEEK	**Upwardly compatible changes:** • TCP/IP origin domain name address used when available and shown on PEEK message line for origin within current space and formatting limitations.
PIPE	**Incompatible changes:** • All CMS Pipelines messages have a new prefix, and many messages have new numbers and text. See “CMS Pipelines” on page 608.
QUERY CMSLEVEL	**Upwardly compatible changes:** • In the response, the CMS level value has changed.
QUERY CMSREL	**Upwardly compatible changes:** • In the response, the release level value has changed.
QUERY FILEDEF	**Upwardly compatible changes:** • Supports a new optional operand, ATTRIBUT, and its response.
RDRLIST	**Upwardly compatible changes:** • 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. • TCP/IP origin domain name address used when available and shown on RDRLIST panel origin area within current space and formatting limitations.
RECEIVE	**Upwardly compatible changes:** • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
SENDFILE	**Upwardly compatible changes:** • TCP/IP domain names accepted as user IDs or as the resolution of nicknames. • New options to specify the transmission method: SMTP, MIME, UFTSYNC, UFTASYNC. • Supports 4-digit-year date formats for entries in the *userid* NETLOG file.
TELL	**Upwardly compatible changes:** • Accepts a TCP/IP domain name as part of the destination information.

Table 144 (Page 3 of 3). CMS Commands Changed since VM/ESA 2.2.0

Command	Changes
VMLINK	**Upwardly compatible changes:** • 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.

CMS Macros

Table 145 lists the CMS compatibility macros that have changed. Table 146 lists the OS simulation macros that have changed. Table 147 lists the simulated OS/MVS supervisor calls that have changed.

Compatibility Macros

See *VM/ESA: Planning and Administration* for information about the DEFNUC macro.

Table 145. CMS Compatibility Macros Changed since VM/ESA 2.2.0

Macro	Changes
DEFNUC	**Incompatible changes:** • 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*). **Upwardly compatible changes:** • The default IPL heading has been updated with the new release level.

OS Simulation Macros and Supervisor Calls

See the *VM/ESA: CMS Application Development Guide for Assembler* for information about the OS macros and supervisor calls that CMS simulates.

Table 146. OS Simulation Macros Changed since VM/ESA 2.2.0

Macro	Changes
CLOSE	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
DCB	**Upwardly compatible changes:** • 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.
GET	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.
OPEN	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
PUT	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.

Table 147. OS Simulation Supervisor Calls Changed since VM/ESA 2.2.0

SVC	Changes
SVC 19 (OPEN)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
SVC 20 (CLOSE)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.

Preferred Routines (CSL Routines)

Table 148 lists the CSL routines that have changed since VM/ESA 2.2.0. See the *VM/ESA: CMS Application Development Reference* for complete descriptions of the routines in this table.

Table 148. CSL Routines Changed since VM/ESA 2.2.0

Routine	Changes
DateTimeSubtract	**Upwardly compatible changes:** • Supports new date and time formats.
DMSQEFL	**Upwardly compatible changes:** • Supports new values for *cp_level* and *cms_level*
DMSQSFSL	**Upwardly compatible changes:** • Supports new value for *server_level*

Compatibility Routines

Table 149 lists the CMS compatibility routines that have changed between VM/ESA 2.2.0 and VM/ESA 2.4.0. See the *VM/ESA: CMS Application Development Guide for Assembler* for more information about the routines in this table.

Table 149. CMS Compatibility Routines Changed since VM/ESA 2.2.0

CMS Routine	Explanation
DMSTVS	**Upwardly compatible changes:** • 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.

CMS Pipelines

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.
- 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.

Note: User-written applications that are sensitive to these changes may require alterations. Published externals (such as PIPGFTXT) have **not** been changed.

- 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. Some specific enhancements that have occurred since VM/ESA 2.1.0 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 *VM/ESA: CMS Pipelines Reference* can be found in the *CMS/TSO Pipelines: Author's Edition*, which is included with the VM/ESA 2.4.0 library.

CMS Messages

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 H, "CMS Pipelines Message Cross-Reference [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]" on page 669.

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/ESA.

The following CMS message does not exist in VM/ESA 2.4.0:

DMS683E

The following CMS messages have changed since VM/ESA 2.2.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CMS messages.

DMS1433I
DMS1437I
DMS2010E
DMS3009R

REXX/VM Changes

This section identifies the REXX/VM externals that have changed since VM/ESA 2.2.0. It contains the following subsections:

- "REXX/VM Functions"

REXX/VM Functions

Table 150 lists the REXX/VM functions that have changed since VM/ESA 2.2.0. For complete descriptions of these functions, see the *VM/ESA: REXX/VM Reference*.

Table 150. REXX/VM Functions Changed since VM/ESA 2.2.0

Function	Changes
DATE	**Upwardly compatible changes:** • New parameters: *output_separator_char*, *input_separator_char*.

GCS Changes

This section identifies the GCS externals that have changed since VM/ESA 2.2.0. It contains the following subsection:

- "GCS Commands and Macros"

GCS Commands and Macros

Table 151 lists the GCS commands and macros that have changed since VM/ESA 2.2.0. See *VM/ESA: Group Control System* for complete descriptions of all GCS commands and macros.

Table 151. GCS Commands and Macros Changed since VM/ESA 2.2.0

Command/Macro	Changes
GCSLEVEL macro	**Upwardly compatible changes:** • Additional equates for new release levels.
QUERY GCSLEVEL command	**Upwardly compatible changes:** • Response contains new values.

VMSES/E Changes

This section identifies the VMSES/E externals that have changed since VM/ESA 2.2.0. It contains the following subsection:

- "VMSES/E Messages"

Also see "Summary of Service Exec Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]" on page 250.

VMSES/E Messages

The following VMSES/E messages have changed since VM/ESA 2.2.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of VMSES/E messages.

VMF1078I
VMF2066E
VMF2114R
VMF2118I
VMF2119I
VMF2120W
VMF2152E
VMF2206W
VMF2507I
VMF2760I
VMF2767I

Chapter 23. Compatibility Tables for Converting from VM/ESA 2.3.0

This chapter identifies the VM/ESA externals that have changed between VM/ESA 2.3.0 and VM/ESA 2.4.0. It contains the following sections:

- "CP Changes"
- "CMS Changes" on page 614
- "REXX/VM Changes" on page 618
- "GCS Changes" on page 618
- "VMSES/E Changes" on page 619

Use the information provided in this chapter to determine if you need to make any changes in the way you use these functions. See "Terminology and Symbols" on page 3 for the meaning of compatibility terms used in the tables.

CP Changes

This section identifies the CP externals that have changed since VM/ESA 2.3.0. It contains the following subsections:

- "CP Commands"
- "User Directory Control Statements" on page 613
- "CP DIAGNOSE Codes" on page 614
- "CP Macros" on page 614
- "CP Messages" on page 614

CP Commands

Table 152 lists the CP commands that have changed since VM/ESA 2.3.0. Refer to the *VM/ESA: CP Command and Utility Reference* for complete descriptions of CP commands.

Table 152 (Page 1 of 3). CP Commands Changed since VM/ESA 2.3.0

Command	Changes
DEFINE (in general)	**Upwardly compatible changes:** • Supports new operand: EXIT. • See DEFINE commands below.
DEFINE CHPID / PATH	**Upwardly compatible changes:** • Supports new operands: CLUSTer_bus_sender_channel, CBS, FICON, FC, FICON_CONVerter, FCV, OSA_Direct_express, OSD, OSA_Express, OSE. • New return codes for message HCP6806E.
DEFINE CRYPTO	**Upwardly compatible changes:** • New response. • New message: HCP1716E.
DEFINE CU / CNTLUNIT	**Upwardly compatible changes:** • The 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. • The range of permitted values for CU_LOGICAL_ADDRESS has been increased.

Table 152 (Page 2 of 3). CP Commands Changed since VM/ESA 2.3.0

Command	Changes
DEFINE DEVICE / IODEVICE	**Upwardly compatible changes:** • The 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.
DELETE CHPID / PATH	**Upwardly compatible changes:** • New return codes for message HCP6806E.
DELETE CU / CNTLUNIT	**Upwardly compatible changes:** • The 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.
DELETE DEVICE / IODEVICE	**Upwardly compatible changes:** • The 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.
DISPLAY (in general)	**Upwardly compatible changes:** • See DISPLAY commands below.
DISPLAY Registers	**Upwardly compatible changes:** • Supports new operand: FPC. • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New response when FPC operand is used. • New messages: HCP6153E, HCP6154E.
MODIFY CHPID / PATH	**Upwardly compatible changes:** • New return codes for message HCP6806E.
MODIFY CU / CNTLUNIT	**Upwardly compatible changes:** • New operands: TYPE CF. • The range of permitted values for CU_LOGICAL_ADDRESS has been increased.
QUERY (in general)	**Upwardly compatible changes:** • See QUERY commands below.
QUERY CHPID	**Upwardly compatible changes:** • New responses for the TYPE operand to indicate OSA Express, OSA Direct-Express, cluster-bus-sender, and FICON channels.
QUERY CPLEVEL	**Upwardly compatible changes:** • In the response, the release level value has changed.
QUERY CRYPTO	**Upwardly compatible changes:** • Supports new operand: CAMQS. • Two new responses.
QUERY EXITS	**Upwardly compatible changes:** • Additional response information provided for a dynamic CP exit.
QUERY VIRTUAL CRYPTO	**Upwardly compatible changes:** • New response.

Table 152 (Page 3 of 3). CP Commands Changed since VM/ESA 2.3.0

Command	Changes
QUERY VIRTUAL OSA	**Upwardly compatible changes:** • New lines in the response for OSA devices that use the Queued-Direct-I/O (QDIO) Facility.
SET (in general)	**Upwardly compatible changes:** • See SET commands below.
SET CPTRACE	**Upwardly compatible changes:** • New trace category and trace codes for QDIO instructions.
SET CRYPTO	**Upwardly compatible changes:** • New operands: MODIFY, ON, OFF. • Changed responses. • Changed messages: HCP1706I, HCP1709E, HCP1710E, HCP1711I. • New messages: HCP1714E, HCP1715E.
STORE (Registers)	**Upwardly compatible changes:** • Supports new operand: FPC *hexword*. • Yreg operands accept register numbers 0-15 if the IEEE Floating Point hardware feature is installed on the processor. • New messages: HCP6153E, HCP6154E.
STORE STATUS	**Upwardly compatible changes:** • 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.
TRACE (in general)	**Upwardly compatible changes:** • New response for floating-point register information. • See TRACE commands below.
TRACE mnemonic2	**Upwardly compatible changes:** • New mnemonic: SIGA.

User Directory Control Statements

Table 153 lists the user directory control statements that have changed since VM/ESA 2.3.0. See the *VM/ESA: Planning and Administration* book for complete descriptions of user directory control statements.

Note: You can use the VMUDQ macro to query the CP user directory. See the *VM/ESA: CP Programming Services* book for more information on this macro.

Table 153. User Directory Control Statements Changed since VM/ESA 2.3.0

Statement	Changes
CRYPTO	**Upwardly compatible changes:** • New operand: MODIFY.
OPTION	**Upwardly compatible changes:** • New operand: DIAG88.

CP DIAGNOSE Codes

Table 154 lists the DIAGNOSE codes that have changed since VM/ESA 2.3.0. See the *VM/ESA: CP Programming Services* book for complete information on the DIAGNOSE codes discussed in this section.

Table 154. DIAGNOSE Codes Changed since VM/ESA 2.2.0

Code	Changes
X'00'	Storage extended identification code **Upwardly compatible changes:** • The value in the program product bit map has changed to indicate the new release level.
X'270'	Pseudo Timer Extended **Upwardly compatible changes:** • Output has been expanded to include the version of DIAGNOSE X'270', the user's default date format, and the system default date format.

CP Macros

Table 155 lists the CP macros that have changed since VM/ESA 2.3.0.

Table 155. CP Macros Changed since VM/ESA 2.3.0

Macro	Changes
HCPTKDEF	**Upwardly compatible changes:** • Supports new conversion type: INSTRUCT.

CP Messages

The following CP messages do not exist in VM/ESA 2.4.0:

HCP8039S
HCP8611T

The following CP messages have changed since VM/ESA 2.3.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CP messages.

HCP1003E
HCP1706I
HCP1709E
HCP1710E
HCP1711I
HCP2234E
HCP6111I
HCP6706E
HCP6789E

CMS Changes

This section identifies the CMS externals that have changed since VM/ESA 2.3.0. It contains the following subsections:

- “CMS Commands”
- “CMS Macros” on page 615
- “Preferred Routines (CSL Routines)” on page 616
- “Compatibility Routines” on page 617
- “CMS Pipelines” on page 617
- “CMS Messages” on page 618

CMS Commands

Table 156 lists the CMS commands that have changed since VM/ESA 2.3.0. See the *VM/ESA: CMS Command Reference* for complete descriptions of CMS commands.

