

VM/Pass-Through Facility



VM/Pass-Through Facility

2.1.1

Administration and Operation

**Note!**

Before using this information and the product it supports, read the information in "Notices" on page 349.

This edition, SC24-5557-02, applies to the VM/Pass-Through Facility, Version 2, Release 1.1, program number 5684-100, and to all subsequent releases of this product until otherwise indicated

Last updated: January 29, 2024

© **Copyright International Business Machines Corporation 1991, 2024. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Preface</b>	xiii
What this Book Contains	xiii
How to Use this Book	xiii
 <b>Summary of Changes</b>	xv
Changes introduced with APAR VM66733 (February 2024):	xv
 <b>Chapter 1. Introduction</b>	1
What is PVM?	1
How PVM Enhances A VM System	1
Networking Terms	3
What Users Can Access Through PVM	3
How Users Can Access PVM	4
 <b>Chapter 2. CVIEW - Product Overview</b>	5
What is CVIEW?	5
Modes of Operation	6
Setup Mode	6
Session Mode	7
Notify Mode	8
Control Mode	8
Technical Overview	8
Accessing the CVIEW Virtual Machine	9
Networking Sessions	9
Networking Considerations	11
PASSTHRU Considerations	11
Display-input Option	11
Accommodating Different Display Station Types	12
Messages	12
 <b>Part 1. Administering, Installing, and Operating PVM</b>	13
 <b>Chapter 3. Administrative Considerations</b>	15
Installation Requirements	15
Storage Considerations	15
DASD Requirements	15
Storage Needs in the Default Configuration	15
Modifying PVM Storage Needs	16
How Much Storage Is Needed in Each User's Virtual Machine for MPVM?	16
How Much Storage Is Needed for a PVM Server?	18
How Much Storage is Need for a PVMG Server?	18
Performance Considerations	19
Security Considerations	19
Notepad Facility	19
Session Security	19
Restrictions	20
General Restrictions	20
Access Security Exits Restrictions	21
PCCF Restrictions	21
Channel to Channel Adapter (CTCA) Restrictions	21

Logical Device Support Restrictions . . . . .	21
<b>Chapter 4. CVIEW Installation Planning</b> . . . . .	23
CVIEW Virtual Machine Requirements . . . . .	23
Disk Space . . . . .	23
Virtual Storage . . . . .	23
Privilege Class . . . . .	23
Channel Mode and Nonshared Virtual Control Units . . . . .	23
User ID . . . . .	23
CVIEW Directory Entry . . . . .	23
CVIEW Operating Environment . . . . .	24
Machine Requirements . . . . .	24
Multiple CVIEW Virtual Machines . . . . .	25
<b>Chapter 5. Defining A PVM Configuration</b> . . . . .	27
Planning a Network . . . . .	27
Creating the Configuration File . . . . .	28
Sample Configuration Files . . . . .	28
Commenting the PVM Configuration File . . . . .	30
Identifying the System Language . . . . .	30
Excluding Double-Byte Character Messages . . . . .	31
Identifying the Local Node Name . . . . .	31
Identifying a Unique Resource Name . . . . .	31
Defining a Link from a Local Node to Another Node . . . . .	31
Defining Groups . . . . .	32
Defining Routes and Alternate Routes . . . . .	32
Authorizing the PVM Operator . . . . .	33
Identifying CMS Subset Mode Commands to PVM . . . . .	33
Automatically Connecting Users to a Node . . . . .	33
Setting PF Keys for PVM Menus . . . . .	34
Saving PVM Information . . . . .	34
Specifying Restrictions . . . . .	35
Specifying Message Requirements . . . . .	36
Specifying Storage Requirements . . . . .	36
Creating a CMS PROFILE EXEC for PVM . . . . .	36
Creating a CMS Communications Directory . . . . .	37
Creating a PVM Profile . . . . .	38
<b>Chapter 6. Defining A PVMG Configuration</b> . . . . .	41
Planning A PVMG Configuration . . . . .	41
Planning the PVMG Server's Role in the PVM Network . . . . .	42
Commenting the PVMG Configuration File . . . . .	44
Identifying the System Language . . . . .	45
Excluding Double-Byte Character Messages . . . . .	45
Identifying the Local Node Name and the VTAM Application Name . . . . .	45
Defining a Link from PVMG to PVM . . . . .	45
Defining the VTAM APPLID to Which Users Log on . . . . .	45
Assigning Node Names to PF Keys . . . . .	46
Defining Routes and Alternate Routes . . . . .	46
Specifying LU Names Used by PVMG with Secure VTAM APPLIDs . . . . .	46
Authorizing PVMG Operators . . . . .	46
Saving PVMG Information . . . . .	47
Specifying Message Requirements . . . . .	47
Specifying the Target User ID for Natural Language Message Reception . . . . .	47

Specifying the User ID of the Programmed Operator Virtual Machine That Receives Encoded Messages	47
Specifying the VTAM APPLID to Disconnect When PVMG Fails	47
Changing the Default Local PVMG Node Session Termination String	48
Specifying VTAM Logon Model Table Entries Used by PVMG	48
Tailoring the PROFILE GCS File	48
<b>Chapter 7. Developing NAMES Files for MPVM</b>	49
Creating Entries for Applications	49
Specifying the Nickname	49
Specifying Characteristics for Application Nicknames	50
Defining Target Applications	50
Defining Other Variables for Applications	50
Specifying Commands to be Invoked	50
Specifying custom Subtags	51
Specifying Nickname Lists	51
Summary of Application NAMES File Entries	51
Defining Configuration Settings—:nick.?CONFIG	52
User Initialization Settings	52
Creating Other Special Entries	52
Setting Global Values for All Sessions—:nick.?INITIAL	52
Setting Default Values for All Sessions—:nick.?DEFAULT	52
Cleaning Up After the Last Session Has Ended—:nick.?FINAL	53
Sample MPVMSYS NAMES File	53
Verifying the MPVMSYS NAMES File	54
Considerations for Individual User NAMES Files	54
Summary of MPVMSYS NAMES Contents	55
<b>Chapter 8. Operating the Network</b>	57
Starting PVM	57
Querying the System	61
Defining a Link Dynamically	63
Displaying the Path of a Session	64
Sending Messages	65
Issuing CMS Subset Mode Commands	65
Quiescing, Resuming, and Dropping Links, Devices, and Users	66
Using the PVM Programmable Operator (PVMPROP)/Command Response Interface (CRI)	68
Stopping PVM	69
Starting PVMG	69
Querying PVMG	69
Querying the System	69
Monitoring User Activity	70
Displaying the Path of a PVMG Session	70
Sending Messages	71
Dropping Users and Nodes	71
Stopping PVMG	73
<b>Chapter 9. Using CVIEW</b>	75
Starting the CVIEW Virtual Machine	75
Automatically Starting CVIEW	75
Manually Starting CVIEW	75
CVIEW Commands	76
CMS	76
CP	77

DROP	77
HELP	78
PASSWORD	78
QUERY	79
SHUTDOWN	81
STOP	82
Establishing a Session	82
Managing a CVIEW Session	85
Consult Sessions	85
Conference Sessions	86
Device Type and Model Assignment	88
Joining a Conference Session in Progress	88
CVIEW PROFILE EXEC	89
CVIEW Log File	89

---

## Part 2. Determining Problems . . . . . 91

<b>Chapter 10. Collecting Problem Information</b>	93
Trace Facility for the PVM Component	94
System Services Tracing	94
TRACE USER Command	94
Tracing Line Driver I/O Activity	95
Line Driver Data Transfer Tracing	96
PCCF Tracing	96
Tracing System Activity	96
The Dump Facility	97
Full Dumps	97
Partial Dumps	98
Analyzing Dump Information	98
Other PVM Problem Determination Actions	98
Tracing MPVM	98
TRACEn Subcommand	100
Logging Messages and Screen Images	101
Diagnosing PVMG Problems	101
PVMG Trace Facility	101
Tracing PVMG System Activity	102
PVMG I/O Tracing	102
Other PVMG Problem Determination Actions	102
The PVMG Dump Facility	102
Analyzing PVMG Dump Information	103

---

## Part 3. Modifying PVM . . . . . 105

<b>Chapter 11. Tuning PVM Performance</b>	107
Enhancing PVM Performance	107
SET RESERVE Command	107
Dispatching Priority	108
Installing MPVM in a Shared Segment	108
MPVM Shared Segment Verification	109
Changing the Names of MPVM Shared Segments	109
Adding Common MPVM Commands and Macros to an Installation Shared Segment	110
Saving MPVM Common Disk Information in a Segment	110

Fine-Tuning the PVM Server	111
Checking Response Time for User Commands	111
Customizing MPVM Sessions	111
Sample MPVM Exits	112
Developing Exits	114
Programming Note	115
PVM Macros Used in the Sample Access Security Exits	115
DVMENR	116
DVMRETN	116
DVMXTRN	117
Session Control Exits	117
Session Initiation Exit (DVMNUE1)	117
Session Termination Exit (DVMNUE2)	118
Session Request Exits (DVMNUE3 and DVMNUE4)	118
Command and Response Exit (DVMNUE5)	120
CINIT Session Initiation Data Exit (DVMNUE6)	121
Network Management Exits	122
Link Statistics Exit (DVMNMEL)	122
Session Statistics Exit (DVMNMES)	124
CMS User Exits	124
Before Initialization Exit (DVMCUEBI)	125
After Initialization Exit (DVMCUEAI)	125
Before Read Exit (DVMCUEBR)	126
After Read Exit (DVMCUEAR)	126
Before Write Exit (DVMCUEBW)	127
After Write Exit (DVMCUEAW)	127
Before Final Exit (DVMCUEBF)	128
Rebuilding PVM for Exits	128
<b>Chapter 12. National Languages on the PVM System</b>	<b>129</b>
National Language Overview	129
Languages Available Through IBM	129
Backup National Language	129
Changing the Backup Language	130
Changing Message Destinations	130
Changing the PVM Translate Table	131
Creating and Using a Message Repository	132
Creating a Message File	132
Compiling the Message File	133
Verifying the Message File	133
Making the Message File Available	133
Rules for Making a custom Repository	134
Comment Record	134
Master Record	134
Time Stamp Records	135
Message and Screen Records	137
Substitution Variables in Messages	139
Special Considerations for Message and Screen Records	139
Control Lines	141
Message Repository Samples	141
<b>Chapter 13. Facilities for Automating PVM and PVMG Operations</b>	<b>145</b>
PVMPROP - A Programmable Operator Interface for PVM and PVMG	145
The Command Response Interface	146

PVMPROP/CRI Message Protocol . . . . .	146
Message Format . . . . .	147
Deciphering Encoded Messages . . . . .	147
Using Command Prefixes . . . . .	149
Sample Application . . . . .	151
 <b>Chapter 14. TCP/IP Line Driver Overview</b> . . . . .	153
The PVM TCP/IP Redirector Task . . . . .	154
Redirector Task Error Processing . . . . .	154
The PVM TCP/IP Line Driver Task . . . . .	154
PVM TCP/IP Line Driver Error Processing . . . . .	154
A Word About Sockets . . . . .	155
Tracing Considerations . . . . .	155
Configuration File Example . . . . .	155
 <b>Chapter 15. Application-to-Application Overview</b> . . . . .	159
PCCF Data Flow . . . . .	159
Data Transmission . . . . .	160
Session Termination . . . . .	160
PCCF Application-to-Application Restrictions . . . . .	161
Application-to-Application Communication Sample Program . . . . .	161
 <b>Chapter 16. Customizing CVIEW</b> . . . . .	165
Listed Consultants — CONSULT IDNAME . . . . .	165
Defining Listed Consultants . . . . .	165
Program Function Key Assignments — DXGKEYS LIST . . . . .	166
PF Key Functions . . . . .	166
Changing PF Key Assignments . . . . .	166
The Customizing EXEC — DXGINIT . . . . .	167
Concurrent Users . . . . .	167
The NOTIFY Panel . . . . .	167
Privileged Users . . . . .	168
Executing the Customization EXEC . . . . .	168
 <b>Part 4. Reference Information</b> . . . . .	171
 <b>Appendix A. PVM Macros Supported as Programming Interfaces</b> . . . . .	175
Product-Sensitive Programming Interface Executable Macros . . . . .	175
 <b>Appendix B. PVM Commands</b> . . . . .	177
How to Read the Syntax Diagrams . . . . .	177
* (Comment) . . . . .	182
AUDIT . . . . .	182
AUTHORIZ . . . . .	183
CDBC . . . . .	185
CMD . . . . .	186
CMS . . . . .	186
CP . . . . .	187
CPQ . . . . .	188
DEFINE . . . . .	189
CTCA Format . . . . .	189
APPC Format . . . . .	190
IUCV Format . . . . .	190



TCPIP Format	191
3088 Format	191
DISC	194
DROP	195
EXEC	196
LMSG	197
LOGMSG	198
MPVM	199
MSG	202
PASSTHRU	203
PAUSE	207
PMSG	208
PVMBLD	209
PVMCMP	210
QUERY	211
QUIESCE	214
QUIT	217
RESUME	217
ROUTE	218
RUNPVM	219
SELECT	220
SET	220
SHOW	222
SHUTDOWN	223
START	224
STATUS	225
TCP	227
TRACE	228
VARY	230
CP Commands Used with PVM	231
DIAL	231
SMSG	232
<b>Appendix C. PVM Configuration File Records</b>	<b>233</b>
* (Comment)	233
AUDIT	233
AUTHORIZ	235
AUTO	235
BUFFERS	236
DUMP	238
EXT	238
GROUP	239
I/O	240
LANG	241
LINK	242
LNODE	244
LOCAL	245
MAXCONN	245
MAXIDLE	246
MAXSESS	246
MAXSUSP	246
MAXTDISC	247
MAXUSERS	248
MSGNOH	248

MSGOPTNS	248
OPID	250
PVMPROP	250
RESID	252
ROUTE	252
SELECT	253
SESSTERM	253
STATTIME	254
SUBSET	254
TCP	255
TDISC	255
TIMEOUT	256
XDBC	256
<b>Appendix D. PVMG Commands</b>	<b>259</b>
How to Read the Syntax Diagrams	260
AUTHORIZ	262
CDBC	264
CMD	265
DISC	266
DROP	266
INIT	267
LMSG	268
LOGMSG	269
MSG	269
PMSG	270
QUERY	271
QUIESCE	273
RESUME	274
ROUTE	274
RUNPVMG	275
SELECT	276
SHOW	276
SHUTDOWN	277
SPOOL	279
START	280
TRACE	281
CP Command Used with PVMG	282
SMSG	282
<b>Appendix E. PVMG Configuration File Records</b>	<b>283</b>
*	283
APPLN	284
AUTHORIZ	285
DUMP	285
LANG	286
LINK	286
LNODE	287
LOCAL	287
LOGMODE	288
LOGONID	289
LU	290
MSGNOH	291
MSGOPTNS	292

OPID	293
PVMPROP	294
ROUTE	294
SELECT	295
SESSDISC	295
SESSTERM	296
XDBC	296
<b>Appendix F. Setting Up an APPC Link</b>	<b>299</b>
<b>Appendix G. PVM Control Block Formats</b>	<b>301</b>
<b>Appendix H. PVM Access Security Exits</b>	<b>319</b>
The REMOTE AUTH File	319
Using Generics	320
Reloading the REMOTE AUTH File	320
Definitions of the REMOTE AUTH File Entries	321
GATELINK	321
REM-CMD	322
REM+USER	323
SNA-USER	324
VM-USER	325
Messages	327
Messages Returned to the User:	327
Messages Returned to the PVM Console	327
Format of SECAUDIT LOG Records	328
Controlling Local Users Using the REMOTE AUTH Table	329
DVMNUE3 Testing	330
SECAUDIT LOG	331
Controlling Remote Users by Using the REMOTE AUTH Table	332
DVMNUE4 Testing	333
SECAUDIT LOG	334
<b>Appendix I. CVIEW Product Panels</b>	<b>335</b>
Setup-mode Panels	335
Consult Session Setup (Recipient)	336
Consult Session Setup (Consultant)	338
Conference Session Setup (Attendee)	340
Conference Session Setup (Moderator)	342
Notify-mode Panels	343
Control-mode Panels	345
Consultant Control	345
Moderator Control	346
<b>Notices</b>	<b>349</b>
Programming Interface Information	350
Trademarks	350
Terms and Conditions for Product Documentation	350
Applicability	351
Personal Use	351
Commercial Use	351
Rights	351
IBM Online Privacy Statement	351

<b>Bibliography</b> . . . . .	353
VM/Pass-Through Facility Library . . . . .	353
RSCS Library . . . . .	353
z/VM Release 2.1 Library . . . . .	354
Related Books . . . . .	355
 <b>Index</b> . . . . .	 357

---

## Preface

This book is for anyone that has to use or maintain VM/Pass-Through Facility (PVM) on a VM system. This book covers the information you need to install, operate, and manage PVM.

General users should refer to the *VM/Pass-Through Facility: User's Guide* for information about using PVM.

**In this book, VM/Pass-Through Facility Version 2 is referred to as PVM. VTAM refers to the Virtual Telecommunications Access Method Version 3 Release 2 or later. The VM/Pass-Through Facility Gateway component is referred to as PVMG, and PVM's simultaneous multiple session component, the VM/Pass-Through Facility Multiple Session Application, is referred to as MPVM.**

A PVM system, or PVM network, can, but does not have to, include PVM's PVMG, and MPVM components. This book is arranged by groups of tasks to help make PVM easier to use.

## What this Book Contains

The main parts of this book are:

### **Part 1, “Administering, Installing, and Operating PVM”**

PVM is a program that runs on a VM system. This program defines a network that allows system users to access other systems. Read this part if you have to plan, install, define, or maintain a PVM system.

### **Part 2, “Determining Problems”**

This part helps you determine if a problem exists in your PVM system. It informs you how to start and keep PVM running so people can use the PVM functions.

### **Part 3, “Modifying PVM”**

System programmers can customize PVM to extend the functions of PVM. This part lists parameters programmers need to be aware of in order to collect data from PVM. Refer to this part if you want to analyze network activity or to run test systems.

### **Part 4, “Reference Information”**

Once you are using or operating PVM, you may need to remind yourself of a particular item. This part includes appendix references to these topics:

- PVM macros supported as programming interfaces
- Commands and configuration file records for PVM
- Commands and configuration file records for PVMG
- Setting up an APPC link
- PVM control block formats
- PVM sample access security exits
- CVIEW product panels

## How to Use this Book

Color is used in this book to illustrate:

- Words or commands you type on your keyboard.
- Part or all of what you see on your display station's screen.

Syntax diagrams are used to describe command syntax and configuration file records. See “How to Read the Syntax Diagrams” on page 177 for instructions on reading syntax diagrams.

PVM users fit into the following categories:

- Administrator
- Operator
- System Programmer
- General user.

This book will help you better if you can place yourself, or what you are trying to do while using PVM, into one of these categories.

**Administrators** plan a PVM network. They assure that the general users are satisfied with the services they get from PVM. Some administrators are also responsible for installing and servicing the PVM product. Administrators may be responsible for recognizing problem conditions. Administrators should review Part 1, “Administering, Installing, and Operating PVM,” and Part 2, “Determining Problems.”

**Operators** manage PVM and its components. Operators perform the daily tasks that keep PVM running. Operators can start PVM, enable users, and stop PVM. They can add systems or delete systems from a PVM network. Operators need to recognize problem conditions. Depending on the installation, both the operator and the administrator for PVM may be the same person. Operators need to review Part 1, “Administering, Installing, and Operating PVM.”

**System programmers** customize PVM for specific needs of the company or department. PVM provides ways to add routines to the product to collect data or run tests. Programmers may be called in to analyze problem conditions. PVM has facilities to look at storage or trace network activity. System programmers can assist in keeping PVM available for general users and operators. Part 3, “Modifying PVM,” is written for system programmers.

**System programmers** who wish to take advantage of the multiple session programmability of PVM's MPVM component should also consult the *VM/Pass-Through Facility: Programming Reference*, and the *VM/Pass-Through Facility: Messages* .

**General users** should consult the *VM/Pass-Through Facility: User's Guide* for information about general user tasks.

---

## Summary of Changes

This section describes the changes in this book.

---

### Changes introduced with APAR VM66733 (February 2024):

- The PVM server no longer requires **SET 370ACCOM ON**. The PVM server will execute in a **MACHINE ESA** mode virtual machine.
- The PVM server will run in **either** CMS or zCMS environments.
- Support for the following devices and functions have been removed, including the ability to configure them:
  1. BSCA
  2. ISFC (CSECOM)
  3. R3270
  4. ROCF
  5. S3270
  6. 3088 DIAG/NODIAG options
  7. 327X
  8. 328X
  9. TRACE command: System Trace File Interface (Diagnose X'E0') support

PVM continues to support the following connectivity configurations:

1. CTCA: Channel-to-channel adapters
2. TCPIP: Secure or non-secure TCPIP
3. APPC: Advanced Program-to-Program Communications
4. 3088: Via FICON,, adapters
5. IUCV: Inter User Communication Vehicle (IUCV) to communicate with a PVM to VTAM gateway (PVMG) server
6. PCCF: For cross-systems IUCV communications
7. GRAF: Graphics for direct-attached display stations

Multiple Pass-Through VM (MPVM), CMS PASSTHRU, and Cooperative Viewing Facility (CVIEW) continue to be supported environments.





---

## Chapter 1. Introduction

This chapter introduces the VM/Pass-Through Facility and its components.

---

### What is PVM?

VM/Pass-Through Facility (PVM) is a communications program used by VM users to access other systems. PVM lets users access, log on, and use another system in defined PVM and System Network Architecture\* (SNA\*) networks as though their display station is directly connected to that system. All they need is a user ID (*ID* is short for identifier) for the system they want to access. A system accessed through PVM is called a **target** system. Once a user is logged on to a target system, they may communicate with that system like any other user local to that system.

PVM also provides support for application-to-application communications between VM host applications and other VM host applications. Each VM host application that participates in an application session must have access to a PVM Version 2 virtual machine.

PVM runs in its own Conversational Monitor System (CMS) virtual machine. As with other virtual machines, PVM is identified by a **user ID**. Because user IDs for PVM can be different from system to system, the term **PVM user ID** refers to a PVM virtual machine.

PVM has four major components:

<b>PVM</b>	The PVM component is the server that allows users to pass through to other applications and systems.
<b>PASSTHRU</b>	PASSTHRU provides single session support for CMS users.
<b>PVMG</b>	The PVMG server runs on its own disconnected virtual machine. PVMG is a Virtual Telecommunications Access Method (VTAM*) application, running on VM's Group Control System (GCS) component, that serves as a communications gateway linking the PVM server machine and the VTAM System Network Architecture (SNA) network. Do not install this server if PVM access to SNA users and applications is not required.
<b>MPVM</b>	MPVM provides simultaneous multiple session support for CMS users.  MPVM runs in the CMS end user's virtual machine and supports automation of CMS end-user session activity. It processes MPVM macros written by CMS users and MPVM programmers. It also processes system nickname files that define network connection parameters.

---

### How PVM Enhances A VM System

PVM enhances a VM system by providing:

- **Multiple, programmable, simultaneous session capability**

The MPVM component of PVM allows CMS users to have multiple interactive sessions with different applications at the same time, plus have access to their own host user ID, all from a single terminal. These applications can be on:

- The same VM system
- Another VM system
- A non-VM system that supports 3270 architecture.

## How PVM Enhances A VM System

While using an application through MPVM, the CMS users are not aware of any MPVM activities. It appears that they have directly accessed the application.

CMS users can use MPVM commands to start sessions with different applications. They may also use a toggle key to switch control to any session they have already started.

Experienced 3270 programmers can write programs using MPVM commands and macros that automate user tasks. For example, an application can be written that simulates a user starting an application, typing data into fields, and then pressing the ENTER key. Many things that a user can do manually at a keyboard can be automated by MPVM.

Others macros can be written that:

- Integrate data from several applications onto a user's single display
- Control how information is transferred from a user's display to an application.

See the *VM/Pass-Through Facility: Programming Reference* for more information about MPVM.

- **PVMG provides a gateway to or from the SNA network**

PVM users, and users in the SNA network, can access applications through the PVMG server.

- **Application-to-application communications using IUCV protocols**

PVM supports application-to-application communication through the PVM network between VM host applications. This is referred to as PVM's PCCF support.

- **Low cost networking and concurrent access to multiple systems**

PVM makes full use of networks from one display station. PVM increases productivity by providing:

- Reliable access to many remote systems including other VM systems with PVM installed.
- Remote access to applications, like CICS
- Reliable communication across many types of connections, such as connections using a 3088 Multisystem Channel Communications Unit. Using Advanced Program-to-Program Communication/VM (APPC/VM) and Advanced Communications Function for Virtual Telecommunications Access Method\* (ACF/VTAM\*).

- **Ease of installation**

After a few changes to the system directory and the system's MAINT user ID, PVM is ready to install. VMSES/E is supported for installing and servicing on a z/VM system.

- **Ease of operation and use**

PVM operation is automated by using execs that set up the PVM system to operate the way it has been configured.

PVM is transparent to its users after the PVM session starts. CMS users of PVM can set up their own defaults for the PASSTHRU or MPVM commands, making it easier to start PVM sessions with the options they want.

General users may want to read the *VM/Pass-Through Facility: User's Guide* before using this book. For many tasks, the *VM/Pass-Through Facility: User's Guide* contains the information necessary to perform the desired task.

## Networking Terms

This section explains some networking terms and their relationship to PVM. Networks consist of systems and Input/Output (I/O) devices connected by communications equipment. In a network, these systems and devices are called **nodes**. The connections between the nodes in a network are called **links**.

Nodes and links are given names to identify them to users and operators. The terms **node ID** and **link ID** indicate a node's name or a link's name.

The term **local** means the system to which a user is directly connected to at a particular time. The term **remote** generally describes a system being accessed via a connection made through communications equipment over PVM-defined links. A **route** is the path from one node in the network to another. An **intermediate** node is a node that is between nodes in a route.

Generally, the link ID is the same as the node ID at the other end of the link. Each link includes a programming routine, called a **line driver**, that controls the communications between the nodes on the link. Line drivers use a set of rules, called a **protocol**, to manage communications. There is a different line driver for each protocol that PVM supports. The line driver used for a particular link depends on the hardware and protocol used on the node at the other end of the link.

---

## What Users Can Access Through PVM

PVM users can access:

- Remote applications, like CICS
- Other VM systems with PVM installed
- The VM system on which PVM is installed
- Applications in the SNA network.

Line drivers are used to manage the links to remote systems in a PVM network. Table 1 describes the types of line drivers in PVM.

*Table 1. PVM Line Drivers*

Type	Type of Connection	Provides Communications with
<b>CTCA</b>	Channel-to-channel adapter (CTCA)	Another VM system with PVM installed. The target system's PVM uses the CP Logical Device Support Facility to establish communications. This CP facility is described in <i>z/VM: CP Programming Services, SC24-6272</i> .
<b>3088</b>	3088 Multisystem Channel Communications Unit	Another VM system with PVM installed. This line driver supports FICON channels.
<b>APPC</b>	APPC/VM protocol over a logical path	Another VM system with PVM installed over ACF/VTAM* controlled links.
<b>IUCV</b>	IUCV line driver connection	A PVMG server on the same VM system as PVM.

---

### How Users Can Access PVM

The following are ways to access PVM:

1. Users can access PVM by connecting to the PVM server using the z/VM CP DIAL command.
2. Users can also access PVM by using the PASSTHRU command from a CMS environment on a VM system that has PVM installed on it.
3. CMS users can access PVM using MPVM, which allows them to have multiple simultaneous sessions through PVM.
4. PVM users can access SNA applications through the PVMG server, and users in the SNA network can also access the PVM network through the PVMG server.

Any PVM user can access an application in the SNA network, and any SNA user can access applications in the PVM network through the PVMG server.

An application may access PVM using IUCV protocols to communicate with itself, another virtual machine application on the local system, or another virtual machine application on any PVM node in the network. Intermediate PVM nodes need not have this support.

Consult the *VM/Pass-Through Facility: User's Guide* for instructions on how to access PVM.

---

## Chapter 2. CVIEW - Product Overview

This chapter is a general introduction to the Cooperative Viewing Facility (CVIEW).

---

### What is CVIEW?

CVIEW is a VM based facility that allows two to twelve users, at separate display stations, to view and share a single VM **interactive session**. An interactive session consists of the activities that occur from the time a user logs on until he logs off. A CVIEW session can be established as either a **consult** or **conference** session. Multiple consult and conference sessions can process at the same time, supporting a total of thirty users at separate display stations.

Users participating in a CVIEW session define their roles and identify the session during “setup mode.” Menu and data-entry panels guide them through the setup process. Each menu and data-entry panel has an associated help panel that provides additional information.

A **consult session** supports two users at separate display stations. One user is designated as the **consultant** and the other user is the **recipient**. The consultant is an expert in a particular subject area and provides help or assistance to the recipient. In a consult session, the keyboards of both users are “active,” allowing either user to enter data or commands.