Table 156. CMS Commands Changed since VM/ESA 2.3.0

Command	Changes
FILEDEF	**Upwardly compatible changes:** • Allows LRECL definitions up to 65535 bytes for OS variable spanned records (under XLRI processing) and non-OS CMS files. • Allows BLKSIZE definitions up to 65535 bytes for non-OS CMS files.
LISTFILE	**Incompatible changes:** • Message DMS550E changed to DMS765E.
LKED	**Upwardly compatible changes:** • The defaults for *value1* and *value2* on the SIZE option have been increased to 400K and 100K, respectively.
MOVEFILE	**Upwardly compatible changes:** • Can process OS variable spanned records (under XLRI processing) and non-OS CMS files with record lengths up to 65535 bytes. • Adjusts output file sizes for compatibility between CMS and OS • Allows greater FILEDEF default flexibility for file attributes (RECFM, LRECL, BLKSIZE) • If default size values are used, fixes record truncation problems when moving data files from fixed to variable format • New message: DMS1116E.
PIPE	**Upwardly compatible changes:** • See “CMS Pipelines” on page 617.
QUERY CMSLEVEL	**Upwardly compatible changes:** • In the response, the CMS level value has changed.
QUERY CMSREL	**Upwardly compatible changes:** • In the response, the release level value has changed.
QUERY FILEDEF	**Upwardly compatible changes:** • Supports a new optional operand, ATTRIBUT, and its response.

CMS Macros

Table 157 lists the CMS preferred macros that have changed since VM/ESA 2.3.0. Table 158 lists the OS simulation macros that have changed. Table 159 lists the simulated OS/MVS supervisor calls that have changed.

Preferred Macros

See the *VM/ESA: CMS Application Development Reference for Assembler* for complete descriptions of these macros.

Table 157. CMS Preferred Macros Changed since VM/ESA 2.3.0

Macro	Changes
DMSQEFL	**Upwardly compatible changes:** • Returns new values for the CMS level.

OS Simulation Macros and Supervisor Calls

See the *VM/ESA: CMS Application Development Guide for Assembler* for information about the OS macros and supervisor calls that CMS simulates.

Table 158. OS Simulation Macros Changed since VM/ESA 2.3.0

Macro	Changes
CLOSE	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
DCB	**Upwardly compatible changes:** • 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.
GET	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.
OPEN	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
PUT	**Upwardly compatible changes:** • Can process QSAM I/O to variable spanned QSAM records or non-OS CMS files up to 65535 bytes in length.

Table 159. OS Simulation Supervisor Calls Changed since VM/ESA 2.3.0

SVC	Changes
SVC 19 (OPEN)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.
SVC 20 (CLOSE)	**Upwardly compatible changes:** • Can process OS formatted variable spanned QSAM records (under XLRI) or non-OS CMS files up to 65535 bytes in length.

Preferred Routines (CSL Routines)

Table 160 lists the CSL routines that have changed since VM/ESA 2.3.0. See the *VM/ESA: CMS Application Development Reference* for complete descriptions of the routines in this table.

Table 160. CSL Routines Changed since VM/ESA 2.3.0

Routine	Changes
DMSQEFL	**Upwardly compatible changes:** • Supports new values for *cp_level* and *cms_level*
DMSQSFSL	**Upwardly compatible changes:** • Supports new value for *server_level*

Compatibility Routines

Table 161 lists the CMS compatibility routines that have changed between VM/ESA 2.3.0 and VM/ESA 2.4.0. See the *VM/ESA: CMS Application Development Guide for Assembler* for more information about the routines in this table.

Table 161. CMS Compatibility Routines Changed since VM/ESA 2.3.0

CMS Routine	Explanation
DMSTVS	**Upwardly compatible changes:** • 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.

CMS Pipelines

Table 162 lists the Pipelines stages, subcommands, and assembler macros that have changed since VM/ESA 2.3.0. For complete descriptions of these functions, see the *VM/ESA: CMS Pipelines Reference*.

Table 162. Pipelines Stages, Subcommands, and Assembler Macros Changed since VM/ESA 2.3.0

Stage, Subcommand, or Macro	Changes
BFSQUERY stage	**Upwardly compatible changes:** • The system information in the output record has been updated.
DATECONVERT stage	**Upwardly compatible changes:** • Provides timestamp output.
PIPEPVR macro	**Upwardly compatible changes:** • New option: *label*.
QUERY stage	**Upwardly compatible changes:** • The information returned for QUERY LEVEL indicates the new level.
READER stage	**Upwardly compatible changes:** • New options: HOLD, NOHOLD, KEEP, NOKEEP, PURGE.

Pipelines Messages

The following Pipelines messages have changed since VM/ESA 2.3.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of Pipelines messages.

FPL1170E
FPL1171W
FPL1182E
FPL1183E
FPL1184E
FPL1185E
FPL1186W

CMS Messages

The following CMS messages have changed since VM/ESA 2.3.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of CMS messages.

DMS2010E

REXX/VM Changes

This section identifies the REXX/VM externals that have changed since VM/ESA 2.3.0. It contains the following subsections:

- "REXX/VM Functions"

REXX/VM Functions

Table 163 lists the REXX/VM functions that have changed since VM/ESA 2.3.0. For complete descriptions of these functions, see the *VM/ESA: REXX/VM Reference*.

Table 163. REXX/VM Functions Changed since VM/ESA 2.3.0

Function	Changes
DATE	**Upwardly compatible changes:** • New parameters: *output_separator_char*, *input_separator_char*.

GCS Changes

This section identifies the GCS externals that have changed since VM/ESA 2.3.0. It contains the following subsection:

- "GCS Commands and Macros"

GCS Commands and Macros

Table 164 lists the GCS commands and macros that have changed since VM/ESA 2.3.0. See *VM/ESA: Group Control System* for complete descriptions of all GCS commands and macros.

Table 164. GCS Commands and Macros Changed since VM/ESA 2.3.0

Command/Macro	Changes
GCSLEVEL macro	**Upwardly compatible changes:** • Additional equate for new release level.
QUERY GCSLEVEL command	**Upwardly compatible changes:** • Response contains new values.

VMSES/E Changes

This section identifies the VMSES/E externals that have changed since VM/ESA 2.3.0. It contains the following subsection:

- “VMSES/E Messages”

Also see “Summary of Service Exec Changes [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0, 2.3.0]” on page 250.

VMSES/E Messages

The following VMSES/E messages have changed since VM/ESA 2.3.0. Refer to *VM/ESA: System Messages and Codes* for complete descriptions of VMSES/E messages.

VMF1203E
VMF1205E
VMF1207I
VMF1211I
VMF2066E
VMF2114R
VMF2118I
VMF2119I
VMF2120W
VMF2152E
VMF2206W
VMF2767I

Part 4. Appendixes, Glossary, Bibliography, Index

Appendix A. Hardware and Software Inventory Tables

This appendix contains tables that you can use to fill in information about your hardware and software inventory. Chapter 4, “Preparing Hardware and Software Inventory” shows examples of how to fill in the tables.

Processors

See “Processors” on page 13 for an example of how to fill in this table.

Old	New	Type and model	Patch or EC level	PT	MSTO	XSTO	Modes	PTFs	REL	PC	EC	SI

Column definitions:

Column	**What it means**
Old	Check this column if you currently use this processor in your old system.
New	Check this column if you plan to use this processor in your new system.
Patch/EC level	Patch or EC level required. The new system requires you to be at a certain Engineering Change level. See your IBM representative for help if your processor is not at the latest level.
PT	Can the processor be physically partitioned?
MSTO	The amount of main storage you have.
XSTO	The amount of extended storage (XSTORE) you have.
Modes	The processor modes supported by the processor, S/370, ESA/370, or ESA/390. See the VM processor support matrix for this information.
PTFs	Service you need to apply to your current system due to the hardware you plan to use in the new system. It may be critical that you apply hardware service to your current system in case you have to back out.
REL	The supported VM releases that you are interested in.
PC	The number of parallel channels.
EC	The number of ESCON* channels.
SI	Does it have VMA/SIE assist?

DASD Used and System DASD Layout

See “DASD Used and System DASD Layout” on page 14 for an example of how to fill in this table.

Old	New	Volume label	Type and model	Dev Num	Used for	2.4.0?	SH?	ES?

Column definitions:

Column **What it means**
Old Check this column if you currently use this DASD in your old system.
New Check this column if you plan to use this DASD in your new system.
Dev Num Real device number of the DASD.
2.4.0? Is this DASD supported in VM/ESA 2.4.0?
SH? Is this DASD shared? Shared DASD could be CSE volumes, volumes used with reserve/release, and so on.
ES? Is the DASD attached through an ESCON controller?

System DASD Layout

O N B	Volume	CP nuc	WARM	CKPT	TDISK	PARM	DRCT	Paging	Spool	System MDISKS	Other (what for)

Column definitions:

O/N/B Specify 'O' if you use it in the old system, 'N' if you use it in the new system, 'B' for both old and new.
Volume Volume label.
CP nuc CP nucleus, start and end.
WARM Warmstart data area, start and end.
CKPT Checkpoint data area, start and end.
TDISK TDISK space, start and end.
PARM PARM space, start and end; PARM space is used with the CP configurability function (which is new if you are converting from VM/ESA 1.1.5 370 Feature). The *VM/ESA: Planning and Administration* book talks about the CP configurability function and PARM space.
DRCT User directory, start and end.
System MDISKS System minidisks and SFS space, start and end.
Other (what for) Note what it is for and the start and end.

O N B	Volume	CP nuc	WARM	CKPT	TDISK	PARM	DRCT	Paging	Spool	System MDISKS	Other (what for)

Column definitions:

O/N/B Specify 'O' if you use it in the old system, 'N' if you use it in the new system, 'B' for both old and new.
Volume Volume label.
CP nuc CP nucleus, start and end.
WARM Warmstart data area, start and end.
CKPT Checkpoint data area, start and end.
TDISK TDISK space, start and end.
PARM PARM space, start and end; PARM space is used with the CP configurability function (which is new if you are converting from VM/ESA 1.1.5 370 Feature). The *VM/ESA: Planning and Administration* book talks about the CP configurability function and PARM space.
DRCT User directory, start and end.
System MDISKS System minidisks and SFS space, start and end.
Other (what for) Note what it is for and the start and end.

Other Devices

See “Other Devices” on page 16 for an example of how to fill in this table.

Old	New	Device	Type and model	2.4.0?	Device number

Column definitions:

Column **What it means**
Old Check this column if you currently use this device in your old system.
New Check this column if you plan to use this device in your new system.
2.4.0? Is this device supported by VM/ESA 2.4.0?

Guests

See “Guests” on page 16 for an example of how to fill in this table.

Old	New	Guest name and release level	Type of guest (V=R, V=F, V=V)	Type of VM it can run in (370, XA, ESA)

Column definitions:

Column	What it means
Old	Check this column if the guest is on your old system.
New	Check this column if you plan to use this guest in your new system.

IBM Licensed Products or Vendor Products

See “IBM Licensed Products or Vendor Products” on page 17 for an example of how to fill in this table.

OLD	NEW	Name and release level	KP	US	REL	VM	16M	BCK	RES	RUN	DEP

Column definitions:

Column **What it means**
OLD Check this column if you currently use this product in your old system.
NEW Check this column if you plan to use this product in your new system.
KP Keep — Must you keep this product (Yes, No, or Maybe)?
US Usage — How much is the product used (Frequently, Sometimes, or Rarely)?
REL VM releases it can run on that you are interested in. Note that some products or parts of products that are not supported may actually run sufficiently on the new system to use during conversion. If you use an unsupported product or release of a product, IBM will not provide service for that product or for problems in VM/ESA brought about by use of the product.
VM Virtual machine it can run in (370, XA, XC).
16M Can it run above the 16MB line?
BCK Can it run in a backlevel CMS on the new system?
RES Where does it reside, SFS or minidisk (MD)?
RUN Where does it run, segment, module, nucleus, or elsewhere?
DEP Dependencies, such as h/w, PTFs, pre-reqs, co-reqs.

OLD	NEW	Name and release level	KP	US	REL	VM	16M	BCK	RES	RUN	DEP

Column definitions:

Column **What it means**
OLD Check this column if you currently use this product in your old system.
NEW Put a check in this column if you plan to use this product in your new system.
KP Keep — Must you keep this product (Yes, No, or Maybe)?
US Usage — How much is the product used (Frequently, Sometimes, or Rarely)?
REL VM releases it can run on that you are interested in.
VM Virtual machine it can run in (370, XA, XC).
16M Can it run above the 16MB line?
BCK Can it run in a backlevel CMS on the new system?
RES Where does it reside, SFS or minidisk (MD)?
RUN Where does it run, segment, nucleus, or elsewhere?
DEP Dependencies, such as h/w, PTFs, pre-reqs, co-reqs.

System Applications

See “System Applications” on page 18 for an example of how to fill in this information.

OLD	NEW	Name and release level	KP	US	REL	VM	16M	BCK	RES	RUN	Time	DEP

Column definitions:

Column	What it means
OLD	Check this column if you currently use this program in your old system.
NEW	Check this column if you plan to use this program in your new system.
KP	Keep — Must you keep this program (Yes, No, or Maybe)?
US	Usage — Is the program used Frequently, Sometimes, or Rarely?
REL	VM releases it can run on that you are interested in.
VM	Virtual machine it can run in (370, XA, XC).
16M	Runs above the 16MB line?
BCK	Runs in a backlevel CMS on the new system?
RES	Where it resides, SFS or minidisk (MD).
RUN	Where it runs, segment or nucleus.
Time	Estimated time to make changes.
DEP	Dependencies, such as h/w, PTFs, pre-reqs, co-reqs.