A **conference session** supports from two to twelve users at separate display stations. During setup mode, one user is designated as the **moderator** and all other users participating in the conference are **attendees**. The moderator has an active keyboard and control the conference session. Attendees have “inactive” keyboards so they cannot enter data or commands. The moderator can activate an attendee's keyboard by assigning **presenter** status to him. The moderator can drop participants from the conference, and attendees can remove themselves from the conference.

In a CVIEW session, users view identical screen images at the same time. The output from host and application processing is always displayed. CVIEW also allows the display of user input before the input is processed by the host or application. This **display-input** option is activated by the consultant and the moderator. After the display-input option is activated, any user with an active keyboard can “direct” his input to all display stations in the session.

Figure 1 on page 6 shows how display stations in consult and conference sessions relate to each other and to the CVIEW virtual machine during an interactive session.

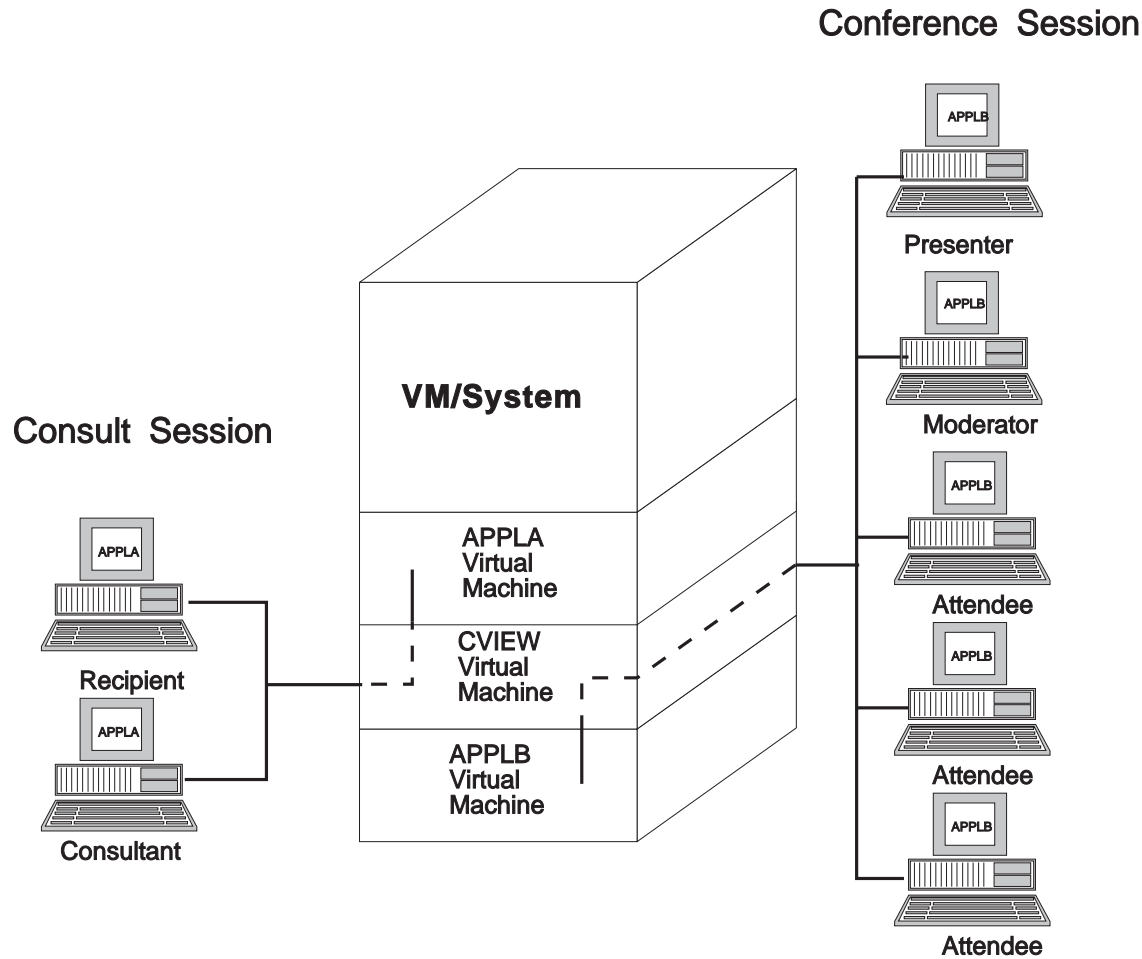


Figure 1. CVIEW Sessions and the CVIEW Virtual Machine

## Modes of Operation

This section describes the different modes of operation for CVIEW.

### Setup Mode

**Setup mode** starts after a user enters 'DIAL CVIEW' from his display station. The first panel, the **HOME** panel, is displayed to the user, indicating the beginning of setup mode. The user selects his role (recipient, consultant, moderator, or attendee) for the session from the HOME panel. See Appendix I, "CVIEW Product Panels" on page 335 for descriptions of all the CVIEW panels.

As setup mode continues, additional panels are displayed to the user to further define the session in which he will participate. For a consult session, setup is complete when the recipient and consultant have responded to all the setup panels required to define the session. For a conference session, setup is complete when the moderator starts the conference session. When setup mode is complete, the CVIEW session starts.

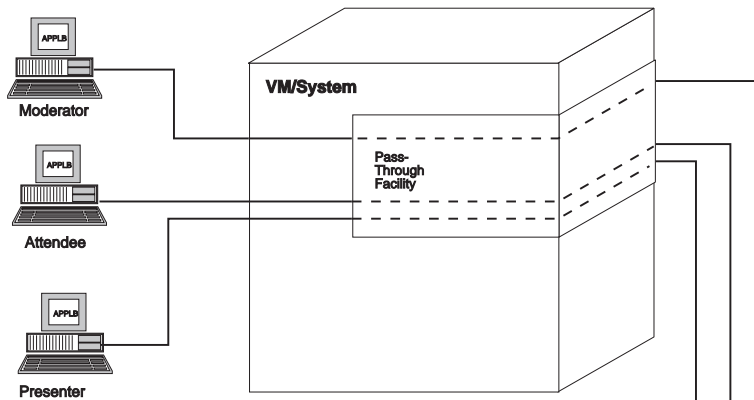
## Session Mode

Before **session mode** starts, the CVIEW virtual machine associates all session participants with a single interactive session. This allows the CVIEW virtual machine to provide identical screen images to the participants at the same time.

At the start of a session, the VM logo of the system where the CVIEW virtual machine runs is displayed to each session participant. In session mode only one user, with an active keyboard, logs on to the system. This user can log on any virtual machine user-ID defined in the system's directory or may Pass-Through to another system node if desired. Session mode continues until a user with an active keyboard logs off the system. After the LOGOFF command has been processed, all session participants return to setup mode, from which they can leave CVIEW or define another CVIEW session.

Figure 2 shows different types of interactive sessions in which users can participate using CVIEW.

### Networking Users



### Local Users

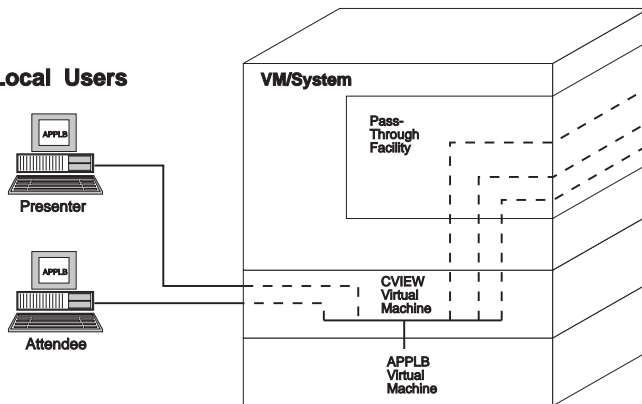


Figure 2. Different Types of Interactive Sessions Using CVIEW

### Notify Mode

**Notify mode** occurs when a consult or conference session is automatically suspended by the CVIEW virtual machine in order to display the **NOTIFY** panel. The NOTIFY panel advises a user that another user is sharing his interactive session and associated resources. The NOTIFY panel is displayed only if its use was selected during customization. Notify mode can occur at two different time intervals during a session. The first panel is displayed a specified number of seconds after session mode begins. Subsequent panels are displayed at preset intervals as the session progresses.

When a NOTIFY panel appears on a user's screen, he must press the ENTER key to provide a positive response before the session can continue. Pressing the HOME or QUIT PF key in response to the NOTIFY panel terminates the entire consult or conference session. NOTIFY panels can be displayed to all session participants or only to participants who have entered data or commands since the last NOTIFY panel was displayed. All participants receiving a NOTIFY panel must respond to it.

### Control Mode

The consultant and moderator can suspend a session to enter **control mode** by pressing the Program Attention 1 (PA1) key. The consultant then sees a **CONSULT SESSION CONTROL** panel. The moderator sees a **CONFERENCE CONTROL** panel.

When the **consultant** activates control mode, he can assign or change a PF key for the display-input option or request CVIEW to forward the PA1 key function to the host or an application program for processing. After the consultant selects a PF key for the display-input option, he presses the ENTER key and the consult session resumes. The PF key assigned for the display-input option can be used by both the consultant and the recipient.

When the **moderator** activates control mode, he can also assign or change a PF key for the display-input option or request CVIEW to forward the PA1 key function to the host or an application program for processing. The PF key assigned for the display-input option can be used by the moderator and any presenters.

In addition, the moderator can request the **CONFERENCE PARTICIPANTS** panel, which when displayed allows the conference session to resume without the moderator if a presenter also exists.

The moderator can assign presenter status to any number of attendees listed on the CONFERENCE PARTICIPANTS panel. Likewise, attendee status can be assigned to any presenters listed. The moderator can drop participants from the conference as necessary. After the moderator changes the status of participants, he must press the ENTER key to process the new assignments. The moderator returns to the conference in progress by pressing the SESSION PF key from the CONFERENCE PARTICIPANTS panel.

---

### Technical Overview

This section contains more detailed information about the way CVIEW operates. It is intended for the system programmer or administrator who must install, customize, and maintain the program.



## Accessing the CVIEW Virtual Machine

CVIEW processes as a CMS application in a disconnected virtual machine. The CVIEW virtual machine is logged on by the person designated as the CVIEW operator. The operator loads the DXGCVIEW exec to start CVIEW. After the CVIEW virtual machine is operational, the operator can enter a CP DISC command.

Users who want to establish a CVIEW session do not log on the CVIEW user-ID. Instead, they enter the 'DIAL CVIEW' command. After a user dials the CVIEW virtual machine, he is in setup mode. The CVIEW virtual machine communicates with his display station as a **GRAF device**, rather than as a virtual machine console.

During setup mode, a **GRAF address** (501 through 530) is assigned to each user's display station. When session mode starts, CP creates a **logical device** for the CVIEW session. The CVIEW virtual machine associates the GRAF addresses in a session with the address of the logical device. During session mode, the **logical device address** is used by the CVIEW virtual machine to communicate with each user's display station as though a single virtual machine console existed for the interactive session.

Figure 3 shows how the operator and end users access the CVIEW virtual machine to perform their specific tasks.

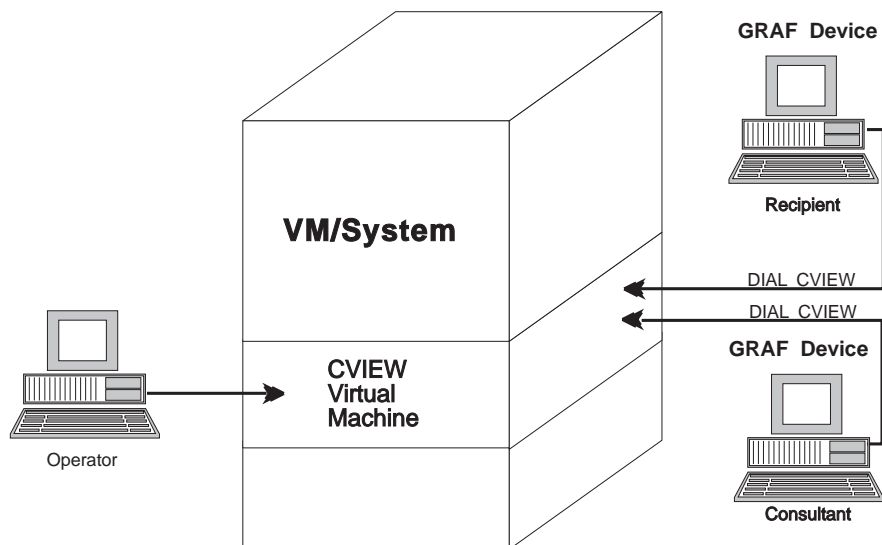


Figure 3. Accessing the CVIEW Virtual Machine

## Networking Sessions

A CVIEW **networking session** includes one or more users who have accessed the CVIEW virtual machine through the IBM VM/Pass-Through Facility. These users are called **networking users**.

The IBM VM/Pass-Through Facility allows display stations from another hardware system to access the target system node where the CVIEW virtual machine is running. A Pass-Through session can be

## CVIEW - Product Overview

established by issuing the CP DIAL command from the CP environment, by issuing the PASSTHRU or MPVM commands from a virtual machine console in the CMS environment, or by using the PVMG bridge.

Figure 4 shows how a networking session relates to the CVIEW virtual machine.