OLD	NEW	Name and release level	KP	US	REL	VM	16M	BCK	RES	RUN	Time	DEP

Column definitions:

Column	What it means
OLD	Check this column if you currently use this program in your old system.
NEW	Check this column if you plan to use this program in your new system.
KP	Keep — Must you keep this program (Yes, No, or Maybe)?
US	Usage — Is the program used Frequently, Sometimes, or Rarely?
REL	VM releases it can run on that you are interested in.
VM	Virtual machine it can run in (370, XA, XC).
16M	Runs above the 16MB line?
BCK	Runs in a backlevel CMS on the new system?
RES	Where it resides, SFS or minidisk (MD).
RUN	Where it runs, segment, module, or elsewhere.
Time	Estimated time to make changes.
DEP	Dependencies, such as h/w, PTFs, pre-reqs, co-reqs.

Your Users' Applications

See “System Applications” on page 18 for an example of how this type of table might be filled.

OLD	NEW	Name and release level	KP	US	REL	VM	16M	BCK	RES	RUN	Time	DEP

Column definitions:

Column **What it means**
OLD Check this column if you currently use this program in your old system.
NEW Check this column if you plan to use this program in your new system.
KP Keep — Must you keep this program (Yes, No, or Maybe)?
US Usage — Is the program used Frequently, Sometimes, or Rarely?
REL VM releases it can run on that you are interested in.
VM Virtual machine it can run in (370, XA, XC).
16M Runs above the 16MB line?
BCK Runs in a backlevel CMS on the new system?
RES Where it resides, SFS or minidisk (MD).
RUN Where it runs, segment, module, or elsewhere.
Time Estimated time to make changes.
DEP Dependencies, such as h/w, PTFs, pre-reqs, co-reqs.

Local Mods

See “Local Mods” on page 20 for an example of how you can fill in this table for your local modifications to VM.

Name	Description	Modules or control blocks it affects	ESA	KP	US	Time

Column definitions:

Column	What it means
ESA	Can the modification be replaced by function in the new system?
KP	Keep — Must you keep this modification (Yes, No, or Maybe)?
US	Usage — Is the modification used Frequently, Sometimes, or Rarely?
Time	Estimated time to make changes.

Appendix B. Full-Page Tables

The following has full-page empty tables you can use to fill information in.

Tables for Information for System Configuration Strategies

You can use the following table for noting the general characteristics of your system configuration that are important to the conversion. See "General Characteristics of Your Environment" on page 22 and "Examples of Determining What Order to Plan Your Strategies" on page 24 for ideas of what to put in this table. Use the information you filled in or thought about in Chapter 4, "Preparing Hardware and Software Inventory."

Processors and I/O characteristics

Use the following tables to fill in any general characteristics of your users and their applications or products that are important to the conversion strategy. Try to group your users. Use the second column to indicate what you think the priority or the order is that the groups should be moved. See “General Characteristics of Your Environment” on page 22 and “Examples of Determining What Order to Plan Your Strategies” on page 24 for ideas of what to put in this table. Use the information you filled in or thought about in Chapter 4, “Preparing Hardware and Software Inventory.”

Table 165. User and application characteristics and conversion order

User and application/product group characteristics	**Priority or order for conversion**

Table 166. User and application characteristics and conversion order

User and application/product group characteristics	Priority or order for conversion

Table for Saved Segment Information

Use this table to fill in information about saved segments. See Table 9 on page 83 or Table 10 on page 94 for an example of how to fill in this information.

Product	Segment Name	Space Name	Default Location	Size (pages)	Run Above 16MB?	How is Segment Built?

Product	Segment Name	Space Name	Default Location	Size (pages)	Run Above 16MB?	How is Segment Built?

Appendix C. APARs Needed During Conversion

The following tables list the APARs, by release, that can help you during your conversion to VM/ESA 2.4.0.

APARs to Apply to VM/ESA 1.1.5 370 Feature

Table 167. APARs Needed for Conversion from VM/ESA 1.1.5 370 Feature

Area Affected	APAR Number	Description
TSAF	VM46331, VM52077	Allows a VM/ESA 1.1.5 370 Feature TSAF virtual machine to run in a collection with a VM/ESA 2.4.0 TSAF virtual machine.

APARs to Apply to VM/ESA 1.2.0

Table 168. APARs Needed for Conversion from VM/ESA 1.2.0

Area Affected	APAR Number	Description
Mixed directory APARs	VM55600 VM58022	Allows directory statements introduced in VM/ESA 1.2.1 to exist in a source directory processed by VM/ESA 1.2.0 DIRECTXA. Allows directory statements introduced in VM/ESA 1.2.2 to exist in a source directory processed by VM/ESA 1.2.0 DIRECTXA.
System IPL	VM59271	Allows a VM/ESA 1.2.0, VM/ESA 1.2.1, or VM/ESA 1.2.2 system to be IPLed using the checkpoint area information related to *ACCOUNT, *LOGREC, *SYMPTOM and *CONFIG after a prior IPL of VM/ESA 2.4.0.

APARs to Apply to VM/ESA 1.2.1

Table 169. APARs Needed for Conversion from VM/ESA 1.2.1

Area Affected	APAR Number	Description
Mixed directory APARs	VM58022	Allows directory statements introduced in VM/ESA 1.2.2 to exist in a source directory processed by VM/ESA 1.2.1 DIRECTXA.
System IPL	VM59271	Allows a VM/ESA 1.2.0, VM/ESA 1.2.1, or VM/ESA 1.2.2 system to be IPLed using the checkpoint area information related to *ACCOUNT, *LOGREC, *SYMPTOM and *CONFIG after a prior IPL of VM/ESA 2.4.0.

APARs to Apply to VM/ESA 1.2.2

Table 170. APARs Needed for Conversion from VM/ESA 1.2.2

Area Affected	APAR Number	Description
System IPL	VM59271	Allows a VM/ESA 1.2.0, VM/ESA 1.2.1, or VM/ESA 1.2.2 system to be IPLed using the checkpoint area information related to *ACCOUNT, *LOGREC, *SYMPTOM and *CONFIG after a prior IPL of VM/ESA 2.4.0.

Appendix D. Sample Exec for Finding Empty Files and External Objects in SFS Directories

ALLEO EXEC is a REXX program that can help you find empty files and external objects in SFS directories. This can help if you need to back off of your new file pool server. You may need to make some adjustments to this program in your environment.

The ALLEO EXEC prompts you for:

1. The file pool ID of the file pool you are interested in (just pressing ENTER gives you the default file pool)
2. User ID or user IDs you are interested in (it also accepts nicknames and '*' for all user IDs).

The ALLEO EXEC outputs a file called ALLEO FILES that lists the file names and directories of empty files and external objects. Following is an example of the output from the ALLEO EXEC:

```
 Userid  Name    Type       Blocks    Records   Dirname
--------------------------------------------------------------------------------
CUPR     EMPTY    FILE           0          0    SERVER8:CUPR.FOURQ93.DOC
CUPR     X        X              0          0    SERVER8:CUPR.FOURQ93.DOC
CUPR     EMP      F              0          0    SERVER8:CUPR.ONEQ93.DOC
CUPR     X        X              0          0    SERVER8:CUPR.ONEQ93.DOC
JULIE    EMPTY    CSLLIB         0          0    SERVER8:JULIE.
JULIE    EMPTY    FILE           0          0    SERVER8:JULIE.
JULIE    EMPTY    TEXT           0          0    SERVER8:JULIE.
JULIE    JULIE    MACLIB         0          0    SERVER8:JULIE.
MEANSMAR HCPMODL  UPDATE         0          0    SERVER8:MEANSMAR.VMK1
MARIANNE EXTERNAL OBJECT         0          0    SERVER8:MARIANNE.
MARIANNE HHH      HHH            0          0    SERVER8:MARIANNE.
MARIANNE PROFILE  EXTERN         0          0    SERVER8:MARIANNE.
```

The ALLEO EXEC follows:

```
/************************************************************/
/**   ALLEO  EXEC                                          **/
/************************************************************/
/* ALLEO    Exec lists all empty files and external objects */
/* for a user or users.                                     */
/* ALLEO    FILES  A is created with this information.      */
/* You must have administration authority to execute the    */
/* exec when doing user IDs other than your own.            */
/* Also note this may take a LONG time depending on         */
/* the number of files being searched.                      */
/*                                                          */
/* The exec will prompt you for:                            */
/*        File pool ID                                      */
/*        User ID                                           */
/*                                                          */
/*   The userid can be a single user name, nickname of      */
/*       one or more users or "*" for all users in the      */
/*       file pool.                                         */
/***********************************************************/
```

© Copyright IBM Corp. 1992, 1999

```
rc = 0
rs = 0
if arg(1)='?' then signal tell

say 'Enter Filepoolid or hit enter to use current (default) filepool:'
pull filepoolid
If filepoolid = '' then do
  'QUERY FILEPOOL CURRENT (FIFO'         /* Get default filepoolid  */
  pull . filepoolid .
 End
Else do
  If (pos(':',filepoolid)) = 0 Then do
    filepoolid = strip(filepoolid)':'
   End
  End


Say 'Enter userid, nickname or * (all filepool users)'
pull input_userid
If strip(input_userid) = '*' Then Do

/* Get list of all users in the specified filepool.                 */

  'PIPE CMS QUERY ENROLL USER FOR ALL' filepoolid ' | stem fpusers.'
  If rc ¬= 0 Then Exit
  count = 2                /* adjust for header in fpuser.1 */
  num_users = fpusers.0
 End
Else Do                        /* Check for nickname list  */

  Call Get_Names
  count = 1
  num_users = fpusers.0
 End

Parse Var filepoolid filepoolid ':' .  /* strip off the colon now.  */
header = ' Userid  Name    Type       Blocks    Records   Dirname '
'PIPE var header | > ALLEO FILES A'
header2 = copies('-',120)
'PIPE var header2 | >> ALLEO FILES A'

DO WHILE count <= num_users
  parse var fpusers.count userid .
  If userid = '<PUBLIC>' Then Iterate  /* Skip userid public...     */
  say time() ' '
  say 'Processing files for' userid
  Call DO_FILES
  'PIPE stem outlines. | >> ALLEO FILES A'
  outlines.0 = 0
  count = count + 1
End

Exit

DO_FILES:

wuerror = '  '
wuerrorlen = 0
```

```
fileid = '*     *    ' strip(filepoolid)':'strip(userid)'.'
fileidlen = length(fileid)
intent = 'SEARCHALL '
commit = 'NOCOMMIT'
intentlen = length(intent)
commitlen = length(commit)
token = ''
CALL CSL 'DMSOPDIR rc rs fileid fileidlen intent intentlen token'
counter = 1
do while rc < 4 & rc > -1
  rc = 0; rs=0
  infileid = '               '
  infmn='0'
  read=' '; write=' '; ext=' '
  status= ' '
  recfm=' '
  lrecl=0
  numblocks=0
  numrec=0
  date='        '
  time='        '
  userid='        '
  wuerrorlen = 0

  dirid= '  ';outfileid='               '
  dirlen = ''
  dirname = copies(' ',153)
   CALL CSL 'DMSGETDA rc rs token infileid infmn status
                      recfm lrecl numblocks numrec
                      date time userid dir_len dirname'

 Empty = 0; External = 0
 If (rc = 0 | rc = 4) Then Do                /* good return code */
   If (status = '1') & (numrec = 0) Then     /* Empty file?      */
     Empty = 1
   If status = '6' Then                      /* External Object? */
     External = 1
   End
 If (Empty = 1) | (External = 1) Then do
    parse var infileid 1 fname 9 ftype .
    output_line = userid left(fname,8) left(ftype,8),
                  format(numblocks,7,0)'  ',
                  format(numrec,8,0) '  ',
                  dirname
    outlines.counter = output_line
    outlines.0 = counter
    counter = counter+1
   End
  End

 /*   now close the directory  */
CALL CSL 'DMSCLDIR rc rs token commit commitlen'
Return

Get_Names:

  ncount=1
  names = input_userid
 dejavu.=0
 name.=''
 DO WHILE names ¬= ''
    parse var names name names
       'NAMEFIND :NICK' name ':USERID :NODE :LIST (FIFO'
       if rc=0 then do
          pull nname
          pull nnode
          pull list
          if list¬='' then names = list names
          if nname='' & list='' then call error,
          '32 Userid not specified for' name "in '"userid() "NAMES' file."
          else if nname¬='' then do
             if nnode='' then nnode=locnode
             if ¬dejavu.nname.nnode then do
                fpusers.ncount=nname
                ncount=ncount+1
                dejavu.nname.nnode=1
                end
             end
           end
       else do
          if ¬dejavu.name.locnode then do
             fpusers.ncount=name
             ncount=ncount+1
             dejavu.name.locnode=1
             end
          end
    end
 fpusers.0 = ncount-1
 RETURN


Tell:

Say 'ALLEO    Exec lists all the empty files and external    '
Say 'objects in a file pool for a user or list of users.     '
Say 'ALLEO    FILES  A is created with this information.     '
Say 'You must have administration authority to execute the   '
Say 'exec when doing user IDs other than your own.           '
Say 'Also note this may take a LONG time depending on        '
Say 'the number of files being searched.                     '
Say '                                                        '
Say 'The exec will prompt you for:                           '
Say '       File pool ID                                     '
Say '       User ID                                          '
Say '                                                        '
Say '  The user ID can be a single user name, nickname of    '
Say '      one or more users or "*" for all users in the     '
Say '      file pool.                                        '
Say '                                                        '
exit(100)
```

Appendix E. Sample Utilities for CP Configurability

This appendix provides reference information for the following sample utility programs supplied with VM/ESA 2.4.0:

HCPTRIO
HCPTSYS

These programs are shipped on the VM/ESA System DDR tapes. 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 *VM/ESA: 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 *VM/ESA: 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 *VM/ESA: System Operation*.
- The sample utility program, HCPDCON, creates a system configuration file from a running VM/ESA 2.4.0 system. For more information, see the *VM/ESA: Planning and Administration* book.

HCPTRIO

Format

```
►►──HCPTRIO──┬─RDEV──CONFIG──*─────────────────┬─┬─FROM──HCPRIO──ASSEMBLE──*────────────┬──►
             └─out_fn──┬─CONFIG──*───────────┬─┘ └─FROM──in_fn──┬─ASSEMBLE──*─────────┬─┘
                       └─out_ft──┬─*──────┬──┘                  └─in_ft──┬─*─────┬────┘
                                 └─out_fm─┘                              └─in_fm─┘

►──┬──────────────┬──────────────────────────────────────────────────────────────────────►◄
   └─(──REPLace───┘
```

Authorization

Privilege Class: G

Use HCPTRIO to generate a system configuration file (SYSTEM CONFIG) from an HCPRIO 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.

Parameters

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 HCPRIO.

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:

   ```
   hcptrio 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.

2. If the output file already exists and the REPLACE option is not specified, HCPTRIO gives a return code of 12.

Examples

Example 1

```
hcptrio 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

```
hcptrio
```

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

Format

```
            ┌─SYSTEM──CONFIG──*─────────────────┐ ┌─FROM──HCPSYS──ASSEMBLE──*────────────┐
►►──HCPTSYS─┼───────────────────────────────────┼─┼──────────────────────────────────────┼──►
            │          ┌─CONFIG──*────────────┐ │ │             ┌─ASSEMBLE──*─────────┐  │
            └─out_fn───┼──────────────────────┼─┘ └─FROM──in_fn─┼─────────────────────┼──┘
                       │         ┌─*──────┐   │                 │        ┌─*──────┐   │
                       └─out_ft──┼────────┼───┘                 └─in_ft──┼────────┼───┘
                                 └─out_fm─┘                              └─in_fm──┘

►──┬──────────────┬──────────────────────────────────────────────────────────────────►◄
   └─(──REPLace───┘
```

Authorization

Privilege Class: G

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

Parameters

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.

2. If the output file already exists and the REPLACE option is not specified, HCPTSYS gives a return code of 12.

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.

Appendix F. Migrating Programs from VMCF to APPC/VM

The Virtual Machine Communication Facility (VMCF) is maintained in VM/ESA only for compatibility purposes. Applications that use VMCF should be converted to another means of communication. Converting applications to other communication methods may also provide increased performance and fewer storage constraints on your VM/ESA system.

This section highlights the differences and similarities between VMCF and APPC/VM. It also gives you ideas on how you can migrate your VMCF applications to APPC/VM.

Note: Migrating your applications from VMCF to APPC/VM communications protocol is recommended. However, if you prefer migrating from VMCF to IUCV, see Appendix G, “Migrating Programs from VMCF to IUCV” on page 661.

Things to Consider During the Migration

Many VMCF functions map easily to APPC/VM functions. For example, the use of a parameter list to call the function desired, the role of external software interrupts, and the SEND data transfer protocols are similar in VMCF and APPC/VM.

Major differences also exist. One major difference is the concept of a **path** in APPC/VM. A path is established between two virtual machines and controls the flow of data between the two. VMCF does not have an equivalent concept. Instead, it uses the target user ID and a user-generated message ID to distinguish VMCF messages.

Another major difference is that APPC/VM functions are called with macros. VMCF is called with DIAGNOSE code X'68'. The APPC/VM macro interface allows applications to use a larger set of options than the DIAGNOSE code X'68' interface.

Presently, APPC/VM does not support communications in a virtual multiprocessing environment as does VMCF.

With careful planning and an understanding of how your programs use VMCF, you can convert them to APPC/VM. If you do not feel comfortable with your knowledge of VMCF, the *VM/ESA: CP Programming Services* book has more information on VMCF. The following sections highlight the differences between VMCF and APPC/VM and help you identify changes you need to migrate your programs.

Comparison of the Major Features

The following table compares the major features of VMCF to features of APPC/VM.

Table 171 (Page 1 of 2). Comparison of VMCF Features to APPC/VM Features

Feature	VMCF	APPC/VM
Invocation	DIAGNOSE code X'68' parameter list - VMCPARM	APPCVM architected instruction parameter list - IPARML

Table 171 (Page 2 of 2). Comparison of VMCF Features to APPC/VM Features

Feature	**VMCF**	**APPC/VM**
Control Functions	AUTHORIZE	IUCV DCLBFR APPCVM CONNECT IUCV ACCEPT
	CANCEL	None
	IDENTIFY	1. With no path - Connect Pending Interrupt 2. Path established - APPCVM SENDREQ
	QUIESCE	None
	REJECT	None
	RESUME	None
	UNAUTHORIZE	IUCV SEVER, ALL=YES IUCV RTRVBFR APPCVM SEVER
SEND Protocols	SEND RECEIVE	APPCVM SENDDATA, RECEIVE = NO APPCVM RECEIVE
	SEND/RECV RECEIVE REPLY	APPCVM SENDDATA, RECEIVE = YES APPCVM RECEIVE APPCVM SENDDATA
	SENDX	APPCVM RECEIVE APPCVM SENDDATA
Interrupts	Type X'4001' - VMCMHDR	Type X'4000' - IPARML
Send Type Interrupts	IDENTIFY	1. CONNECT (X'81') 2. SENDREQ (X'88')
	SEND	Message Pending (X'89')
	SENDR	Message Pending (X'89')
	SENDX	Message Pending (X'89')
Response Type Interrupts	RECEIVE	Function Complete (X'87')
	REPLY	Function Complete (X'87')
	REJECT	Sever Interrupt (X'83')

Preparing Your Program for APPC/VM Communications

When converting from VMCF to APPC/VM, you first need to enable your application for sending and receiving data with APPC/VM. Also, you must adjust to the differences in APPC/VM. This section outlines the key concepts.

DIAGNOSE Code X'68' Versus APPC/VM Architected Instruction

DIAGNOSE code X'68' starts communications between a VMCF function and CP. It specifies the address of a 40-byte parameter list called VMCPARM through register Rx. VMCPARM starts the requested VMCF function. In addition, this DIAGNOSE code uses register Ry to identify where CP is to place VMCF return codes.

In APPC/VM, a macro is used to format and communicate a function request to CP. This macro generates and processes an architected instruction to request communication services from CP. Similar to VMCF, the APPC/VM macro specifies the address of a 40-byte parameter list called IPARML.

VMCPARM Versus IPARML

The VMCPARM parameter list specifies the following:

- A code indicating the VMCF function to call
- The user ID of the virtual machine associated with the request
- An application-generated message ID for the request
- Other required and optional information for the request

The IPARML parameter list in APPC/VM parallels VMCPARM. For instance, APPC/VM has other required and optional information for each request. Some significant differences between IPARML and VMCPARM are:

- IPARML does not contain a code to call a function because it is handled implicitly through the APPC/VM macro interface.
- IPARML requires a path ID, which was supplied by CP during the APPC/VM CONNECT process, for an established path to the virtual machine associated with the request.

Differences in Error Reporting

VMCF functions relay errors detected by CP through the Ry register designated on the DIAGNOSE code X'68' call or through the error code field VMCMEFLG.

Similarly, APPC/VM functions report errors through fields IPRCODE and IPAUDIT in the IPARML after the unsuccessful completion of a function. In addition, APPC/VM generates condition codes to help you evaluate a function's completion status, and APPC/VM interrupts contain error fields IPCODE and IPAUDIT.

Differences in Software External Interrupt Processing

CP uses special external interrupt code X'4001' to communicate requests to virtual machines that use VMCF. To process these interrupts, a virtual machine must be enabled to receive external interrupts by:

- Having bit 7 in its virtual PSW set on
- Having bit 31 in control register 0 set on
- Using the VMCF AUTHORIZE function to define at least 40 bytes of virtual storage as an interrupt buffer to CP

Also, the external interrupt message header (VMCMHDR) is used for VMCF interrupts. Its fields are, for the most part, a copy of the VMCPARM parameter list fields.

APPC/VM software interrupt processing is similar to VMCF. However, it uses interrupt code X'4000' instead of X'4001'. An external interrupt buffer area of 40 bytes is identified to CP using the IUCV DCLBFR function. Like VMCF's VMCMHDR, APPC/VM's external interrupt buffer is, for the most part, a copy of the IPARML parameter list fields. Also, bit 7 in the virtual PSW and bit 30 in control register 0 must be set on.

Differences in Interrupt Types

VMCF has Send and Response classes of interrupts that closely map to the APPC/VM message class of interrupts. Additionally, you should understand the APPC/VM Connect Pending and Connect Complete interrupt types before converting your VMCF application.

Migrating Your VMCF Control Functions to APPC/VM

You perform more steps to establish the proper communications environment in APPC/VM, but have the same function as VMCF.

Replacing the AUTHORIZE and UNAUTHORIZE Control Functions

The AUTHORIZE and UNAUTHORIZE control functions, which enable and disable virtual machines for communications in VMCF, are replaced in APPC/VM by a more elaborate protocol.

How It Worked in VMCF: To enable a virtual machine to communicate with other virtual machines, a VMCF user issues an AUTHORIZE function that:

- Tells CP that it is ready to communicate using VMCF
- Informs CP of the location of its External Interrupt Buffer
- May select the PRIORITY option
- May limit inbound messages to one virtual machine with the SPECIFIC option
- Has the ability to reset any of these delimiters by issuing a subsequent AUTHORIZE function

Note: A common VMCF practice is to issue a VMCF IDENTIFY to a target (receiving) virtual machine to signal its ability to communicate using VMCF.

This control protocol applies to all possible target virtual machines with the exception of the virtual machine defined by the AUTHORIZE SPECIFIC option. In addition, each VMCF user is limited to 50 outstanding send messages and may have a limit on the number of inbound messages if the MAXVMCFI directory option is used.

To end the VMCF communication environment, the UNAUTHORIZE function is issued.

How It Works in APPC/VM: Enabling a virtual machine to use APPC/VM consists of several more steps than VMCF. The path concept, which guides a conversation between two virtual machines using APPC/VM, is new. To establish an APPC/VM conversation, do the following:

- Ensure each virtual machine has the appropriate directory entries that permit an application to exploit APPC/VM.

- Issue an IUCV DCLBFR. This tells CP that the application is ready to communicate using APPC/VM, and it informs CP of the location of the External Interrupt Buffer for interrupt processing. The interrupt buffer is used for all established APPC/VM paths.
- Issue the appropriate CONNECT function to the target virtual machine. The successful completion of CONNECT provides the application with the appropriate path ID.

  The target virtual machine is then presented with a Connect Pending Interrupt message with its specific path ID. The target virtual machine then issues an IUCV ACCEPT function for that new path, and the source (sending) virtual machine is returned a CONNECT Complete Interrupt message on its side of the path.

There are a few other things you should consider when you are converting your AUTHORIZE and UNAUTHORIZE control functions:

- Because the path concept is in use, APPC/VM has implicitly instituted the AUTHORIZE SPECIFIC option.
- APPC/VM paths allow only one outstanding message per path. In addition, the MAXCONN directory option controls the number of paths that can be established by a virtual machine.
- To change the nature of a conversation with a target virtual machine, you may want to establish an additional path with that virtual machine. You can also completely SEVER the existing path and establish a new path to the target virtual machine reflecting the desired options.
- Finally, the IUCV RTRVBFR function has the same impact on APPC/VM communications activity as the VMCF UNAUTHORIZE function has on VMCF communications.

Replacing the QUIESCE and RESUME Control Functions

There is no explicit APPC/VM replacement for QUIESCE and RESUME.

How It Worked in VMCF: The VMCF QUIESCE function allows a virtual machine to prevent any more send messages from arriving while still allowing the virtual machine to transmit additional messages. Once you issue VMCF RESUME, you can again receive inbound send messages. Then a source virtual machine usually calls the VMCF IDENTIFY function to targeted virtual machines. This restarts VMCF communications.

How It Works in APPC/VM: Although there is no explicit APPC/VM replacement for these VMCF control functions, you can get around this. You can simulate QUIESCE and RESUME in your applications by refusing to issue the APPCVM SENDDATA or the APPCVM RECEIVE functions until you are ready to continue the conversation with the target virtual machine.

Replacing the CANCEL and REJECT Control Functions

Only IUCV provides a comparable substitute for these two VMCF subfunctions. Because APPC/VM enforces the concept of one outstanding message per path at one time, CANCEL and REJECT functions are not needed.