### Networking Users

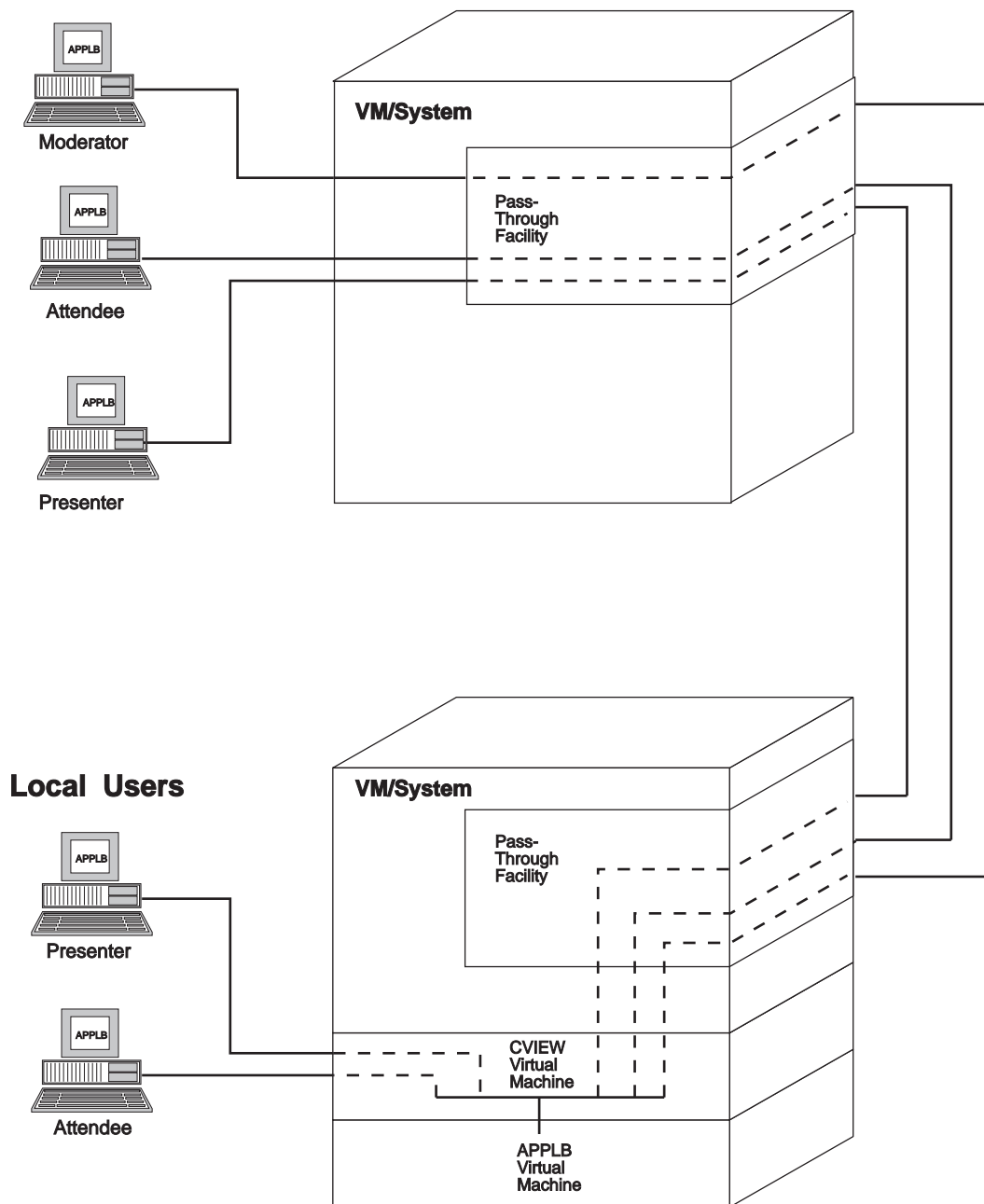


Figure 4. An Example of a Networking Session

## Networking Considerations

After a user initiates a Pass-Through session, a logical device is created on the target system node for his Pass-Through session. This establishes communications with the target system node as though the user's display station were locally attached. (The CVIEW virtual machine also uses a logical device for communications in a consult or conference session.) The CVIEW virtual machine associates all display stations (GRAFs) in a session with a single logical device.

While CVIEW can support as many as 30 networking users, each one must wait in setup mode until all other networking users have dialed the CVIEW virtual machine. To accommodate additional consult or conference sessions that include networking users, you can create multiple CVIEW virtual machines on the same system. CP allows multiple virtual machines (for example, Pass-Through and CVIEW) to simultaneously create logical devices. Refer to the *z/VM: CP Programming Services, SC24-6272* for the Logical Device Support Facility limitations.

## PASSTHRU Considerations

The PASSTHRU command, which can be issued by Pass-Through users from within a virtual machine, supports a feature called temporary disconnect. This feature allows a Pass-Through user connected to a target system use the PASSTHRU command to temporarily disconnect from the Pass-Through session with that system and return control to local CMS. When the desired interaction with the local CMS system is complete, the user can resume the session with the target system by reissuing the PASSTHRU command.

If the session with the target system is not resumed within an installation-specified time limit, Pass-Through terminates the session.

Networking users who have issued the PASSTHRU command might cause the CVIEW virtual machine to temporarily suspend if the Pass-Through temporary-disconnect feature is used. The CVIEW virtual machine would be suspended until the session loaded using the PASSTHRU command is resumed or the installation-specified Pass-Through time limit expires (the default is 20 minutes).

Other conditions can result in similar problems. Any condition that causes the Pass-Through session to end may result in a temporary disconnect. A preventive step to minimize the time that the CVIEW virtual machine is suspended under these conditions would be to re-define the Pass-Through limit for temporary disconnects to the minimum.

---

## Display-input Option

Any consult or conference session participant whose keyboard is active can use the display-input option. In setup and control modes, a consultant or moderator can assign a PF key as the display-input key. After input data or commands are keyed in, the user presses that PF key and the input appears on all display stations in the session. The input is not processed by the host or application program until the ENTER key is pressed.

The display-input PF key can be pressed repeatedly to show a series of steps or corrections. When the cursor is moved with a cursor movement key, pressing the display-input PF key shows the new location of the cursor on all screens.

The display-input PF key shows only screen input changes that are processed by the Read Modified 3270 data-stream command. The Read Modified command does not send null codes back to CVIEW. Therefore, the results of pressing the ERASE EOF key do not show up when a user presses the display-input PF key.

The use of the display-input PF key is optional; the consultant or moderator can specify NONE instead of a PF key number. If a PF key is assigned, that key functions only as the display-input key in session mode. All other PF keys operate the way they were defined to your system and application program.

---

## Accommodating Different Display Station Types

CVIEW supports a variety of display station types and models. Because CVIEW displays identical screen images to all participants in a session, it must assign one display station type and model to each session when the session starts. The display station types and models CVIEW can assign to a session are 3277-2, 3278-2, 3278-3, 3278-4, or 3278-5. See “Device Type and Model Assignment” on page 88 for the specific display stations supported by CVIEW.

After CVIEW assigns a display station type and model to the session, the CVIEW virtual machine reads and writes information to all display stations in the session as if they were that same type and model. A user can only join a conference session after it has been started by the moderator if his display station type and model is compatible with the type and model already assigned to the conference session. See “Joining a Conference Session in Progress” on page 88 for details.

---

## Messages

CVIEW issues information, eventual action, and action messages to the user during setup mode as required. The messages explain responses if required of the user.

The information displayed to the CVIEW virtual machine console depends on the setting of EMSG for the virtual machine. Messages displayed to the CVIEW virtual machine console appear with either the message number and text, text only, or do not appear at all. It is recommended that EMSG be set to TEXT for the CVIEW virtual machine.

For detailed information on CVIEW messages, see *VM/Pass-Through Facility VM/Pass-Through Facility: Messages*, SC24-5648.

## Part 1. Administering, Installing, and Operating PVM

<b>Chapter 3. Administrative Considerations</b>	15
Installation Requirements	15
Storage Considerations	15
How Much Storage is Need for a PVMG Server?	18
Performance Considerations	19
Security Considerations	19
Restrictions	20
<b>Chapter 4. CVIEW Installation Planning</b>	23
CVIEW Virtual Machine Requirements	23
CVIEW Directory Entry	23
CVIEW Operating Environment	24
Multiple CVIEW Virtual Machines	25
<b>Chapter 5. Defining A PVM Configuration</b>	27
Planning a Network	27
Creating the Configuration File	28
Creating a CMS PROFILE EXEC for PVM	36
Creating a CMS Communications Directory	37
Creating a PVM Profile	38
<b>Chapter 6. Defining A PVMG Configuration</b>	41
Planning A PVMG Configuration	41
Planning the PVMG Server's Role in the PVM Network	42
Tailoring the PROFILE GCS File	48
<b>Chapter 7. Developing NAMES Files for MPVM</b>	49
Creating Entries for Applications	49
Defining Configuration Settings—:nick.?CONFIG	52
Creating Other Special Entries	52
Sample MPVMSYS NAMES File	53
Verifying the MPVMSYS NAMES File	54
Considerations for Individual User NAMES Files	54
Summary of MPVMSYS NAMES Contents	55
<b>Chapter 8. Operating the Network</b>	57
Starting PVM	57
Querying the System	61
Defining a Link Dynamically	63
Displaying the Path of a Session	64
Sending Messages	65
Issuing CMS Subset Mode Commands	65
Quiescing, Resuming, and Dropping Links, Devices, and Users	66
Using the PVM Programmable Operator (PVMPROP)/Command Response Interface (CRI)	68
Stopping PVM	69
Starting PVMG	69
Querying PVMG	69
Displaying the Path of a PVMG Session	70
Sending Messages	71
Dropping Users and Nodes	71

Stopping PVMG . . . . .	73
<b>Chapter 9. Using CVIEW . . . . .</b>	<b>75</b>
Starting the CVIEW Virtual Machine . . . . .	75
CVIEW Commands . . . . .	76
CMS . . . . .	76
CP . . . . .	77
DROP . . . . .	77
HELP . . . . .	78
PASSWORD . . . . .	78
QUERY . . . . .	79
SHUTDOWN . . . . .	81
STOP . . . . .	82
Establishing a Session . . . . .	82
Managing a CVIEW Session . . . . .	85
CVIEW PROFILE EXEC . . . . .	89
CVIEW Log File . . . . .	89

---

## Chapter 3. Administrative Considerations

This chapter contains information to consider before installing PVM

---

### Installation Requirements

- Each node in a PVM network must have a unique node ID, including each PVMG server. If more than one link to a non-PVM system node is required, that node must be assigned a unique node ID for each link. There can be a group of links between two PVM nodes (not including the PVMG server), but there cannot be more than one group between the two PVM nodes.
- APPC/VM considerations:
  - For PVM to define APPC links, Advanced Communications Function for VTAM (ACF/VTAM) Version 3 Release 2 (or higher) is required.
  - The AVS component of VM provides the gateway support needed to communicate over a SNA-defined network. APPC/VM uses AVS to access this VTAM controlled link. See *z/VM: CP Planning and Administration, SC24-6271* for detailed information on installing AVS.
- For 3088 type devices, specify SHARED=NO on the CNTLUNIT macro in the IOCP Gen.
- Class B privilege must be assigned to the PVM server when:
  - The PVM configuration file contains a MSGNOH record
  - Warning messages are allowed to be issued to PVM users
- Class B privilege must be assigned to the PVMG server when the PVMG configuration file contains a MSGNOH record.

Almost all data in the PVM and PVMG configuration files must be entered in uppercase. The session end string specified on the SESSTERM record is the only piece of data that can be entered in mixed case. PVM configuration file records must be entered in the sequence illustrated in Figure 10 on page 30. PVMG configuration file records do not need to be entered in a particular order.

---

### Storage Considerations

When planning a PVM installation, there are several considerations concerning the storage needed for PVM operation. This section covers some of these considerations.

### DASD Requirements

**Note:** For PVM DASD requirements, see the Program Directory supplied with the product tape.

Be sure that adequate auxiliary storage space is provided for daily PVM operations (specifically the PVM system disk). For example, additional storage is required for the AUDIT records if that option is used.

### Storage Needs in the Default Configuration

PVM (the PVM MODULE) is designed to run in a minimum of 5M virtual machine when run with certain supplied defaults. These defaults allow an installation to run with:

- 100 users
- 500 sessions
- 4 links

- 50 blocks.

PVM has three configuration file records that help reserve storage for a PVM system. These are the BUFFERS, EXT, and I/O records.

**BUFFERS Record:** The BUFFERS record specifies how many 4096-byte pages of virtual storage will be reserved for use as internal PVM buffers. The number of buffers required by the system depends on the number and type of active line drivers, the number of active PVM users, and the number of commands being processed by PVM. If there is no BUFFERS record in the PVM CONFIG file, PVM uses a default of 30 pages. See “BUFFERS” on page 236 for more information on the BUFFERS record.

**EXT Record:** The EXT record specifies how many entries will be created to handle various external interrupts. One entry is used by the system as the default VMCF message handler, and one entry is used by the system to handle SMSG interrupts. (VMCF stands for Virtual Machine Communication Facility, a CP function that provides a method of communication and data transfer between virtual machines operating under the same VM system.) In addition, there must be one entry for each active user accessing PVM from CMS (by using the PASSTHRU command). There must also be one entry for each user gaining entry to the local CP system as a target node. Also, the APPC type line driver uses 2 entries for each active line driver. The IUCV type line driver uses one entry for each active line driver to PVMG. MPVM uses 2 entries for each active user. Each entry requires 16 bytes of storage. If there is no EXT record in the PVM CONFIG file, PVM uses a default of 50 entries.

**I/O Record:** The I/O record specifies how many entries will be created to handle I/O interrupts. One entry is used by the system to handle attention interrupts. In addition, there must be one entry for each active line driver (START LINE), and one entry for each active attached display support task (START GRAF). Each entry requires 16 bytes of storage. If there is no I/O record in the PVM CONFIG file, PVM uses a default of 50 entries. See “EXT” on page 238 for more information about the EXT record.

## Modifying PVM Storage Needs

Change the amount of PVM storage by using the BUFFERS, EXT, and I/O configuration file records. Appendix C, “PVM Configuration File Records” contains algorithms to compute the values for these records based on the PVM system configuration.

This section describes things to consider before installing PVM, including how much storage:

- The MPVM component requires in a user's virtual machine
- Is required by the PVM and PVMG server virtual machines.

## How Much Storage Is Needed in Each User's Virtual Machine for MPVM?

The amount of storage that MPVM requires in each user's virtual machine depends on whether shared segments are used. Whether shared segments are used or not, remember that the storage figures shown here are only estimates.

**Virtual Storage Required When Using Shared Segments:** By building a shared segment, there is just one copy in storage, rather than individual copies in each user's virtual machine. For example, a shared segment could contain all of MPVM's CMS end user interface code.

An MPVM shared segment can contain the following items:

- Code for the MPVM user module that contains the MPVM user interface



- The message repository, which contains text for all messages. (The default language for these message texts is American English)
- The command table, which contains MPVM variable names and keywords for use on MPVM user commands.

See “Installing MPVM in a Shared Segment” on page 108 for information about installing MPVM in a shared segment.

In addition, commonly-used MPVM macros and commands can be added to the CMSINST installation shared segment. Refer to “Adding Common MPVM Commands and Macros to an Installation Shared Segment” on page 110.