Migrating Your VMCF Data Transfer Functions

This section discusses the migration path for the four VMCF data transfer functions:

VMCF Protocol	VMCF Functions Used for Protocol
SEND	SEND, RECEIVE, and REJECT
SEND/RECV	SEND/RECV, RECEIVE, REPLY, and REJECT
SENDX	SENDX
IDENTIFY	IDENTIFY

For SEND and SEND/RECV protocols, the migration to APPC/VM is straightforward. To emulate SENDX, you can exploit APPCVM RECEIVE's “receive ahead” characteristic. For IDENTIFY, a direct migration path does not exist, but there are ways around this by using Connect Interrupt types or by using APPCVM SENDREQ.

Replacing the SEND Data Transfer Function

APPC/VM SEND is similar to VMCF SEND; however, the APPC/VM path concept is used. To convert the VMCF SEND data transfer function to APPC/VM, make the following adjustments:

- Use the APPCVM SENDDATA, RECEIVE=NO macro.
- Realize that the target virtual machine will now see an APPC/VM message pending interrupt type instead of the VMCF SEND interrupt.
- Realize that the target virtual machine will issue an APPCVM RECEIVE.
- Realize that the source virtual machine will receive an APPC/VM function complete interrupt instead of a VMCF response interrupt.

Replacing the SEND/RECV Data Transfer Functions

The points made for the SEND data transfer function migration are applicable to the SEND/RECV data transfer functions. The macro specifications used to replace the VMCF SEND/RECV functions are:

For the source side	APPCVM SENDDATA, RECEIVE=YES
For the target side	APPCVM RECEIVE APPCVM SENDDATA, RECEIVE=NO.

Replacing the SENDX Data Transfer Function

APPC/VM provides a good replacement for VMCF SENDX. This is possible because the APPCVM RECEIVE macro can be issued on a path before a message has been sent.

The following is a typical scenario that can be used to replace the SENDX function:

- The potential target virtual machine of an APPC/VM Message Pending interrupt issues an APPCVM RECEIVE. It can give an address for the start of the receive buffer area that is located immediately after the interrupt buffer, which was defined earlier with an IUCV DCLBFR. This coordination of the interrupt buffer and receive buffer areas mimics the way VMCF SENDX data is presented to a target virtual machine.
- The source virtual machine issues an APPCVM SENDDATA, RECEIVE=NO. It is then presented with an APPC/VM Function Complete Interrupt immediately following the completion of its SENDDATA.
- The target virtual machine receives an APPC/VM Function Complete interrupt, and the data is placed in the receive buffer.

Replacing the IDENTIFY Data Transfer Function

A VMCF IDENTIFY is frequently used by the source virtual machine to signal a change in its communication status. A common use is to signal that a machine is ready to receive VMCF activity. This function is usually issued by the source virtual machine after a VMCF AUTHORIZE or a VMCF RESUME.

Under APPC/VM, the replacement of the VMCF IDENTIFY is difficult. The use of the APPC/VM Connect Pending interrupt as an IDENTIFY-like signal is one possible substitute. Also, once an APPC/VM path is established, the APPCVM SENDREQ macro may be used like IDENTIFY to imitate the VMCF IDENTIFY data transfer function.

Note: Because there is no APPC/VM equivalent to the VMCF and IUCV QUIESCE/RESUME data transfer protocols, the use of an IDENTIFY function is not applicable to APPC/VM.

Appendix G. Migrating Programs from VMCF to IUCV

Virtual Machine Communication Facility (VMCF) is maintained in VM/ESA only for compatibility purposes. Applications that use VMCF should be converted to another means of communication. Converting applications to other communication methods may also provide increased performance and fewer storage constraints on your VM/ESA system. In VM/ESA 1.1.0, CMS RETRIEVE processing moved from VMCF to IUCV. In VM/ESA 1.1.1, *MSG system service processing moved from VMCF to IUCV. Both achieved increased performance and fewer storage constraints.

This section highlights the differences and similarities between VMCF and IUCV. It also gives you ideas on how you can migrate your VMCF applications to IUCV.

Note: Migrating your applications from VMCF to APPC/VM is the recommended migration route. See Appendix F, “Migrating Programs from VMCF to APPC/VM” on page 653.

Things to Consider During the Migration

Many VMCF functions map easily to IUCV functions. For example, the use of a parameter list to call the function desired, the role of external software interrupts, and the SEND data transfer protocols are similar in VMCF and IUCV.

Major differences also exist. One major difference is the concept of a **path** in IUCV. A path is established between two virtual machines and controls the flow of data between the two. VMCF does not have an equivalent concept. Instead, it uses the target user ID and a user-generated message ID to distinguish VMCF messages.

Another major difference is that IUCV functions are called with macros. VMCF is called with DIAGNOSE code X'68'. The IUCV macro interface allows applications to use a larger set of options than the DIAGNOSE code X'68' interface.

With careful planning and an understanding of how your programs use VMCF, you can convert them to IUCV. If you do not feel comfortable with your knowledge of VMCF, the *VM/ESA: CP Programming Services* book has more information on VMCF. The following sections highlight the differences between VMCF and IUCV and help you identify changes you need to migrate your programs.

Comparison of Major Features

The following table compares the major features of VMCF to comparable features of IUCV and APPC/VM.

Table 172 (Page 1 of 2). Comparison of VMCF Features to IUCV and APPC/VM Features

Feature	VMCF	IUCV	APPC/VM
Invocation	DIAGNOSE code X'68' parameter list - VMCPARM	IUCV architected instruction parameter list - IPARML	APPCVM architected instruction parameter list - IPARML

Table 172 (Page 2 of 2). Comparison of VMCF Features to IUCV and APPC/VM Features

Feature	VMCF	IUCV	APPC/VM
Control Functions	AUTHORIZE	IUCV DCLBFR IUCV CONNECT IUCV ACCEPT	IUCV DCLBFR APPCVM CONNECT IUCV ACCEPT
	CANCEL	IUCV PURGE	None
	IDENTIFY	1. With no path - Connect Pending Interrupt 2. Path established - IUCV SEND 1WAY with PARM data	1. With no path - Connect Pending Interrupt 2. Path established - APPCVM SENDREQ
	QUIESCE	IUCV QUIESCE	None
	REJECT	IUCV REJECT	None
	RESUME	IUCV RESUME	None
	UNAUTHORIZE	IUCV SEVER, ALL=YES IUCV RTRVBFR	IUCV SEVER, ALL=YES IUCV RTRVBFR APPCVM SEVER
SEND Protocols	SEND RECEIVE	IUCV SEND 1WAY IUCV RECEIVE	APPCVM SENDDATA, RECEIVE = NO APPCVM RECEIVE
	SEND/RECEIVE RECEIVE REPLY	IUCV SEND 2WAY IUCV RECEIVE IUCV REPLY	APPCVM SENDDATA, RECEIVE = YES APPCVM RECEIVE APPCVM SENDDATA
	SENDX	None	APPCVM RECEIVE APPCVM SENDDATA
Interrupts	Type X'4001' - VMCMHDR	Type X'4000' - IPARML	Type X'4000' - IPARML
Send Type Interrupts	IDENTIFY	Message pending with PARM data (X'08' and X'09') Connection Resume (X'05') Connect (X'01')	1. CONNECT (X'81') 2. SENDREQ (X'88')
	SEND	Message Pending (X'08' and X'09')	Message Pending (X'89')
	SENDR	Message Pending (X'08' and X'09')	Message Pending (X'89')
	SENDX	None	Message Pending (X'89')
Response Type Interrupts	RECEIVE	Message Complete (X'06' and X'07')	Function Complete (X'87')
	REPLY	Message Complete (X'06' and X'07')	Function Complete (X'87')
	REJECT	Sever Interrupt (X'03')	Sever Interrupt (X'83')

Preparing Your Program for IUCV Communications

When converting from VMCF to IUCV, you first need to enable your application for sending and receiving data with IUCV. Also, you must adjust to the differences in IUCV. This section outlines the key concepts.

DIAGNOSE Code X'68' Versus the IUCV Architected Instruction

DIAGNOSE code X'68' starts communications between a VMCF function and CP. It specifies the address of a 40-byte parameter list called VMCPARM through register Rx. VMCPARM starts the requested VMCF function. In addition, this DIAGNOSE code uses register Ry to identify where CP is to place VMCF return codes.

In IUCV, a macro is used to format and to communicate a function request to CP. This macro generates and processes an architected instruction to request communication services from CP. Similar to VMCF, the IUCV macro specifies the address of a 40-byte parameter list called IPARML.

VMCPARM Versus IPARML

The VMCPARM parameter list specifies the following:

- A code indicating the VMCF function to call
- The user ID of the virtual machine associated with the request
- An application-generated message ID for the request
- Other required and optional information for the request

The IPARML parameter list in IUCV parallels VMCPARM. For instance, the IUCV TRGCLS option can replace the VMCF message ID for certain functions. Also, the IUCV PRMMSG and USERDATA options provide the same ability to include data within an interrupt that is relayed directly to the target virtual machine, which is like any unused fields in the VMCPARM parameter list when a VMCF function is started.

Some significant differences between VMCPARM and IPARML are:

- IPARML does not contain a code to call a function because it is handled implicitly through the IUCV macro interface.
- IPARML requires a path ID, which was supplied by CP during the IUCV CONNECT process, for an established path to the virtual machine associated with the request.

Differences in Error Reporting

VMCF functions relay errors detected by CP through the Ry register designated on the DIAGNOSE code X'68' call or through the error code field VMCMEFLG.

Similarly, IUCV functions report errors through fields IPRCODE and IPAUDIT in the IPARML after the unsuccessful completion of a function. In addition, IUCV generates condition codes to help you evaluate a function's completion status, and IUCV interrupts contain error fields IPCODE and IPAUDIT.

Differences in Software External Interrupt Processing

CP uses special external interrupt code X'4001' to communicate requests to virtual machines that use VMCF. To process these interrupts, a virtual machine must be enabled to receive external interrupts by:

- Having bit 7 in its virtual PSW set on
- Having bit 31 in control register 0 set on
- Using the VMCF AUTHORIZE function to define at least 40 bytes of virtual storage as an interrupt buffer to CP

Also, the external interrupt message header (VMCMHDR) is used for VMCF interrupts. Its fields are, for the most part, a copy of the VMCPARM parameter list fields.

IUCV software interrupt processing is similar to VMCF. However, it uses interrupt code X'4000' instead of X'4001'. An external interrupt buffer area of 40 bytes is identified to CP using the IUCV DCLBFR function. Like VMCF's VMCMHDR, IUCV's external interrupt buffer is, for the most part, a copy of the IPARML parameter list fields. Also, bit 7 in the virtual PSW and bit 30 in control register 0 must be set on.

Differences in Interrupt Types

VMCF has Send and Response classes of interrupts that closely map to the IUCV message class of interrupts. Additionally, you should understand the IUCV Connect Pending and Connect Complete interrupt types before converting your VMCF application.

Migrating Your VMCF Control Functions to IUCV

You perform more steps to establish the proper communications environment in IUCV, but you have the same function as VMCF.

Replacing the AUTHORIZE and UNAUTHORIZE Control Functions

The AUTHORIZE and UNAUTHORIZE control functions, which enable and disable virtual machines for communications in VMCF, are replaced in IUCV by a more elaborate protocol.

How It Worked in VMCF

To enable a virtual machine to communicate with other virtual machines, a VMCF user issues an AUTHORIZE function that:

- Tells CP it is ready to communicate using VMCF
- Informs CP of the location of its External Interrupt Buffer
- May select the PRIORITY option
- May limit inbound messages to one virtual machine with the SPECIFIC option
- Has the ability to reset any of these delimiters by issuing a subsequent AUTHORIZE function

Note: A common VMCF practice is to call a VMCF IDENTIFY to a target (receiving) virtual machine to signal its ability to communicate using VMCF.

This control protocol applies to all possible target virtual machines with the exception of the virtual machine defined by the AUTHORIZE SPECIFIC option. In addition, each VMCF user is limited to 50 send messages outstanding and may have a limit on the number of inbound messages (if the MAXVMCFI directory option is used).

To end the VMCF communication environment, the UNAUTHORIZE function is issued.

How it Works in IUCV

Enabling a virtual machine to use IUCV consists of more steps than VMCF. The **path** concept, which guides a conversation between two virtual machines using IUCV, is new. To establish an IUCV conversation, do the following:

- Ensure each virtual machine has the appropriate directory entries that permit an application to exploit IUCV.
- Issue an IUCV DCLBFR. This informs CP that the application is ready to communicate using IUCV, and it informs CP of the location of the External Interrupt Buffer for interrupt processing. The interrupt buffer is used for all established IUCV paths.
- Issue an IUCV CONNECT function to the target (receiving) virtual machine.

  The target virtual machine is then presented with a Connect Pending Interrupt with its path ID. The target virtual machine then issues an IUCV ACCEPT function for that new path, and the source (sending) virtual machine is returned a CONNECT Complete Interrupt message on the appropriate path.

There are a few other things you should consider when you are converting your AUTHORIZE and UNAUTHORIZE control functions.