Assuming MPVM is installed in a shared segment, here is how to approximate the storage that an MPVM user requires for each virtual machine:

1. Estimate the size of dynamic working storage obtained during execution. This is difficult to predict. Working storage is obtained for a variety of user-initiated activities. However, storage is definitely required for the following:
  - a. The screen size, multiplied by 2 for screen-image buffers. (Every MPVM terminal screen has two display *planes*; this is described in the *VM/Pass-Through Facility: Programming Reference* book.) For example, for a 24x80 screen size, the total is 3840 bytes ( $2 \times 24 \times 80 = 3840$ , round to 4K).
  - b. If any target application accessed through PVM loads programmed symbol sets, add storage for a third screen-image buffer. For example, for a 24x80 screen size, add 1920 bytes ( $24 \times 80 = 1920$ , round to 2K).
  - c. A maximum-supported screen size ( $3 \times 62 \times 160 = 13020$  bytes, round to 30K) for a console read buffer, which may be needed by target applications.
  - d. 16K for a basic dynamic storage area
  - e. Anywhere from 32K to 160K for execution storage buffers for the portion of MPVM that still resides in the user's virtual machine. Use 85K as an average.

Here is a typical example for the user virtual machine storage required when (1) using an MPVM shared segment and (2) storing most MPVM macros and commands in the installation shared segment:

A 32x80 terminal ( $32 \times 80 \times 2 = 5120$ , round up)	=	6K
+ No programmed symbol sets loaded	=	0K
+ Storage for a console read buffer	=	30K
+ Dynamic storage area	=	16K
+ Other user virtual machine requirements	=	137K
	-----	
		137K

**Storage Required When Not Using a Shared Segment:** If shared segments are not being used, the storage required in the previous section, *plus* the following is needed:

- Approximately 198K for a private copy of the MPVM user module (including the message repository)
- About 100K for the commonly used MPVM commands and macros that are automatically loaded into storage.
- About 4K for the command language (variables and keywords) table

Here is a typical example for the user virtual machine

+ Storage required from previous section	=	137K
+ Private MPVM user module	=	198K
+ PVM commands and macros	=	100K
+ Message repository and command table	=	22K
		-----
		457K

Using shared segments for MPVM can save about 320K per virtual machine.

## How Much Storage Is Needed for a PVM Server?

The PVM server virtual machine typically runs as a disconnected server virtual machine. It initially directs all PVM connections and maintains screen image buffers for each MPVM session. Below are some guidelines to help determine the size of PVM server virtual machine needed:

- The MAXUSER setting times 1200 bytes. The MAXUSER setting defines the maximum number of MPVM users allowed concurrently. The default is 100.
- Each active session times 6-30K bytes. This is the bulk of the storage, used for screen image buffers. The number here depends on terminal sizes, support of extended data streams, and use of program symbol sets. 8K per session as an average is used.

The default maximum number of sessions is 500: 100 users, 5 sessions per user.

- The number of defined links (PVM and PVMG) times 12K
- The BLOCKS setting times 16K. PVM initially allocates 50 storage blocks (16K each) for the PVM module. However, this value might need to be increased.

Here is an example:

100 users (default) * 1200 bytes per user	=	120K
+ 500 sessions * 8K (average) per session	=	4000K
+ 4 links * 12K (approximate) per link	=	48K
+ 50 blocks (default) * 16K bytes per block	=	800K
		-----
		4968K

Because 1024K=1M, 4968K is approximately 4.7M. The PVM server virtual machine in this example could probably run comfortably with 5MB of virtual storage.

In the sample directory distributed with PVM, the PVM server virtual machine is defined as 5MB with a maximum of 8MB.

---

## How Much Storage is Need for a PVMG Server?

The PVMG server virtual machine runs as a disconnected server virtual machine. Below are some guidelines to help determine the size of the PVMG server virtual machine needed:

- PVMG program storage = 450K
- Additional storage for each link = 10K
- Additional storage for 50 users = 50K
- Additional language repositories = 43K
- System trace table = 16K
- Link trace table for each link = 8K.

In the sample directory distributed for PVMG, the PVMG server virtual machine is defined as 4M with a maximum of 8M.

---

## Performance Considerations

See “Enhancing PVM Performance” on page 107 to find out ways to improve PVM performance. The performance options should be used for peak PVM performance.

---

## Security Considerations

There are a number of possible security exposures to consider when planning a PVM network. This section covers some of the areas to take into consideration when planning a PVM installation.

### Notepad Facility

Sensitive data fields, not usually visible on the user's screen (for example, passwords), can be exposed if copied into the PASSTHRU DATA or *nickname* MPVMDATA files by the MPVM or PASSTHRU Notepad facility. It is the user's responsibility to safeguard this file if a local security problem exists.

### Session Security

PVM users begin with a PVM session to gain access to their desired target, then start a session with the target system or application. With *some* targets, the sequence for terminating sessions is important. If termination does not occur properly, a security exposure could exist.

In such cases, it is essential that the target session be terminated first, *before* the PVM session is terminated. If this sequence is not followed, the target session can remain active and open to access by subsequent users.

To avoid this situation, users should follow the proper termination sequence.

Users should be aware that *any* condition that causes the PVM session to end before the user has properly logged off from the target system or application can result in the same security exposure. Some such conditions are:

- The user terminates the PVM session by entering the Session Terminate string (default ####) or pressing the designated Session Terminate key.
- The user temporarily disconnects from the target session and does not resume the session within the defined or default (20 minutes) time limit.
- The operator (or other authorized user) issues a DROP command that affects the session.
- PVM abends because of either an internal problem or a problem in the VM operating system.
- A hardware problem affects the user's display station or the communication path between the display station and the target system.

This event is, of course, beyond the user's control. In some sessions, however, the user can take preventive steps to safeguard session security. These are sessions involving another PVM system or PVMG system. (That is, when the target is either a non-PVM system accessible through another PVM or PVMG system, or the target is an application available on another PVM system.)

The preventive steps consist of establishing an intermediate PVM session (on the other PVM system) before initiating the target session. The user needs a user ID to log on the other PVM system. Once logged on and running CMS, the user issues the PASSTHRU command and then selects:

- The desired non-PVM target system, or

## Restrictions

- The PVM system to start a local session and then dials the desired target application on that system.

The intermediate PVM session protects the target session in the event of a problem affecting the communication path between the two PVM systems.

There are several methods an installation can use to minimize or eliminate this security exposure. It can do one or all of the following:

- Remind users to end *all* target sessions *first* (using the standard logoff procedure). (Do this by sending a memorandum to PVM users or by issuing a message to them at their display stations.)
- Modify the target application program to require a new logon sequence after it receives a power-on reset condition. (PVM notifies the target of an interruption or termination of the PVM session by indicating a power-on reset condition. The security exposure exists because some target applications ignore the condition.)

---

## Restrictions

Be aware of several PVM restrictions. Some restrictions are hardware related, some are system related, and others are implementation restrictions.

### General Restrictions

- When logged on to the PVM or PVMG user ID, the operator or administrator should enter these commands to prevent errors:

```
cms set fullscreen off
cdbc cons off
```

When logged on to the PVMG user ID, the operator must prefix PVMG commands with pvmg.

- The SYS REQUEST key is not passed when the PASSTHRU command is used.
- The maximum data length that can be transferred as one unit to or from a 3270 display station varies by line type. For APPC, 3088, and CTCA link types, the maximum input and output data length is 64K bytes.
- The PASSTHRU Notepad facility requires that the user's display station have PF keys. MPVM's Notepad facility does not require PF keys. The notepads do not store extended attributes in the PASSTHRU DATA or *nickname* MPVMDATA files. MPVM will store extended attributes if the ATTRIBUTE variable setting is turned on. Also, single APL and TEXT characters whose hex representation requires 2 bytes, such as underscored APL characters, are stored as blanks.
- Issuing the CMS HX command on the PVM user ID while PVM is executing causes unpredictable results. Use the PVM QUIT or SHUTDOWN command to halt execution of PVM. (See Appendix B, "PVM Commands" for information on the format and use of these commands.)
- The 3270 ERASE ALL UNPROTECTED (EAU) command cannot be totally supported because of a Diagnose X'58' restriction. If an EAU command is received, it is simulated with a WRITE command and the 3270 order, ERASE UNPROTECTED TO ADDRESS. This causes the following conditions:
  - All Modified Data Tags (MDTs) are reset, unlike the EAU command which only resets the MDTs of unprotected fields
  - Always unlocks the keyboard.

When the PASSTHRU command is invoked, the local CP takes control of the display station to write any messages. The currently displayed screen is lost and any subsequent application-generated output is

displayed on a new screen. This new screen does not have any of the defined fields or data that were present on the previous screen. This makes it impossible to enter any data or commands. (Pressing CLEAR signals the host application that the screen has been cleared.) This forces the application to rewrite the entire screen. CP TERMINAL BREAKIN GUESTCTL reduces this problem to a minimum but it is available only to locally attached, non-SNA 3270 devices.

**PSPI**

## Access Security Exits Restrictions

The Access Security Exits section of this book provides sample exit code for the PVM server component of PVM. These security exit routines enable an installation to control sessions originating at a local node, outside the local node, and commands that are destined for or originated at another node. The security exits are built in to the PVM module supplied on the PVM 2.1.1 product tape and when built by VMSES/E. If these exits are not required for PVM 2.1.1, place a dummy REMOTE AUTH file on the PVM's 191 disk or a disk which the PVM user id has access to.

**PSPI end**

**GUPI**

## PCCF Restrictions

- When developing an application using IUCV, the application programmer must consider that the USERDTA field of the IPARML control block is used by both the PVM IUCV application-to-application support and CMS IUCV support.
- The maximum data length on an IUCV SEND or an IUCV REPLY is 32768 bytes if both the origin and target applications are on VM systems.
- CP IUCV PRIORITY cannot be used unless authorized by appropriate CP Directory statements.
- One-way SENDs are not synchronized and the *message complete* interrupt does **not** mean that the opposite end has received the data.

**GUPI end**

## Channel to Channel Adapter (CTCA) Restrictions

- Two CTCA links cannot be used to connect two VM systems through the same two channels.

## Logical Device Support Restrictions

- All command-chained CCWs are treated as separate unchained CCWs. Therefore, results that depend on commands being chained are unpredictable. The one emulated condition is a READ chained from a WRITE of length 4, containing a WCC character and a SBA character. A scan routine processes the data from the READ and discards all data prior to the address referenced by the SBA in the WRITE command. Note though that this emulated condition does not occur if the CCW chain is issued to a virtual display station. In this case, CP ignores the WRITE CCW.
- Data-chained CCWs have the data combined into a single buffer. The combined length cannot exceed the PVM data length restrictions.



---

## Chapter 4. CVIEW Installation Planning

This chapter describes the distribution media and the operating environment required for the installation of CVIEW, and the requirements for migrating to CVIEW.

---

### CVIEW Virtual Machine Requirements

You must create a directory entry for the CVIEW virtual machine before installing CVIEW. CVIEW requires specific options on several directory definition control statements. The following topics describe the directory-entry requirements.

#### Disk Space

The minimum disk space requirement for the CVIEW virtual machine A-disk is five (5) cylinders of 3380 disk space, or the equivalent. If you format the 191 A-disk prior to installation, use a block size of 4096 for optimum performance.

#### Virtual Storage

The CVIEW virtual machine requires 5 megabytes of virtual storage if it is to support up to 30 concurrent users at separate display stations. The number of concurrent users to be supported by the CVIEW virtual machine is selected during customization. See Chapter 16, "Customizing CVIEW," for details.

#### Privilege Class

CVIEW requires a privilege class of 'B' or 'G'. Privilege class B provides message no header (MSGNOH). MSGNOH returns messages without the standard header associated with the MESSAGE command to users for CVIEW privileged and general commands. An important consideration to note in assigning privilege class B is that any user specified during customization as a CVIEW privileged user can enter privilege class B CP commands to the CVIEW virtual machine.

#### Channel Mode and Nonshared Virtual Control Units

You must specify block multiplexer (BMX) mode of channel operation and nonshared virtual control units (VCUNOSHR) on the OPTION control statement. In BMX mode, the virtual channel is not busy until the initial start I/O command (SIO) is complete; this allows the successful start of multiple SIOs to different devices on the same channel. The VCUNOSHR option ensures that the CVIEW virtual machine does not use shared subchannel protocols.

**Note:** The BMX and VCUNOSHR options are only good under the 370 option.

#### User ID

For further installation details, see the PVM 2.1.1 Program Directory.

---

### CVIEW Directory Entry

The control statements in Figure 5 satisfy the directory entry requirements for the CVIEW virtual machine.

```
USER CVIEW XXXXXXXX 5M 5M G (Note 1)
OPTION BMX VCUNOSHR (Note 2)
ACCOUNT number
IPL CMS
CONSOLE 01F 3215
SPOOL 00C 3505
SPOOL 00D 3525
SPOOL 00E 1403
LINK MAINT 190 190 RR
LINK MAINT 19E 19E RR
LINK P684100E 405 191 MR
```

Figure 5. Control Statements for a CVIEW Directory Entry

### Notes:

1. The USER control statement defines the user-ID as CVIEW, the virtual storage as 5 megabytes, and the privilege class as G.
2. The OPTION control statement defines the mode of channel operation as BMX. BMX and VCUNOSHR are not supported in VM/ESA.

Virtual addresses 501 through 530 are reserved for use by CVIEW and cannot be assigned to any resource associated with the CVIEW virtual machine.

For additional information about directory entries, see the *z/VM: CP Planning and Administration*, SC24-6271.

---

## CVIEW Operating Environment

The following section describes what is required to operate CVIEW.

### Machine Requirements

CVIEW supports only display stations that are compatible with at least one of the following IBM device types:

- IBM 3277 Model 2 (screen size: 24 X 80)
- IBM 3278 Model 2 (screen size: 24 X 80)
- IBM 3278 Model 3 (screen size: 32 X 80)
- IBM 3278 Model 4 (screen size: 43 X 80)
- IBM 3278 Model 5 (screen size: 27 X 132)

The format of the data used in a CVIEW session must conform to 3270 architecture for local display stations. Extended data streams (EDS), which includes extended color, extended highlighting, programmed symbols, and partitions, are not supported.

CVIEW supports the following IBM display stations in any combination:

- IBM 3277 - Model 2
- IBM 3278 - Models 2, 3, 4, 5
- IBM 3279 - All models when operated in 4-color compatibility mode
- IBM 3178



- IBM 3179
- IBM 3180
- IBM 3290
- IBM 3270 PC - When operated in “host session” mode
- IBM 3270 PC/G - When operated in “host session” mode
- IBM 3270 PC/GX - When operated in “host session” mode
- IBM PC XT/370 - When operated in “remote session” mode

CVIEW requires at least two (2) display stations to establish a session.

## Multiple CVIEW Virtual Machines

You can create additional CVIEW virtual machines to support additional networking sessions and to support more than 30 concurrent users. CP allows multiple virtual machines to simultaneously create logical device. If the Pass-Through Facility is installed on your system, it uses a virtual machine that creates logical devices. This chapter presents the steps necessary to create additional CVIEW virtual machines.