- Because the path concept is in use, IUCV has implicitly instituted the AUTHORIZE SPECIFIC option.
- The IUCV CONNECT and the IUCV ACCEPT functions allow for PRIORITY message processing to be provided on an individual path basis.
- IUCV paths allow you to specify the maximum number of messages permitted on each path by using the MSGLIM parameter on the IUCV CONNECT and IUCV ACCEPT functions or the MSGLIMIT operand on the IUCV directory control statement. In VMCF, you are limited to 50 outstanding send messages, and you can limit the number of inbound messages by specifying the MAXVMCFI directory option.
- To change the nature of a conversation with a possible target virtual machine, you may want to establish an additional path to that virtual machine. You can also completely SEVER the existing path and establish a new path to the target virtual machine reflecting the desired options.
- The IUCV RTRVBFR function has the same impact on IUCV communications activity as the VMCF UNAUTHORIZE function has on VMCF communications.

Replacing the QUIESCE and RESUME Control Functions

VMCF QUIESCE and RESUME are basically the same as IUCV QUIESCE and RESUME.

How It Worked in VMCF

The VMCF QUIESCE function allows a virtual machine to prevent any more send messages from arriving while still allowing the virtual machine the option to transmit additional messages. Once you issue VMCF RESUME, you can again receive inbound send messages. Then, a source virtual machine usually calls the VMCF IDENTIFY function to target virtual machines. This restarts VMCF communications between the two.

How It Works in IUCV

The IUCV QUIESCE and IUCV RESUME functions match up easily to their VMCF counterparts. In addition, each function allows the virtual machine to QUIESCE and to RESUME on an individual as well as group basis. The IUCV CONNECT QUIESCE and CONNECT RESUME interrupts that are automatically generated, also easily replace the need to make an explicit call like the VMCF IDENTIFY function to end a virtual machine's quiesced state.

Replacing the CANCEL and REJECT Control Functions

IUCV provides functions similar to VMCF CANCEL and REJECT. Because IUCV uses the concept of more than one outstanding send message on an IUCV path, the IUCV PURGE function closely mimics the VMCF CANCEL function. Accordingly, the IUCV REJECT function is quite similar to the VMCF REJECT function with minor variations.

Migrating Your VMCF Data Transfer Functions

This section discusses the migration path for the four VMCF data transfer functions:

VMCF Protocol	VMCF Functions Used for Protocol
SEND	SEND, RECEIVE, and REJECT
SEND/RECV	SEND/RECV, RECEIVE, REPLY, and REJECT
SENDX	SENDX
IDENTIFY	IDENTIFY

For the SEND and SEND/RECV protocols, the migration to IUCV is straight forward. For the SENDX protocol, there is no replacement in IUCV; you have to use an APPC/VM replacement. For IDENTIFY, there is no direct migration, but, you can get around this by using Connect Interrupt types or by using an IUCV SEND, TYPE=1WAY with the PARMMSG option.

Replacing the SEND Data Transfer Function

IUCV SEND is similar to VMCF SEND; however, the IUCV path concept is used. To convert the VMCF SEND data transfer function to IUCV, make the following adjustments:

- Use the IUCV SEND, TYPE=1WAY function.
- Realize that the target virtual machine will now see an IUCV Message Pending interrupt type instead of the VMCF SEND interrupt.
- Realize that the target virtual machine will issue an IUCV RECEIVE or IUCV REJECT function in place of the VMCF RECEIVE or the VMCF REJECT function.
- Realize that a source virtual machine will receive the appropriate IUCV Message Complete Interrupt type instead of a VMCF Response Interrupt.

Replacing the SEND/RECV Data Transfer Functions

The points made for the VMCF SEND data transfer function migration are applicable to the SEND/RECV data transfer function. The IUCV SEND, TYPE=2WAY protocol, which uses the IUCV SEND, TYPE=2WAY, IUCV RECEIVE, IUCV REPLY and IUCV REJECT functions, is similar to the VMCF SEND/RECV data transfer functions.

Replacing the SENDX Data Transfer Function

IUCV does not provide a replacement for the SENDX data transfer protocol. If this protocol is critical to your applications, consider migrating to APPC/VM instead of IUCV. See Appendix F, “Migrating Programs from VMCF to APPC/VM” on page 653.

Replacing the IDENTIFY Data Transfer Function

A VMCF IDENTIFY is used primarily by the source virtual machine to signal a change in its communication status. A common use is to signal that a machine is ready to receive VMCF activity. This function is usually issued by the source virtual machine after a VMCF AUTHORIZE or a VMCF RESUME.

There are several ways to replace the VMCF IDENTIFY data control function. The following are two ways:

- You can treat a Connect Pending Interrupt from a target virtual machine like an IDENTIFY.
- After an IUCV path has been established between two virtual machines using the IUCV CONNECT-ACCEPT control protocol:
  - Both virtual machines can use an IUCV SEND, TYPE=1WAY, PRMMSG=YES like an IDENTIFY.
  - The Connect Resume interrupt from a virtual machine that issues an IUCV RESUME is like the VMCF IDENTIFY function after a VMCF RESUME.

Appendix H. CMS Pipelines Message Cross-Reference [1.1.5, 1.2.0, 1.2.1, 1.2.2, 2.1.0, 2.2.0]

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 the *VM/ESA: System Messages and Codes* book.

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
DMS2572E-FPL338E
DMS2573E-FPL339E
DMS2574I-FPL340I
DMS2575I-FPL341I
DMS2576I-FPL342I
DMS2577E-FPL343E
DMS2578I-FPL344I
DMS2579E-FPL345E
DMS2580E-FPL346E
DMS2581E-FPL347E
DMS2582I-FPL348I
DMS2583E-FPL1015E
DMS2584E-FPL573E
DMS2585E-FPL575E
DMS2586I-FPL369I
DMS2587E-FPL576E
DMS2588E-FPL611E
DMS2589I-FPL612I
DMS2590E-FPL613E
DMS2591E-FPL614E
DMS2592E-FPL615E
DMS2593E-FPL616E
DMS2594E-FPL617E
DMS2595W-FPL620W
DMS2596E-FPL627E
DMS2597E-FPL635E
DMS2598E-FPL1019E
DMS2599E-FPL371E
DMS2600W-FPL564W
DMS2601E-FPL685E
DMS2602E-FPL639E
DMS2603E-FPL642E
DMS2604E-FPL307E
DMS2605E-FPL651E
DMS2606E-FPL652E
DMS2607E-FPL653E
DMS2608E-FPL654E
DMS2609E-FPL655E
DMS2610E-FPL656E
DMS2611E-FPL657E
DMS2612E-FPL660E
DMS2613E-FPL662E
DMS2614E-FPL664E
DMS2615E-FPL1032E
DMS2616E-FPL1033E
DMS2618E-FPL1036E
DMS2619E-FPL686E
DMS2620E-FPL1038E
DMS2621E-FPL1039E
DMS2622E-FPL1040E
DMS2623E-FPL1041E
DMS2624E-FPL1049E
DMS2625E-FPL680E
DMS2626E-FPL663E
DMS2627E-FPL694E
DMS2628E-FPL1124E
DMS2629E-FPL700E
DMS2630E-FPL701E
DMS2631I-FPL702I
DMS2632I-FPL703I
DMS2633E-FPL704E
DMS2634E-FPL705E
DMS2635E-FPL706E
DMS2636E-FPL707E
DMS2637E-FPL708E
DMS2638E-FPL709E
DMS2639E-FPL710E
DMS2640E-FPL711E
DMS2641E-FPL712E
DMS2642E-FPL713E
DMS2643E-FPL714E
DMS2644E-FPL715E
DMS2645E-FPL716E
DMS2646I-FPL717I
DMS2647I-FPL718I
DMS2648I-FPL719I
DMS2649I-FPL720I
DMS2650E-FPL000E
DMS2651I-FPL001I
DMS2652I-FPL002I
DMS2653I-FPL003I
DMS2654I-FPL004I
DMS2655E-FPL010E
DMS2656E-FPL011E
DMS2657E-FPL012E
DMS2658I-FPL721I
DMS2659E-FPL014E
DMS2660E-FPL015E
DMS2661E-FPL016E
DMS2662E-FPL017E
DMS2663E-FPL018E
DMS2664W-FPL019W
DMS2665I-FPL020I
DMS2666E-FPL021E
DMS2668E-FPL023E
DMS2669W-FPL024W
DMS2670E-FPL1100E
DMS2672E-FPL027E
DMS2673I-FPL028I
DMS2674E-FPL029E
DMS2675I-FPL030I
DMS2676I-FPL031I
DMS2677I-FPL1110I
DMS2678I-FPL033I
DMS2679I-FPL034I
DMS2680I-FPL035I
DMS2681I-FPL036I
DMS2682I-FPL037I
DMS2683I-FPL038I
DMS2684I-FPL039I
DMS2685E-FPL040E
DMS2686E-FPL041E
DMS2687E-FPL042E
DMS2688E-FPL043E
DMS2689E-FPL044E
DMS2690W-FPL045W
DMS2691E-FPL046E
DMS2692E-FPL047E
DMS2693E-FPL048E
DMS2694E-FPL049E
DMS2695E-FPL050E
DMS2696E-FPL051E
DMS2697E-FPL052E
DMS2698E-FPL053E
DMS2699E-FPL054E
DMS2700I-FPL1111I
DMS2701E-FPL056E
DMS2702I-FPL725I
DMS2703E-FPL058E
DMS2704E-FPL059E
DMS2705E-FPL060E
DMS2706E-FPL061E
DMS2707E-FPL062E
DMS2708E-FPL063E
DMS2709E-FPL064E
DMS2710E-FPL065E
DMS2711E-FPL066E
DMS2712E-FPL067E
DMS2713E-FPL068E
DMS2714E-FPL069E
DMS2715E-FPL070E
DMS2716E-FPL071E
DMS2717E-FPL072E
DMS2718E-FPL073E
DMS2719E-FPL074E
DMS2720E-FPL075E
DMS2721I-FPL076I
DMS2722I-FPL077I
DMS2723E-FPL078E
DMS2724E-FPL079E
DMS2725E-FPL080E
DMS2726E-FPL081E
DMS2727E-FPL082E
DMS2728E-FPL083E
DMS2729E-FPL084E
DMS2730E-FPL085E
DMS2731I-FPL086I
DMS2732E-FPL087E
DMS2733E-FPL088E
DMS2734E-FPL089E
DMS2735E-FPL090E
DMS2736E-FPL091E
DMS2737E-FPL092E
DMS2738E-FPL093E
DMS2739E-FPL094E
DMS2740E-FPL095E
DMS2741E-FPL096E
DMS2742E-FPL097E
DMS2743E-FPL098E
DMS2744E-FPL099E
DMS2745E-FPL100E
DMS2746E-FPL101E
DMS2747E-FPL102E
DMS2748E-FPL103E
DMS2749E-FPL104E
DMS2750E-FPL105E
DMS2751E-FPL107E
DMS2752E-FPL108E
DMS2753E-FPL109E

DMS2754E-FPL110E
DMS2755E-FPL111E
DMS2756E-FPL112E
DMS2757E-FPL113E
DMS2758E-FPL114E
DMS2759E-FPL115E
DMS2760E-FPL116E
DMS2761E-FPL117E
DMS2762E-FPL118E
DMS2763E-FPL119E
DMS2764E-FPL120E
DMS2765I-FPL726I
DMS2766E-FPL122E
DMS2767I-FPL727I
DMS2768E-FPL124E
DMS2769E-FPL125E
DMS2770E-FPL126E
DMS2771E-FPL127E
DMS2772E-FPL128E
DMS2773E-FPL129E
DMS2774E-FPL131E
DMS2775E-FPL132E
DMS2776E-FPL133E
DMS2777E-FPL134E
DMS2778I-FPL728I
DMS2779I-FPL729I
DMS2780E-FPL137E
DMS2781E-FPL138E
DMS2782E-FPL139E
DMS2783E-FPL140E
DMS2784E-FPL141E
DMS2785E-FPL142E
DMS2786E-FPL143E
DMS2787E-FPL144E
DMS2788I-FPL145I
DMS2789E-FPL146E
DMS2790E-FPL147E
DMS2791E-FPL148E
DMS2792E-FPL150E
DMS2793E-FPL151E
DMS2794E-FPL152E
DMS2795E-FPL154E
DMS2795E-FPL732E
DMS2796E-FPL155E
DMS2797E-FPL156E
DMS2798E-FPL157E
DMS2799E-FPL159E
DMS2800E-FPL161E
DMS2801E-FPL162E
DMS2802E-FPL163E
DMS2803E-FPL164E
DMS2804E-FPL165E
DMS2805E-FPL166E
DMS2806E-FPL733E
DMS2807E-FPL169E
DMS2808E-FPL170E
DMS2809E-FPL172E
DMS2810E-FPL173E
DMS2811E-FPL174E
DMS2812E-FPL734E
DMS2813E-FPL735E
DMS2814I-FPL177I
DMS2815E-FPL178E
DMS2816E-FPL179E
DMS2817E-FPL180E
DMS2818E-FPL181E
DMS2819W-FPL182W
DMS2820E-FPL183E
DMS2821E-FPL184E
DMS2822E-FPL185E
DMS2823I-FPL186I
DMS2824E-FPL187E
DMS2825I-FPL189I
DMS2826E-FPL190E
DMS2827E-FPL191E
DMS2828I-FPL192I
DMS2829E-FPL193E
DMS2830E-FPL194E
DMS2831E-FPL195E
DMS2832E-FPL196E
DMS2833E-FPL197E
DMS2834E-FPL198E
DMS2835E-FPL209E
DMS2836E-FPL211E
DMS2837E-FPL212E
DMS2838E-FPL214E
DMS2839E-FPL215E
DMS2840E-FPL736E
DMS2841E-FPL219E
DMS2842E-FPL220E
DMS2843E-FPL222E
DMS2844E-FPL223E
DMS2845E-FPL224E
DMS2846E-FPL225E
DMS2847E-FPL226E
DMS2848E-FPL227E
DMS2849E-FPL229E
DMS2850E-FPL230E
DMS2851E-FPL231E
DMS2852E-FPL232E
DMS2853E-FPL233E
DMS2854E-FPL234E
DMS2855E-FPL235E
DMS2856E-FPL236E
DMS2857E-FPL237E
DMS2858E-FPL238E
DMS2859E-FPL241E
DMS2860W-FPL245W
DMS2861E-FPL253E
DMS2862I-FPL256I
DMS2863E-FPL257E
DMS2864E-FPL261E
DMS2865E-FPL264E
DMS2866E-FPL279E
DMS2867E-FPL280E
DMS2868W-FPL281W
DMS2869E-FPL282E
DMS2870W-FPL283W
DMS2871E-FPL284E
DMS2871E-FPL737E
DMS2872E-FPL287E
DMS2873E-FPL738E
DMS2874E-FPL289E
DMS2875E-FPL290E
DMS2876E-FPL291E
DMS2877E-FPL292E
DMS2878I-FPL293I
DMS2879E-FPL297E
DMS2880I-FPL298I
DMS2881E-FPL740E
DMS2882E-FPL301E
DMS2883E-FPL302E
DMS2884E-FPL303E
DMS2885E-FPL304E
DMS2886E-FPL305E
DMS2887E-FPL308E
DMS2888E-FPL309E
DMS2889E-FPL310E
DMS2890E-FPL311E
DMS2891E-FPL313E
DMS2892E-FPL314E
DMS2893E-FPL315E
DMS2894E-FPL317E
DMS2895E-FPL318E
DMS2896E-FPL319E
DMS2898E-FPL333E
DMS2899E-FPL334E
DMS2900E-FPL350E
DMS2901E-FPL352E
DMS2902E-FPL354E
DMS2903I-FPL355I
DMS2904I-FPL356I
DMS2905E-FPL357E
DMS2906E-FPL358E
DMS2907E-FPL359E
DMS2908E-FPL360E
DMS2909I-FPL361I
DMS2910E-FPL362E
DMS2911E-FPL363E
DMS2912E-FPL364E
DMS2913E-FPL365E
DMS2914E-FPL366E
DMS2915E-FPL741E
DMS2916E-FPL368E
DMS2917E-FPL742E
DMS2918E-FPL370E
DMS2919I-FPL743I
DMS2920I-FPL744I
DMS2921E-FPL373E
DMS2922E-FPL380E
DMS2925E-FPL391E
DMS2926E-FPL392E
DMS2927E-FPL393E
DMS2928E-FPL745E
DMS2929E-FPL400E
DMS2930E-FPL401E
DMS2931I-FPL402I
DMS2932E-FPL746E
DMS2933E-FPL747E
DMS2934E-FPL405E
DMS2934E-FPL748E
DMS2935E-FPL406E
DMS2936E-FPL407E
DMS2937I-FPL412I
DMS2938E-FPL409E
DMS2939E-FPL410E
DMS2940I-FPL411I
DMS2941E-FPL420E
DMS2942I-FPL413I
DMS2943E-FPL749E
DMS2944E-FPL750E
DMS2946E-FPL752E
DMS2947E-FPL753E
DMS2948E-FPL780E
DMS2949E-FPL509E
DMS2950E-FPL510E
DMS2951E-FPL511E
DMS2952E-FPL512E
DMS2953E-FPL513E
DMS2954E-FPL514E
DMS2955E-FPL515E
DMS2956E-FPL516E
DMS2957E-FPL517E
DMS2958E-FPL518E
DMS2959E-FPL530E
DMS2960E-FPL531E
DMS2961E-FPL532E
DMS2962E-FPL533E
DMS2963E-FPL534E
DMS2964E-FPL535E
DMS2965E-FPL536E
DMS2966I-FPL537I
DMS2967I-FPL538I
DMS2968E-FPL539E
DMS2969E-FPL540E
DMS2970E-FPL541E
DMS2971E-FPL542E
DMS2972E-FPL543E
DMS2973I-FPL544I
DMS2974E-FPL545E
DMS2975E-FPL546E
DMS2976E-FPL547E
DMS2977I-FPL548I
DMS2979I-FPL1112I
DMS2980I-FPL552I
DMS2981I-FPL1113I
DMS2982E-FPL554E
DMS2983I-FPL555I
DMS2984E-FPL556E
DMS2986I-FPL1114I
DMS2987I-FPL1115I
DMS2988I-FPL560I
DMS2989E-FPL561E
DMS2990E-FPL562E
DMS2991E-FPL306E
DMS2992E-FPL650E
DMS2993E-FPL569E
DMS2994E-FPL571E
DMS2998E-FPL335E
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
FPL001I-DMS2651I
FPL002I-DMS2652I
FPL003I-DMS2653I
FPL004I-DMS2654I
FPL010E-DMS2655E
FPL011E-DMS2656E
FPL012E-DMS2657E
FPL014E-DMS2659E
FPL015E-DMS2660E
FPL016E-DMS2661E
FPL017E-DMS2662E
FPL018E-DMS2663E
FPL019W-DMS2664W
FPL020I-DMS2665I
FPL021E-DMS2666E
FPL023E-DMS2668E
FPL024W-DMS2669W
FPL027E-DMS2672E
FPL028I-DMS2673I
FPL029E-DMS2674E
FPL030I-DMS2675I
FPL031I-DMS2676I
FPL033I-DMS2678I
FPL034I-DMS2679I
FPL035I-DMS2680I
FPL036I-DMS2681I
FPL037I-DMS2682I
FPL038I-DMS2683I
FPL039I-DMS2684I
FPL040E-DMS2685E
FPL041E-DMS2686E
FPL042E-DMS2687E
FPL043E-DMS2688E
FPL044E-DMS2689E
FPL045W-DMS2690W
FPL046E-DMS2691E
FPL047E-DMS2692E
FPL048E-DMS2693E
FPL049E-DMS2694E
FPL050E-DMS2695E
FPL051E-DMS2696E
FPL052E-DMS2697E
FPL053E-DMS2698E
FPL054E-DMS2699E
FPL056E-DMS2701E
FPL058E-DMS2703E
FPL059E-DMS2704E
FPL060E-DMS2705E
FPL061E-DMS2706E
FPL062E-DMS2707E
FPL063E-DMS2708E
FPL064E-DMS2709E
FPL065E-DMS2710E
FPL066E-DMS2711E
FPL067E-DMS2712E
FPL068E-DMS2713E
FPL069E-DMS2714E
FPL070E-DMS2715E
FPL071E-DMS2716E
FPL072E-DMS2717E
FPL073E-DMS2718E
FPL074E-DMS2719E
FPL075E-DMS2720E
FPL076I-DMS2721I
FPL077I-DMS2722I
FPL078E-DMS2723E
FPL079E-DMS2724E
FPL080E-DMS2725E
FPL081E-DMS2726E
FPL082E-DMS2727E
FPL083E-DMS2728E
FPL084E-DMS2729E
FPL085E-DMS2730E
FPL086I-DMS2731I
FPL087E-DMS2732E
FPL088E-DMS2733E
FPL089E-DMS2734E
FPL090E-DMS2735E
FPL091E-DMS2736E
FPL092E-DMS2737E
FPL093E-DMS2738E
FPL094E-DMS2739E
FPL095E-DMS2740E
FPL096E-DMS2741E
FPL097E-DMS2742E
FPL098E-DMS2743E
FPL099E-DMS2744E
FPL100E-DMS2745E
FPL101E-DMS2746E
FPL102E-DMS2747E
FPL103E-DMS2748E
FPL104E-DMS2749E
FPL105E-DMS2750E
FPL107E-DMS2751E
FPL108E-DMS2752E
FPL109E-DMS2753E
FPL110E-DMS2754E
FPL111E-DMS2755E
FPL112E-DMS2756E
FPL113E-DMS2757E
FPL114E-DMS2758E
FPL115E-DMS2759E
FPL116E-DMS2760E
FPL117E-DMS2761E
FPL118E-DMS2762E
FPL119E-DMS2763E
FPL120E-DMS2764E
FPL122E-DMS2766E
FPL124E-DMS2768E
FPL125E-DMS2769E
FPL126E-DMS2770E
FPL127E-DMS2771E
FPL128E-DMS2772E
FPL129E-DMS2773E
FPL131E-DMS2774E
FPL132E-DMS2775E
FPL133E-DMS2776E
FPL134E-DMS2777E
FPL137E-DMS2780E
FPL138E-DMS2781E
FPL139E-DMS2782E
FPL140E-DMS2783E
FPL141E-DMS2784E
FPL142E-DMS2785E
FPL143E-DMS2786E
FPL144E-DMS2787E
FPL145I-DMS2788I
FPL146E-DMS2789E
FPL147E-DMS2790E
FPL148E-DMS2791E
FPL150E-DMS2792E
FPL151E-DMS2793E
FPL152E-DMS2794E
FPL154E-DMS2795E
FPL155E-DMS2796E
FPL156E-DMS2797E
FPL157E-DMS2798E
FPL159E-DMS2799E
FPL161E-DMS2800E
FPL162E-DMS2801E
FPL163E-DMS2802E
FPL164E-DMS2803E
FPL165E-DMS2804E
FPL166E-DMS2805E
FPL169E-DMS2807E
FPL170E-DMS2808E
FPL172E-DMS2809E
FPL173E-DMS2810E
FPL174E-DMS2811E
FPL177I-DMS2814I
FPL178E-DMS2815E
FPL179E-DMS2816E
FPL180E-DMS2817E
FPL181E-DMS2818E
FPL182W-DMS2819W
FPL183E-DMS2820E
FPL184E-DMS2821E
FPL185E-DMS2822E
FPL186I-DMS2823I
FPL187E-DMS2824E
FPL189I-DMS2825I
FPL190E-DMS2826E
FPL191E-DMS2827E
FPL192I-DMS2828I
FPL193E-DMS2829E
FPL194E-DMS2830E
FPL195E-DMS2831E
FPL196E-DMS2832E
FPL197E-DMS2833E
FPL198E-DMS2834E
FPL209E-DMS2835E
FPL211E-DMS2836E
FPL212E-DMS2837E
FPL214E-DMS2838E
FPL215E-DMS2839E
FPL219E-DMS2841E
FPL220E-DMS2842E
FPL222E-DMS2843E
FPL223E-DMS2844E
FPL224E-DMS2845E
FPL225E-DMS2846E
FPL226E-DMS2847E
FPL227E-DMS2848E
FPL229E-DMS2849E
FPL230E-DMS2850E
FPL231E-DMS2851E
FPL232E-DMS2852E
FPL233E-DMS2853E
FPL234E-DMS2854E
FPL235E-DMS2855E
FPL236E-DMS2856E
FPL237E-DMS2857E
FPL238E-DMS2858E
FPL241E-DMS2859E
FPL245W-DMS2860W
FPL253E-DMS2861E
FPL256I-DMS2862I
FPL257E-DMS2863E
FPL261E-DMS2864E
FPL264E-DMS2865E
FPL279E-DMS2866E
FPL280E-DMS2867E
FPL281W-DMS2868W
FPL282E-DMS2869E
FPL283W-DMS2870W
FPL284E-DMS2871E
FPL287E-DMS2872E
FPL289E-DMS2874E
FPL290E-DMS2875E
FPL291E-DMS2876E
FPL292E-DMS2877E
FPL293I-DMS2878I
FPL297E-DMS2879E
FPL298I-DMS2880I
FPL301E-DMS2882E
FPL302E-DMS2883E
FPL303E-DMS2884E
FPL304E-DMS2885E
FPL305E-DMS2886E
FPL306E-DMS2991E
FPL307E-DMS2604E
FPL308E-DMS2887E
FPL309E-DMS2888E
FPL310E-DMS2889E
FPL311E-DMS2890E
FPL313E-DMS2891E
FPL314E-DMS2892E
FPL315E-DMS2893E
FPL317E-DMS2894E
FPL318E-DMS2895E
FPL319E-DMS2896E
FPL333E-DMS2898E
FPL334E-DMS2899E
FPL335E-DMS2998E
FPL336E-DMS2999E
FPL337E-DMS2571E
FPL338E-DMS2572E
FPL339E-DMS2573E
FPL340I-DMS2574I
FPL341I-DMS2575I
FPL342I-DMS2576I
FPL343E-DMS2577E
FPL344I-DMS2578I
FPL345E-DMS2579E
FPL346E-DMS2580E
FPL347E-DMS2581E
FPL348I-DMS2582I
FPL350E-DMS2900E
FPL352E-DMS2901E
FPL354E-DMS2902E
FPL355I-DMS2903I
FPL356I-DMS2904I
FPL357E-DMS2905E
FPL358E-DMS2906E
FPL359E-DMS2907E
FPL360E-DMS2908E
FPL361I-DMS2909I
FPL362E-DMS2910E
FPL363E-DMS2911E
FPL364E-DMS2912E
FPL365E-DMS2913E
FPL366E-DMS2914E
FPL368E-DMS2916E
FPL369I-DMS2586I
FPL370E-DMS2918E
FPL371E-DMS2599E
FPL373E-DMS2921E
FPL380E-DMS2922E
FPL391E-DMS2925E
FPL392E-DMS2926E
FPL393E-DMS2927E
FPL400E-DMS2929E
FPL401E-DMS2930E
FPL402I-DMS2931I
FPL405E-DMS2934E
FPL406E-DMS2935E
FPL407E-DMS2936E
FPL409E-DMS2938E
FPL410E-DMS2939E
FPL411I-DMS2940I
FPL412I-DMS2937I
FPL413I-DMS2942I
FPL420E-DMS2941E
FPL509E-DMS2949E
FPL510E-DMS2950E
FPL511E-DMS2951E
FPL512E-DMS2952E