Figure 6 shows how additional CVIEW virtual machines relate to the base CVIEW virtual machine.

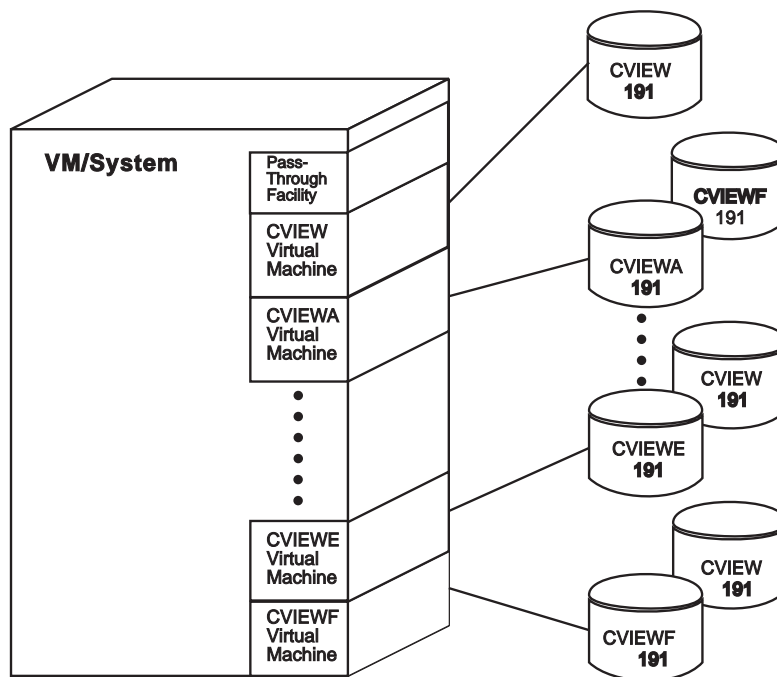


Figure 6. A System with Multiple CVIEW Virtual Machines

To create additional CVIEW virtual machines, follow these steps:

1. You must install CVIEW before you can create additional CVIEW virtual machines. If you have not installed CVIEW, do so now. See the PVM 2.1.1 Program Directory for instructions.
2. You must create a directory entry for each additional CVIEW virtual machine. Each additional virtual machine requires the same directory entries as the base CVIEW virtual machine except for disk space

## CVIEW Installation Planning

and the user-ID. The 191 A-disk for each additional virtual machine requires only five (5) cylinders. The user-ID cannot be CVIEW. See the PVM 2.1.1 Program Directory for directory entry requirements.

3. Log on the base CVIEW virtual machine. (Enter: L CVIEW)
4. If CVIEW is running, enter: SHUTDOWN
5. Rename the CONSULT IDNAME file on the base CVIEW virtual machine A-disk to change the file mode from A0 to A1. This step allows the customization exec, DXGINIT, to copy the CONSULT IDNAME file to the 191 A-disk for the additional CVIEW virtual machine.
6. Log off the base CVIEW virtual machine.
7. Log on to the user-ID assigned to one of the additional CVIEW virtual machines.
8. Enter: CP LINK CVIEW 191 xxx RR password
9. Enter: AC xxx B/A

Steps 8 and 9 link and access the base CVIEW 191 disk as a read-only extension of the A-disk.

10. The additional CVIEW virtual machine must be customized. Copy the base CONSULT IDNAME file to your new A disk as filemode 0. Enter the DXGINIT command. If you want different listed consultants for the additional virtual machine, respond *No* to the following customization prompt and proceed to step 11.

*Have you entered the consultant IDs in the CONSULT IDNAME file?  
Enter Yes or No. Press ENTER for YES.*

If your response was *Yes*, proceed to step 12.

11. If your response was *No* for step 10, customization ends. Edit the CONSULT IDNAME file at this time. When you finish editing the CONSULT IDNAME file, enter: DXGINIT and respond *Yes*.
12. Customization prompts you for the privileged users for the additional virtual machine. If the prompt lists CVIEW, change CVIEW to the user-ID assigned to this additional CVIEW virtual machine.
13. After DXGINIT finishes, edit the PROFILE EXEC to include the following commands:  
CP LINK CVIEW 191 xxx RR password  
ACC xxx B/A  
EXEC DXGCVIEW
14. To define another CVIEW virtual machine, log off and return to step 7.
15. Log on to the base CVIEW virtual machine and rename the CONSULT IDNAME file to change the file mode from A1 back to A0.
16. After each additional CVIEW virtual machine has been defined and customized, you can log on each additional virtual machine and start CVIEW. Users can access an additional CVIEW virtual machine by issuing a 'DIAL cviewid' command, where cviewid is the user-ID assigned to the additional CVIEW virtual machine.

---

## Chapter 5. Defining A PVM Configuration

Before starting PVM, the system administrator, should plan the network, create a configuration file to define the PVM network, and create a PVM profile exec and a PROFILE EXEC to automate system initialization. The same tasks are needed for PVMG. If MPVM is installed, create a systems names file called MPVMSYS NAMES. See Chapter 7, “Developing NAMES Files for MPVM” on page 49 for instructions on creating the MPVMSYS NAMES file.

Also, there are some requirements and conditions to consider before planning the network. See Chapter 3, “Administrative Considerations” on page 15 for this information.

---

### Planning a Network

Before defining the PVM network, it is a good idea to plan out what it will look like. To help plan and keep track of a PVM network, the following information is required:

- Local PVM system name
- Target system names
- PVMG server names
- Types of connection paths for the target systems
- Routing paths.

**Note:** In order to tailor PVM files, log onto PVM and have R/W access to PVM's 191 minidisk (PVM links to P684100E's 401 minidisk as 191 in MR mode, through the CP directory). The 401 minidisk may been to be detached from P684100E to gain write access, because M/R mode only allows one R/W link at a time. Once logged on, shutdown PVM (if running) by issuing:

```
SHUTDOWN IMMED
```

Then tailor the following files:

- PVM CONFIG
- PROFILE PVM
- MPVMSYS NAMES
- MPVM NAMES (as a sample for users).

## Defining PVM

The following table describes the sample files supplied on the PVM 2.1.1 LOCALSAM (2C2) disk for the PVM user id:

<i>Table 2. PVM 2.1.1 Supplied Sample Files for the PVM user id</i>	
File Name	Sample Description
PROFILE PVMSAMP	Sample profile PVM file.
PROFPVM EXECSAMP	Sample profile exec file.
PVMSAMP DIRECT	Sample directory file.
PVM CONFSAMP	Sample PVM CONFIG file.
UCOMDIR NAMESAMP	Sample CMS communications directory.

## Creating the Configuration File

Use a PVM configuration file on the PVM user ID to define the network. It contains lines called records. The records tell PVM how the nodes in the network are connected.

There are a few restrictions regarding the configuration file:

- All data in the file must be in uppercase, except for the session termination string specified on the SESSTERM record.
- Input for the file's records is restricted to columns 1 through 71, with no continuation.
- Some records must appear in a particular order See Figure 10 on page 30 for more information on the required order.
- The file type must be CONFIG.

## Sample Configuration Files

The following is a configuration file for the PVM system, whose node ID is RMTPVM1.

```
*****
* Configuration file for node RMTPVM1 - RMTPVM1 CONFIG *
*****
LANG AMENG
XDBC CONS
LOCAL RMTPVM1
RESID PVM1
* Set up PVM to PVM links
LINK 200 RMTPVM2A CTCA
LINK 201 RMTPVM2B CTCA
LINK 500 RMTPVM5A 3088
LINK EAGLE RMTPVM5B APPC USER SMITH
LINK * PVMG1 IUCV USER PVMG
GROUP RMTPVM2 RMTPVM2A RMTPVM2B
GROUP RMTPVM5 RMTPVM5A RMTPVM5B
ROUTE RMTSYS6 PVMG1
ROUTE VMCMS RMTPVM2
```

Figure 7 (Part 1 of 2). Sample Configuration File for RMTPVM1 -- RMTPVM1 CONFIG

```

ROUTE RMTVM3 RMTVM2
ROUTE DEFAULT RMTVM5
* The configuration file records to this point are required to
* define all routes from RMTVM1 to the other nodes.
* The following records might apply to any PVM node.
AUTHORIZ OPR
AUTHORIZ JONES RMTVM2
SUBSET LISTFILE
AUDIT DATA RECORD
LNODE * AUTO
* Set the PF keys on the Selection Menu.
SELECT RMTVM1
SELECT RMTVM3
SELECT
SELECT RMTVM5
* Set time restrictions.
STATIME 1200
MAXIDLE 6000
MAXSESS 4
MAXSUSP 1500
MAXUSER 600
MAXDISC 3600
TDISC 600
TIMEOUT 500
MSGNOH
Specify that a VMDUMP is to be taken if abnormal termination of
PVM occurs. The VMDUMP option should be specified if VM Dump Tool
is to be used.
DUMP VMDUMP OPERATOR
* The BUFFERS, EXT, and I/O entries are not illustrated in this
* sample configuration
MSGOPTNS Y Y Y Y

```

Figure 7 (Part 2 of 2). Sample Configuration File for RMTVM1 -- RMTVM1 CONFIG

The following are partial samples of PVM configuration files for other PVM nodes. The records following the ROUTE records are not shown in these samples because they are similar to the ones in the configuration file for RMTVM1.

```

*****
* Configuration file for node RMTVM2 - RMTVM2 CONFIG *
*****
LOCAL RMTVM2
LINK 300 RMTVM1A CTCA
LINK 301 RMTVM1B CTCA
GROUP RMTVM1 RMTVM1A RMTVM1B
ROUTE VSEAPL RMTVM1
ROUTE MVSTSO RMTVM1
ROUTE RMTVM5 RMTVM1
ROUTE RMTVM3 RMTVM1
ROUTE RMTSYS6 RMTVM1

```

Figure 8. Sample Configuration File for a Target System -- RMTVM2 CONFIG

## Defining PVM

```
*****
* Configuration file for node RMTPVM5 - RMTPVM5 CONFIG *
*****
LOCAL RMTPVM5
RESID PVM5
LINK 600 RMTPVM1C 3088
LINK HAWK RMTPVM1D APPC USER JONES
GROUP RMTPVM1 RMTPVM1C RMTPVM1D
ROUTE RMTPVM2 RMTPVM1
ROUTE RMTPVM3 RMTPVM1
ROUTE RMTPVM4 RMTPVM1
ROUTE MVSAPL RMTPVM1
ROUTE VSEAPL RMTPVM1
ROUTE MVSTSO RMTPVM1
ROUTE VMCMS RMTPVM1
ROUTE RMTSYS6 RMTPVM1
```

Figure 9. Sample Configuration File for a Target Node Routed through an Intermediate Node -- RMTPVM5 CONFIG

The following sections describe the configuration file records for the sample configuration files for RMTPVM1, RMTPVM2, and RMTPVM5. The full record format and parameter descriptions are in Appendix C, "PVM Configuration File Records." The records are introduced in the order that they appear in the configuration file for the sample PVM system. Figure 10 illustrates the order of records in a PVM configuration file.

---

```
LOCAL
LINK
LINK
LINK
GROUP
ROUTE
ROUTE
AUTHORIZ
```

---

Figure 10. Order of Configuration File Records

The AUDIT, AUTO, BUFFERS, DUMP, EXT, I/O, LNODE, MAXCONN, MAXIDLE, MAXSESS, MAXSUSP, MAXTDISC, MAXUSER, MSGNOH, MSGOPTNS, OPID, PVMPROP, SELECT, SESSTERM, STATTIME, SUBSET, TCP, TDSIC, and TIMEOUT records may follow in any order following the last AUTHORIZE record.

The \*, LANG, and XDBC records can be placed anywhere in the configuration file.

The RESID record may follow anywhere in the configuration file after the LOCAL record.

## Commenting the PVM Configuration File

Use the comment (\*) record to put comments in the configuration file. For example,

```
* Configuration file for node RMTPVM1 - RMTPVM1 CONFIG
```

is the first record of the configuration file for RMTPVM1. The asterisk (\*) must be in column 1.

Comment records can be placed anywhere in the configuration file.

## Identifying the System Language

Use the LANG record to identify the default national language for a node. For example,

```
LANG AMENG
```

assigns American English as the default national language for RMTPVM1.

The LANG record can be placed anywhere in the configuration file.

## Excluding Double-Byte Character Messages

Use the XDBC record to exclude users from receiving DBCS messages. For example, if the LANG record identifies a double-byte character language such as KANJI, and the PVM console should not receive a DBCS message, enter the following:

```
XDBC CONS
```

Now any DBCS messages received by PVM for the console are intercepted and the equivalent message in the backup language is sent to the console in place of the DBCS message. The AUDIT file also receives DBCS message in the backup language.

See “QUERY” on page 211 for information on the DBCS settings and “CDBC” on page 185 for information on changing DBCS-exclusion settings.

The XDBC record can be placed anywhere in the configuration file.

## Identifying the Local Node Name

Use the LOCAL record to identify the local node name. For example,

```
LOCAL RMTPVM1
```

assigns the node ID RMTPVM1 to the local PVM system.

The LOCAL record must precede any other records in the configuration file except for comment records, the LANG record, or the XDBC record.

## Identifying a Unique Resource Name

Use the RESID record to define an APPC/VM resource identifier. For example,

```
RESID PVM1
```

assigns PVM1 as the identifier for node RMTPVM1. It is used as the resource identifier in the UCOMDIR NAMES file of other PVM nodes, not the local node. (:tpn. is the tag used in the UCOMDIR NAMES file to identify this resource. It is the target identifier for APPC links.)

The RESID record must follow the LOCAL record.

## Defining a Link from a Local Node to Another Node

Use the LINK record to define a link from the local node to another node, or to a PVMG node.

For example,

```
LINK 200 RMTPVM2A CTCA
LINK 201 RMTPVM2B CTCA
LINK 500 RMTPVM5A 3088
LINK EAGLE RMTPVM5B APPC USER SMITH
LINK * PVMG1 IUCV USER PVMG
```

are the links defined from node RMTPVM1.

## Defining PVM

- PVM to PVM connections are made from RMTPVM1 (address 200) to RMTPVM2A (address 300) via type CTCA, from RMTPVM1 (address 201) to RMTPVM2B (address 301) via type CTCA, and from RMTPVM1 (address 500) to RMTPVM5A (address 600) via type 3088.
- EAGLE refers to a nickname entry that is found in UCOMDIR NAMES. The entry defines the target node to connect to.
- An IUCV connection to the PVMG machine, PVMG1, at the VM user ID PVMG.

LINK records must follow the LOCAL record.

## Defining Groups

Use the GROUP record to form a group from already defined links. For example,

```
GROUP RMTPVM2 RMTPVM2A RMTPVM2B
GROUP RMTPVM5 RMTPVM5A RMTPVM5B
```

forms a group named RMTPVM2 containing CTCA link RMTPVM2A and CTCA link RMTPVM2B. Group RMTPVM5 contains the APPC link RMTPVM5B and 3088 link RMTPVM5A.

Use the following GROUP records to form the same groups as defined above:

```
GROUP RMTPVM2 RMTPVM2A
GROUP RMTPVM2 RMTPVM2B
GROUP RMTPVM5 RMTPVM5A
GROUP RMTPVM5 RMTPVM5B
```

GROUP records must follow all the LOCAL and LINK records.

## Defining Routes and Alternate Routes

Use the ROUTE record to define a routing path from the local PVM node to a specified target node. For example, ROUTE records for RMTPVM1 are:

```
ROUTE RMTSYS6 PVMG1
ROUTE VMCMS RMTPVM2
ROUTE DEFAULT RMTPVM5
```

The ROUTE records define routing paths from RMTPVM1 to:

- N2N6AVSC VTAM application routed through PVMG1 to the local PVMG node
- VMCMS through RMTPVM2.
- RMTPVM2, RMTPVM4, and MVSAPL through RMTPVM5 because of the DEFAULT keyword. The path to RMTPVM2 is an alternate path defined in case the usual CTCA route to RMTPVM2 is not available. RMTPVM5's configuration file must define the remaining linkage to MVSAPL through RMTPVM4.

ROUTE records must immediately follow all LINK and GROUP records.

PVM uses the LINK and ROUTE records to determine the path used to a target node. The procedure is:

1. If the target is a node defined by a LINK or GROUP record, and the link is currently connected, that path is used.
2. If the target node is not defined on a LINK, GROUP, or ROUTE record and a default route is specified on the ROUTE record using the DEFAULT keyword, then the session is routed through the specified default node.



3. The routing list is searched for an entry that has the target system node as an end link. If an entry is not found, the session cannot be completed. If the entry is found, the node ID of the *intermediate node* is checked to see if the direct link to the desired target node is connected.
  - a. If so, that path is used.
  - b. If not, the routing list is searched for a ROUTE entry (alternate route) for the *intermediate node*. If found, the route is checked to see if this link is connected. If connected, the session is routed through this alternate link. If the route entry is not found, session initiation cannot be completed.

Alternate paths are specified in the PVM configuration file. Care must be taken when specifying ROUTE records. The system programmers must ensure that the routing is consistent among the PVM nodes and that no routing loops occur.

## Authorizing the PVM Operator

Use the AUTHORIZ record to identify users allowed to issue the restricted subset of PVM commands and the nodes from which these commands may be issued. For example,

```
AUTHORIZ OPR
AUTHORIZ JONES RMTPVM2
```

authorizes the person identified by the user ID OPR to enter restricted PVM commands from the local node, RMTPVM1. The second AUTHORIZ statement authorizes user JONES to enter restricted PVM commands from node RMTPVM2 to RMTPVM1.

AUTHORIZ records must follow all LOCAL, LINK, GROUP, and ROUTE records.

## Identifying CMS Subset Mode Commands to PVM

Use the SUBSET record to identify to PVM command names that will be processed as CMS Subset mode commands. For example,

```
SUBSET LISTFILE
```

informs PVM that the CMS LISTFILE command will be processed as a CMS Subset mode command. See "SUBSET" on page 254 for more information on the SUBSET record.

**Note:** Using the SUBSET record may degrade PVM performance.

SUBSET records must follow all LOCAL, LINK, GROUP, ROUTE, and AUTHORIZ records. All SUBSET records must appear together.

## Automatically Connecting Users to a Node

Use of the LNODE serves two purposes when used with:

- Only one operand (a *nodeid*), the *nodeid* is stored by PVM in the destination field of the Initial Menu. This makes it easier for the users receiving an Initial Menu to select that node.
- Two operands (a node ID and the AUTO keyword), then all instances of when either the Initial Menu or the Selection Screen would be presented to the user are changed so that the user is given automatic access to the node specified by the *nodeid* operand. In this case, the installed system will find that the only instance of when a user is not given automatic access to the specified *nodeid* is when a CMS user starts the PASSTHRU command specifying a specific *nodeid* that is different from the *nodeid* specified in the LNODE record. For example:

```
LNODE * AUTO
```

## Defining PVM

automatically connects all users to RMTPVM1 (assumed to be the local node name). PVM will bypass presenting both the Initial Menu and Selection Menu. The system logo screen from RMTPVM1 is displayed.

LNODE records must follow all LOCAL, LINK, GROUP, ROUTE and AUTHORIZ records.

## Setting PF Keys for PVM Menus

Use the SELECT record to assign node names to PF keys on the PVM Selection Menu and the Initial Menu.

For example, the records:

```
SELECT RMTPVM1
SELECT
SELECT
SELECT RMTPVM3
SELECT
SELECT RMTPVM5
```

assign

- RMTPVM1 to PF1
- nothing to PF2 (PF2 is skipped)
- nothing to PF3 (PF3 is skipped)
- RMTPVM3 to PF4
- nothing to PF5 (PF5 is skipped)
- RMTPVM5 to PF6.

SELECT records must follow all LOCAL, LINK, ROUTE, GROUP, and AUTHORIZ records.

## Saving PVM Information

The AUDIT and DUMP records save information about PVM.

Use the AUDIT record to save PVM console data in a CMS file. This log can be helpful in finding solutions to problems with a PVM system. For example,

```
AUDIT DATA RECORD
```

causes the console data to be stored in a CMS file named DATA RECORD. The first record in the audit file contains a pointer to the last record written. If the audit file, DATA RECORD is full, the data wraps to the beginning of the file, overlaying the existing records. Specify a maximum size for the audit file that prevents the PVM minidisk from getting full.

Use the DUMP record to specify the type of dump to be taken if abnormal termination of PVM occurs. For example, the record:

```
DUMP VMDUMP
```

causes a VMDUMP dump to be taken of all PVM storage.

All AUDIT or DUMP records must follow all LOCAL, LINK, ROUTE, GROUP, and AUTHORIZ records.

## Specifying Restrictions

There are eight records that are used to specify various restrictions for certain events in PVM.

Use the STATTIME record to specify the number of seconds between calls to the Link Statistics Exit (DVMNMEL). For example, the following record tells PVM to call the Link Statistics Exit every 1200 seconds.

```
STATTIME 1200
```

Use the TDISC record to specify how long a PASSTHRU user is allowed to remain temporarily disconnected from a remote application. For example, the record:

```
TDISC 600
```

lets users remain temporarily disconnected for 600 seconds before PVM terminates the session.

Use the TIMEOUT record to specify the interval between status checks of remote nodes on network links. For example, the record:

```
TIMEOUT 500
```

specifies that network line drivers should check the status of the remote nodes at 500-second intervals.

**Note:** The value specified on the TIMEOUT record determines if a link is still active. The method of testing varies with line driver type. Each emulator and network line driver sets a timer to the TIMEOUT value when the associated link is first connected.

Time expiration is handled by PVM line drivers as follows:

Network line driver — A new time interval is set if line activity occurs during the current interval.

Otherwise, a new time interval is set, and a transmission to the remote end is attempted. If this transmission is not successful, the line driver is terminated with an indication of a line time out condition.

Restrictions may also be set for MPVM users by using the:

- MAXSESS record, which specifies the maximum number of sessions a MPVM user may establish (see “MAXSESS” on page 246)
- MAXIDLE record, which specifies the maximum time a MPVM user may remain idle before PVM terminates all of the user's sessions (see “MAXIDLE” on page 246) A user is considered idle when there is no communication from the MPVM user to PVM for any session.
- MAXSUSP record, which specifies the maximum time a MPVM user may have a session suspended before that session is terminated by PVM (see “MAXSUSP” on page 246)
- MAXTDISC record, which specifies the maximum time a MPVM user may remain temporarily disconnected before all of the user's sessions are terminated by PVM (see “MAXTDISC” on page 247)
- MAXUSERS record, which specifies the maximum number of MPVM users who can connect to PVM (see “MAXUSERS” on page 248).

All STATTIME, MAXSESS, MAXSUSP, MAXIDLE, MAXUSER, MAXTDISC, TDISC, or TIMEOUT records must follow all LOCAL, LINK, ROUTE, GROUP, and AUTHORIZ records.

### Specifying Message Requirements

Use the MSGNOH and the MSGOPTNS records to indicate how messages are displayed. Use the MSGNOH record to indicate that responses to all commands entered by the CP SMSG command must be returned using the CP MSGNOH command rather than the CP MSG command.

Use the MSGOPTNS record to determine whether the time stamps or message numbers for messages will be directed to the PVM user ID, the audit file, the users, and/or the operator. For example, the record:

```
MSGOPTNS Y Y Y Y
```

tells PVM to specify the time stamps and the message numbers on the messages to the PVM user ID and the audit file.

MSGOPTNS records must follow all LOCAL, LINK, ROUTE, GROUP, and AUTHORIZ records.

### Specifying Storage Requirements

Use the BUFFERS, EXT, and I/O records to get more storage for the PVM system when the default is not enough. For more information on these records and how to use them, see “Storage Considerations” on page 15

All BUFFERS, EXT, and I/O records must follow all LOCAL, LINK, ROUTE, GROUP, and AUTHORIZ records.

---

### Creating a CMS PROFILE EXEC for PVM

The PROFILE EXEC file has CP and CMS commands in it. Include the SET RUN ON command in the PROFILE EXEC so when disconnecting from the PVM virtual machine and logging on again, it immediately returns to the PVM environment and receives the log messages and a reconnected message. A CP BEGIN command is not required to receive these messages.

To automate the initialization of PVM, the final statement in the CMS PROFILE EXEC for PVM must invoke the RUNPVM command supplied with the product. If default names are not used for the PVM module (PVM MODULE), the configuration file (PVM CONFIG), and the PVM profile exec (PROFILE PVM), start the RUNPVM command in its correct format. For more information on the format of the RUNPVM command, see “RUNPVM” on page 219.

A sample PROFILE EXEC is provided with the distribution tape that can be updated for each installation. The PROFILE EXEC will be similar to the following (the sample below assumes the default names are used):

```

/* This exec is used for automatic startup of PVM Version 2 */
Address 'COMMAND'
TRACE I
'CP SPOOL CONSOLE TO MAINT START'
'CP SET RUN ON'

/*****
/* Attach the necessary lines to our user ID - this assumes that */
/* we have the necessary class 'B' to perform this task.          */
*****/
'CP ATTACH 200 * 200 '
'CP ATTACH 300 * 201 '
'CP ATTACH 500 * 500 '
'CP ATTACH 501 * 501 '

'CP SPOOL CONSOLE STOP CLOSE'

/* Get PVM up and running */
'EXEC RUNPVM'

exit

```

Figure 11. Sample PROFILE EXEC for node RMTPVM1

## Creating a CMS Communications Directory

To set up an APPC link, load the CMS communications directory into PVM virtual machine's storage. A CMS communications directory is a special NAMES file. It defines symbolic destination names for target nodes and resource names. This file must have a file type of NAMES, and the file name can be any valid eight-character name. The user default file name is UCOMDIR.

The CMS communications directory is reloaded when PVM is started using the RUNPVM command or when CMS is IPLed.

The following table describes the entry tags contained in the UCOMDIR NAMES file for PVM virtual machine.

Entry Tag	Meaning
:nick.	defines the symbolic destination name for the target node.
:luname.	specifies the name of the global gateway on this node and the name of the global gateway at the target node. (The system administrator defines the names of the global gateways.)
:tpn.	specifies the resource ID of the target node. This is the same name found in the RESID record of the target configuration file. Or, if the target configuration file does not contain a RESID record, :tpn. is the PVM node ID.
:modename.	specifies the mode name for the SNA session connecting the gateway and the target nodes.
:security.	specifies the security level of the conversation. Use the security level of PGM or SAME and specify the user ID and password entries.
:userid.	specifies the user ID at the target node.
:password.	specifies the password of the user at the target node.

For security reasons, do not have the user ID and password information in the UCOMDIR NAMES file. To avoid this situation, place the user ID and password in the requester virtual machine's CP directory entry using the APPCPASS directory statement.

## Defining PVM

The following figures are samples of CMS communications directories for node RMTPVM1 and node RMTPVM5.

For information on the APPCPASS directory statement and for details on creating CMS communications directories, see *z/VM: CP Planning and Administration SC24-6271*.

```
:nick.EAGLE      :lname.PVMGATE1 PVMGATE5
                  :tpn.PVM5
                  :modename.VMINT
                  :security.PGM
                  :userid.JONES
                  :password.IANIAN
```

Figure 12. Sample CMS Communications Directory for RMTPVM1

```
:nick.HAWK       :lname.PVMGATE5 PVMGATE1
                  :tpn.PVM1
                  :modename.VMINT
                  :security.PGM
                  :userid.SMITH
                  :password.WALTER
```

Figure 13. Sample CMS Communications Directory for RMTPVM5

---

## Creating a PVM Profile

The PVM profile has PVM commands in it. It provides a way to automate many of the daily operations of starting lines or devices. A PVM profile must have a filetype of PVM.

A sample PVM profile called PROFILE PVMSAMP is provided with the distribution tape and may be modified and saved as PROFILE PVM. This file is required for fully automated system initialization. It issues START commands for the defined lines, and by issuing START GRAF (attached display support) commands, allows users to access PVM through the CP DIAL command. If it exists, the PVM profile is executed during system initialization.

The information in the examples below illustrate how PROFILE PVM files for the PVM systems in the sample might be written. Line addresses on the START LINE entries must match those in the configuration file LINK records for the associated PVM system. These addresses must be from from X'100' to X'FFF'. To start a PCCF task, use the PCCF option on the START command.