FPL513E-DMS2953E
FPL514E-DMS2954E
FPL515E-DMS2955E
FPL516E-DMS2956E
FPL517E-DMS2957E
FPL518E-DMS2958E
FPL530E-DMS2959E
FPL531E-DMS2960E
FPL532E-DMS2961E
FPL533E-DMS2962E
FPL534E-DMS2963E
FPL535E-DMS2964E
FPL536E-DMS2965E
FPL537I-DMS2966I
FPL538I-DMS2967I
FPL539E-DMS2968E
FPL540E-DMS2969E
FPL541E-DMS2970E
FPL542E-DMS2971E
FPL543E-DMS2972E
FPL544I-DMS2973I
FPL545E-DMS2974E
FPL546E-DMS2975E
FPL547E-DMS2976E
FPL548I-DMS2977I
FPL552I-DMS2980I
FPL554E-DMS2982E
FPL555I-DMS2983I
FPL556E-DMS2984E
FPL560I-DMS2988I
FPL561E-DMS2989E
FPL562E-DMS2990E
FPL564W-DMS2600W
FPL569E-DMS2993E
FPL571E-DMS2994E
FPL573E-DMS2584E
FPL575E-DMS2585E
FPL576E-DMS2587E
FPL611E-DMS2588E
FPL612I-DMS2589I
FPL613E-DMS2590E
FPL614E-DMS2591E
FPL615E-DMS2592E
FPL616E-DMS2593E
FPL617E-DMS2594E
FPL620W-DMS2595W
FPL627E-DMS2596E
FPL635E-DMS2597E
FPL639E-DMS2602E
FPL642E-DMS2603E
FPL650E-DMS2992E
FPL651E-DMS2605E
FPL652E-DMS2606E
FPL653E-DMS2607E
FPL654E-DMS2608E
FPL655E-DMS2609E
FPL656E-DMS2610E
FPL657E-DMS2611E
FPL660E-DMS2612E
FPL662E-DMS2613E
FPL663E-DMS2626E
FPL664E-DMS2614E
FPL680E-DMS2625E
FPL685E-DMS2601E
FPL686E-DMS2619E
FPL694E-DMS2627E
FPL700E-DMS2629E
FPL701E-DMS2630E
FPL702I-DMS2631I
FPL703I-DMS2632I
FPL704E-DMS2633E
FPL705E-DMS2634E
FPL706E-DMS2635E
FPL707E-DMS2636E
FPL708E-DMS2637E
FPL709E-DMS2638E
FPL710E-DMS2639E
FPL711E-DMS2640E
FPL712E-DMS2641E
FPL713E-DMS2642E
FPL714E-DMS2643E
FPL715E-DMS2644E
FPL716E-DMS2645E
FPL717I-DMS2646I
FPL718I-DMS2647I
FPL719I-DMS2648I
FPL720I-DMS2649I
FPL721I-DMS2658I
FPL725I-DMS2702I
FPL726I-DMS2765I
FPL727I-DMS2767I
FPL728I-DMS2778I
FPL729I-DMS2779I
FPL732E-DMS2795E
FPL733E-DMS2806E
FPL734E-DMS2812E
FPL735E-DMS2813E
FPL736E-DMS2840E
FPL737E-DMS2871E
FPL738E-DMS2873E
FPL740E-DMS2881E
FPL741E-DMS2915E
FPL742E-DMS2917E
FPL743I-DMS2919I
FPL744I-DMS2920I
FPL745E-DMS2928E
FPL746E-DMS2932E
FPL747E-DMS2933E
FPL748E-DMS2934E
FPL749E-DMS2943E
FPL750E-DMS2944E
FPL752E-DMS2946E
FPL753E-DMS2947E
FPL780E-DMS2948E
FPL1015E-DMS2583E
FPL1019E-DMS2598E
FPL1032E-DMS2615E
FPL1033E-DMS2616E
FPL1036E-DMS2618E
FPL1038E-DMS2620E
FPL1039E-DMS2621E
FPL1040E-DMS2622E
FPL1041E-DMS2623E
FPL1049E-DMS2624E
FPL1100E-DMS2670E
FPL1110I-DMS2677I
FPL1111I-DMS2700I
FPL1112I-DMS2979I
FPL1113I-DMS2981I
FPL1114I-DMS2986I
FPL1115I-DMS2987I
FPL1124E-DMS2628E

Glossary

A list of VM/ESA terms and their definitions is available through the online VM/ESA 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 *VM/ESA: CMS User’s Guide*; for more about the HELP command, see the *VM/ESA: CMS Command Reference*.

You can find additional information about IBM terminology in the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

Bibliography

This bibliography lists the publications that provide information about your VM/ESA system. The VM/ESA library includes VM/ESA base publications, publications for additional facilities included with VM/ESA, and publications for VM/ESA optional features.

VM/ESA publications may be available as Adobe Portable Document Format (PDF) files, IBM BookManager® files, or printed books. For abstracts of VM/ESA publications and other library-related information, including current editions and available publication formats, see *VM/ESA: General Information*.

VM/ESA Base Publications

Evaluation

VM/ESA: Licensed Program Specifications, GC24-5744

VM/ESA: General Information, GC24-5745

Installation and Service

VM/ESA: Installation Guide, GC24-5836

VM/ESA: Service Guide, GC24-5838

VM/ESA: VMSES/E Introduction and Reference, GC24-5837

Planning and Administration

VM/ESA: Planning and Administration, SC24-5750

VM/ESA: CMS File Pool Planning, Administration, and Operation, SC24-5751

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

VM/ESA: REXX/EXEC Migration Tool for VM/ESA, GC24-5752

VM/ESA: Running Guest Operating Systems, SC24-5755

VM/ESA: Connectivity Planning, Administration, and Operation, SC24-5756

VM/ESA: Group Control System, SC24-5757

VM/ESA: Performance, SC24-5782

Customization

IBM VM/ESA: CP Exit Customization, SC24-5672

Operation

VM/ESA: System Operation, SC24-5758

VM/ESA: Virtual Machine Operation, SC24-5759

Application Programming

VM/ESA: CP Programming Services, SC24-5760

VM/ESA: CMS Application Development Guide, SC24-5761

VM/ESA: CMS Application Development Reference, SC24-5762

VM/ESA: CMS Application Development Guide for Assembler, SC24-5763

VM/ESA: CMS Application Development Reference for Assembler, SC24-5764

VM/ESA: CMS Application Multitasking, SC24-5766

VM/ESA: REXX/VM Primer, SC24-5598

VM/ESA: REXX/VM User's Guide, SC24-5465

VM/ESA: REXX/VM Reference, SC24-5770

IBM VM/ESA: Distributed Graphical User Interface Toolkit, SC24-5724

IBM VM/ESA: Reusable Server Kernel Programmer's Guide and Reference, SC24-5852

VM/ESA: Enterprise Systems Architecture/Extended Configuration Principles of Operation, SC24-5594

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

External Security Interface (RACROUTE) Macro Reference for MVS and VM, GC28-1366

End Use

VM/ESA: CP Command and Utility Reference, SC24-5773

VM/ESA: CMS Primer, SC24-5458

VM/ESA: CMS User's Guide, SC24-5775

VM/ESA: CMS Command Reference, SC24-5776

IBM VM/ESA: Graphical User Interface Facility, SC24-5789

VM/ESA: CMS Pipelines User's Guide, SC24-5777

VM/ESA: CMS Pipelines Reference, SC24-5778

CMS/TSO Pipelines: Author's Edition, SL26-0018

VM/ESA: XEDIT User's Guide, SC24-5779

VM/ESA: XEDIT Command and Macro Reference, SC24-5780

VM/ESA: Quick Reference, SX24-5290

Diagnosis

VM/ESA: System Messages and Codes, GC24-5841

VM/ESA: Dump Viewing Facility, GC24-5853

VM/ESA: Diagnosis Guide, GC24-5854

VM/ESA: CP Diagnosis Reference, SC24-5855

VM/ESA: CP Diagnosis Reference Summary, SX24-5292

VM/ESA: CMS Diagnosis Reference, SC24-5857

Note: CP and CMS control block information is not provided in book form. This information is available on the IBM VM/ESA operating system home page (http://www.ibm.com/s390/vm).

Publications for Additional Facilities

OpenEdition® for VM/ESA

IBM OpenEdition for VM/ESA: POSIX Conformance Document, GC24-5842

IBM OpenEdition for VM/ESA: User's Guide, SC24-5727

IBM OpenEdition for VM/ESA: Command Reference, SC24-5728

IBM OpenEdition for VM/ESA: Advanced Application Programming Tools, SC24-5729

IBM OpenEdition for VM/ESA: Callable Services Reference, SC24-5726

IBM OpenEdition for VM/ESA: Sockets Reference, SC24-5741

IBM C for VM/ESA: Library Reference, SC23-3908

Debug Tool User's Guide and Reference, SC09-2137

DFSMS/VM®

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

S/390® Open Systems Adapter Support Facility for VM/ESA

Planning for the System/390 Open Systems Adapter Feature, GC23-3870

IBM VM/ESA: Open Systems Adapter Support Facility User's Guide, SC28-1992

Language Environment®

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 VM/ESA

VM/ESA: TCP/IP Function Level 320 Planning and Customization, SC24-5847

VM/ESA: TCP/IP Function Level 320 User's Guide, SC24-5848

VM/ESA: TCP/IP Function Level 320 Programmer's Reference, SC24-5849

VM/ESA: TCP/IP Function Level 320 Messages and Codes, GC24-5850

VM/ESA: TCP/IP Function Level 320 Diagnosis Guide, GC24-5851

OpenEdition Distributed Computing Environment Feature for VM/ESA

IBM OpenEdition DCE for VM/ESA: Introducing the OpenEdition Distributed Computing Environment, SC24-5735

IBM OpenEdition DCE for VM/ESA: Planning, SC24-5737

IBM OpenEdition DCE for VM/ESA: Configuring and Getting Started, SC24-5734

IBM OpenEdition DCE for VM/ESA: Administration Guide, SC24-5730

IBM OpenEdition DCE for VM/ESA: Administration Reference, SC24-5731

IBM OpenEdition DCE for VM/ESA: Application Development Guide, SC24-5732

IBM OpenEdition DCE for VM/ESA: Application Development Reference, SC24-5733

IBM OpenEdition DCE for VM/ESA: User's Guide, SC24-5738

IBM OpenEdition DCE for VM/ESA: Messages and Codes, SC24-5736

LAN File Services/ESA

Discovering LAN File Services/ESA, GK2T-5762

Introducing LAN File Services/ESA, GH24-5259

LAN File Services/ESA: Licensed Program Specifications, GH24-5260

LAN File Services/ESA: VM Guide and Reference, SH24-5264

LAN Resource Extension and Services/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 PDF versions of many VM/ESA publications and publications for some related IBM licensed programs. It also 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/ESA.

IBM Online Library Omnibus Edition: VM Collection, SK2T-2067

Note: Only unlicensed publications are included.

Index

Readers' Comments

**Virtual Machine/
Enterprise Systems Architecture
Conversion Guide and Notebook
Version 2 Release 4.0**

Publication No. GC24-5839-01

You may use this form to report errors, to suggest improvements, or to express your opinion on the appearance, organization, or completeness of this book.

Date: ___________________________________

____

____

____

____

____

____

____

____

____

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

Note

Report system problems to your IBM representative or the IBM branch office serving you.
U.S. customers can order publications by calling the IBM Software Manufacturing Solutions at **1-800-879-2755**.

In addition to using this postage-paid form, you may send your comments by:

FAX	1-607-752-2327	Internet	pubrcf@vnet.ibm.com
IBM Mail	USIB2L8Z@IBMMAIL	IBMLink	GDLVME(PUBRCF)

Would you like a reply? ___**YES** ___ **NO** If yes, please tell us the type of response you prefer.

___ Electronic address: ____

___ FAX number: ___________________________________

___ Mail: (Please fill in your name and address below.)

_____________________________ _____________________________
Name Address

_____________________________ _____________________________
Company or Organization

_____________________________ _____________________________
Phone No.

Readers' Comments
GC24-5839-01

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

Department G60
International Business Machines Corporation
Information Development
1701 North Street
ENDICOTT NY 13760-5553

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

File Number: S370/S390-34
Program Number: 5654-030

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

GC24-5839-01

Spine information:

IBM VM/ESA Conversion Guide and Notebook *Version 2 Release 4.0*