PVM profiles must have a file type of PVM. The PVM profile files for the PVM VM systems illustrated previously might be:

```
*****
* PVM Profile for node RMTPVM1 - RMTPVM1 PVM *
*****
START LINE 200
START LINE 201
START LINE 500
START NODE RMTPVM5B
START NODE PVMG1
START GRAF 450
START GRAF 451
START GRAF 452
START GRAF 453
START GRAF 454
START PCCF JOEUSER
(etc.)
```

Figure 14. PVM Profile for RMTPVM1 -- RMTPVM1 PVM. Sample for the local system.

```
*****
* PVM Profile for node RMTPVM2 - RMTPVM2 PVM *
*****
START LINE 300
START LINE 301
START GRAF 460
START GRAF 461
(etc.)
```

Figure 15. PVM Profile for RMTPVM2 -- RMTPVM2 PVM. Sample for a target system.

```
*****
* PVM Profile for node RMTPVM3 - RMTPVM3 PVM *
*****
START GRAF 470
(etc.)
```

Figure 16. PVM Profile for RMTPVM3 -- RMTPVM3 PVM. Sample for an intermediate node.

```
*****
* PVM Profile for node RMTPVM4 - RMTPVM4 PVM *
*****
START GRAF 480
(etc.)
```

Figure 17. PVM Profile for RMTPVM4 -- RMTPVM4 PVM. Sample for a target node routed through an intermediate node.

```
*****
* PVM Profile for node RMTPVM5 - RMTPVM5 PVM *
*****
START LINE 600
START NODE RMTPVM1D
START GRAF 490
(etc.)
```

Figure 18. PVM Profile for RMTPVM5 -- RMTPVM5 PVM. Sample for a target node routed through an intermediate node.





---

## Chapter 6. Defining A PVMG Configuration

Before starting PVMG, the system administrator should plan where the PVMG server fits into the PVM network, and create a PVMG CONFIG configuration file that defines the PVMG configuration. Create a PVMG PROFILE to automate PVMG initialization.

There are other requirements and conditions to consider before planning a PVMG configuration. See Chapter 3, “Administrative Considerations” on page 15 for this information.

---

### Planning A PVMG Configuration

Use the PVMG configuration file on the PVMG user ID to define the PVMG connections to VTAM and PVM. The PVMG CONFIG file contains lines called records. The records tell PVMG how PVM and SNA applications are connected. The PVMG CONFIG file is processed during PVMG initialization. There are a few restrictions regarding the configuration file:

- It is a fixed length 80 byte file
- Except for comments, and session termination string values on the SESSTERM record, all input must be in uppercase
- Input for the file's records is restricted to a single line, and there is no continuation
- Operands must be separated by one or more blanks
- The file type must be CONFIG.

**Note:** In order to tailor PVMG files, log onto PVMG and establish R/W access to PVMG's 191 minidisk (PVMG links to P684100E's 402 minidisk as 191 in MR mode, through the CP directory). It may be necessary to detach the 402 minidisk from P684100E to gain write access, because M/R mode only allows one R/W link at a time. Once logged on, shutdown PVMG (if running) by issuing:

PVMG SHUTDOWN IMMED

**Note:** When entering PVMG commands from the PVMG user ID, prefix PVMG commands with pvmg.

Now tailor:

- PVMG CONFIG
- PROFILE GCS.

## Defining PVMG

The following table describes the sample files supplied on the PVM 2.1.1 LOCALSAM (2C2) disk for the PVMG user id:

<i>Table 3. PVM 2.1.1 Supplied Sample Files for the PVMG user id</i>	
File Name	Sample Description
PROFILE GCSSAMP	Sample profile GCS file.
PVMG CONFSAMP	Sample PVMG CONFIG file.
PVMG VTAMLST	Sample VTAMLST file for the PVMG component.
PVMGSAMP DIRECT	Sample directory file.

## Planning the PVMG Server's Role in the PVM Network

<i>Table 4. Configuration File Planning Table Number 1. Defines PVMG links — systems connected to the local PVMG.</i>		
Target Name	Connection Type and Address	Comments
RMTPVM1	IUCV	This is the link to RMTPVM1

<i>Table 5. Configuration File Planning Table Number 2. Defines PVMG routes to other systems in the network.</i>		
Target Name	Route Session Requests Through	Comments
RMTSYS6	PVMG1	Defines the connection to N2N6AVSC through VTAM

Table 6 on page 43, Table 8 on page 43, and Figure 19 on page 43, illustrate a sample PVMG configuration that is defined in this chapter. Create similar tables and diagrams to help configure the PVM/PVMG network .

```

LANG AMENG
OPID OPERATOR
PVMPROP OPERATOR
LOCAL PVMG1 N2ECAPV2
LINK * RMTPVM1 IUCV USER PVM
LOGONID N2N6AVSC
SELECT RMTPVM2
* Route Table:
ROUTE RMTPVM2 RMTPVM1
ROUTE DEFAULT RMTPVM1
AUTHORIZ OPERATOR *
SESSDISC N2ECAVSC
SESSTERM #####
XDBC CONS
*N2N6AVSC
LU N2N6AVSC N2ECAP01-20 RESTRICT
* APPLID TABLE
APPLN RMTSYS6 N2N6AVSC
* Logon Mode Table Entry Name Keywords and Values:
LOGMODE 3277 S3270
LOGMODE 3278-2B D4B32782
LOGMODE 3278-2E D4E32782
LOGMODE 3278-3B D4B32783
LOGMODE 3278-3E D4E32783
LOGMODE 3278-4B D4B32784
LOGMODE 3278-4E D4E32784
LOGMODE 3278-5B D4B32785
LOGMODE 3278-5E D4E32785
LOGMODE 3279-2B D4B32792
LOGMODE 3279-2E D4E32792
LOGMODE 3279-3B D4B32793
LOGMODE 3279-3E D4E32793
LOGMODE 3290 D329001
DUMP VMDUMP OPERATOR
MSGOPTNS Y Y Y Y Y Y Y Y

```

Figure 19. Sample Configuration File for PVMG1

Table 6. Configuration File Planning Table Number 3. Defines the PVMG link to the local PVM.

Line Address	Node ID	Driver Type	User	VM ID	Comments
*	RMTPVM1	IUCV	USER	PVM	This is the link to the local PVM node.

The following tables contain detailed information about the records in the sample PVMG configuration file above.

Table 7. PVMG Configuration File Planning Table Number 4. Defines the VTAM APPLID to which the PVM terminals log on.

APPLID	APPLICATION	Comments
VTAM	ASC	See the LOGONID configuration record

Table 8. PVMG Configuration File Planning Table Number 5. Defines the Routing from PVMG Nodes to Target Nodes

Target ID	Next ID	Comments
DEFAULT	RMTPVM1	See the ROUTE configuration record

## Defining PVMG

*Table 9. PVMG Configuration File Planning Table Number 6. Specifying the VTAM Logon Mode Table Entries Used by PVMG*

Entry Description	Entry Name	Comments
3277	S3270	3270 LOCAL NON-SNA
3278-2B	D4B32782	3274 MODEL 1B/1D WITH MODEL 2 SCREEN (LOCAL NON-SNA) PRIMARY SCREEN 24X80 (1920) NO ALTERNATE SCREEN DEFINED
3278-2E	D4E32782	3274 MODEL 1B/1D WITH MODEL 2 SCREEN (LOCAL NON-SNA) PRIMARY SCREEN 24X80 (1920) NO ALTERNATE SCREEN DEFINED
3278-3B	D4B32783	3274 MODEL 1B/1D WITH MODEL 3 SCREEN (LOCAL NON-SNA) PRIMARY SCREEN 24X80 (1920) ALTERNATE SCREEN 32X80 (2560)
3278-3E	D4E32783	3274 MODEL 1B/1D WITH MODEL 3 SCREEN (LOCAL NON-SNA) PRIMARY SCREEN 24X80 (1920) ALTERNATE SCREEN 32X80 (2560)
3278-4B	D4B32784	3274 MODEL 1B/1D WITH MODEL 4 SCREEN (LOCAL NON-SNA) PRIMARY SCREEN 24X80 (1920) ALTERNATE SCREEN 43X80 (3440)
3278-4E	D4E32784	3274 MODEL 1B/1D WITH MODEL 4 SCREEN (LOCAL NON-SNA) PRIMARY SCREEN 24X80 (1920) ALTERNATE SCREEN 43X80 (3440)
3278-5B	D4B32785	3274 MODEL 1B/1D WITH MODEL 5 SCREEN (LOCAL NON-SNA) PRIMARY SCREEN 24X80 (1920) ALTERNATE SCREEN 27X132 (3564)
3278-5E	D4E32785	3274 MODEL 1B/1D WITH MODEL 5 SCREEN (LOCAL NON-SNA) PRIMARY SCREEN 24X80 (1920) ALTERNATE SCREEN 27X132 (3564)
3279-2B	D4B32792	3279 MODEL 2B - NO EDS PRIMARY SCREEN 24X80 (1920) ALTERNATE SCREEN 24X80 (1920)
3279-2E	D4E32792	3279 MODEL 2B - WITH EDS PRIMARY SCREEN 24X80 (1920) ALTERNATE SCREEN 24X80 (1920)
3278-3B	D4B32793	3279 MODEL 3 - NO EDS PRIMARY SCREEN 24X80 (1920) ALTERNATE SCREEN 32X80 (2560)
3279-3E	D4E32793	3279 MODEL 3 - WITH EDS PRIMARY SCREEN 24X80 (1920) ALTERNATE SCREEN 32X80 (2560)
3290	D329001	LOGMODE TABLE ENTRY FOR 3290 TERMINAL PRIMARY SCREEN SIZE 24X80 ALTERNATE SCREEN SIZE 62X160

## Commenting the PVMG Configuration File

Use the comment (\*) record to put comments in the configuration file. For example,

\* Configuration file for node PVMG1

is the first record of the configuration file for PVMG1. The asterisk (\*) must be in column 1.

Comment records can be placed anywhere in the configuration file.

## Identifying the System Language

Use the LANG record to identify the default national language for a node. For example,

```
LANG AMENG
```

assigns American English as the default national language for PVMG1.

The LANG record should be placed at the top of the PVMG configuration file.

## Excluding Double-Byte Character Messages

Use the XDBC record to exclude users from receiving DBCS messages. For example, if the LANG record identifies a double-byte character language such as KANJI and the PVMG console should not receive DBCS message, enter the following:

```
XDBC CONS
```

Now the PVMG console receives any DBCS messages in the backup language.

See “QUERY” on page 211 for information on the DBCS settings and “CDBC” on page 185 for information on changing DBCS-exclusion settings.

## Identifying the Local Node Name and the VTAM Application Name

Use the LOCAL record to identify the local PVMG node name. For example,

```
LOCAL PVMG1
```

assigns the node ID (PVMG1) to the local PVMG system.

```
LOCAL PVMG1 N2ECAPV2
```

assigns a node ID (PVMG1) and VTAM application name (N2ECAPV2) to the local PVMG system.

## Defining a Link from PVMG to PVM

Use the LINK record to define a link for connecting with the PVM virtual machine. LINK records must follow the LOCAL records in the configuration file. The following records define a LINK from node RMTPVM1.

```
LINK * RMTPVM1 IUCV USER PVM
```

is the link defined from node PVMG1 to RMTPVM1.

**Note:** If the link on the PVM side is not active, PVMG will automatically retry the IUCV connect at 60 seconds intervals until either communications is established, or a PVMG DROP node command is issued by an operator.

## Defining the VTAM APPLID to Which Users Log on

Use the LOGONID record to define the VTAM APPLID to which the PVMG node is to log the PVM terminals if they access a target application using the PVMG node ID.

```
LOGONID N2N6AVSC
```

### Assigning Node Names to PF Keys

Use the SELECT record to assign frequently used node names to PF keys. An installation can define up to eleven PF keys in the PVMG configuration file.

The first SELECT record *nodeid* is assigned to PF1, the second SELECT record *nodeid* is assigned to PF2, and so forth to PF7. The eighth SELECT record is assigned to PF9 (PF8 is reserved for the SCROLL FORWARD function). The ninth SELECT record is assigned to PF10, and so forth to PF12.

If a SELECT record does not specify the node ID, the PF key associated with that SELECT record is omitted. This allows a PF key to be skipped.

```
SELECT RMTPVM2
SELECT RMTPVM3
SELECT RMTPVM4
```

### Defining Routes and Alternate Routes

Use the route record to define the routing path from the local PVMG node to a specified target node. For PVMG to PVM communications, this route must be consistently defined on both ends of network paths.

Use the second form of the route command to define default routing for those nodes not defined by LINK or ROUTE records. Only one default route record can be entered.

```
ROUTE DEFAULT RMTPVM1
```

### Specifying LU Names Used by PVMG with Secure VTAM APPLIDs

To create a secure LU session to a VTAM application, specify a nodename and user ID on the LU record (see “LU” on page 290).

An example of a secure LU would be:

```
LU vtamapplid RMTPVM1 SMITH
```

Only user ID SMITH at RMTPVM1 would have access to *vtamapplid*.

To create a range of restricted LU sessions to the VTAM network, specify RESTRICT on the LU record.

An example of a range of restricted LUs would be:

```
LU N2N6AVSC N2ECAP01-20 RESTRICT
```

This causes 20 logical units to be generated. Only one session may be assigned to a LU.

### Authorizing PVMG Operators

Use the AUTHORIZ record to identify users allowed to enter the restricted subset of PVMG commands and the nodes from which these commands may be issued. For example,

```
AUTHORIZ OPR
AUTHORIZ JONES RMTPVM1
```

authorizes the person identified by the user ID OPR to enter restricted PVMG commands from the local node, PVMG1. The second AUTHORIZ statement authorizes user JONES at RMTPVM1 to enter restricted PVMG commands.

## Saving PVMG Information

The DUMP record saves information about PVMG. Use the DUMP record to specify the type of dump to be taken if abnormal termination of PVMG occurs. For example, the record:

```
DUMP VMDUMP
```

causes a VMDUMP dump to be taken of all PVMG storage.

```
DUMP CP JONES
```

causes a CP DUMP to be taken of all PVMG storage, and sent to user ID JONES.

## Specifying Message Requirements

Use the MSGNOH and the MSGOPTNS records to indicate how and where messages are displayed. Use the MSGNOH record to indicate that responses to all commands entered by the CP SMSG command must be returned using the CP MSGNOH command rather than the CP MSG command.

If there is more than one MSGNOH record in the PVMG configuration file, only the last MSGNOH record in the file is accepted. The other MSGNOH records are ignored.

```
MSGNOH
```

Use the MSGOPTNS record to determine if the time stamps or message numbers for messages will be directed to the PVMG user ID, the users, and/or the operator. For example, the record:

```
MSGOPTNS Y Y
```

tells PVMG to specify the time stamps and the message numbers on the messages to the PVMG user ID.

## Specifying the Target User ID for Natural Language Message Reception

Use the OPID record to specify the virtual machine ID that receives messages that usually go to the system operator. This record can also be used to stop sending messages to the system operator. For example, the following record forwards all messages to the JONES user ID:

```
OPID JONES
```

The following example prevents messages from being sent to a CP, or alternate, operator.

```
OPID NONE
```

## Specifying the User ID of the Programmed Operator Virtual Machine That Receives Encoded Messages

Use the PVMPROP record to identify a virtual machine that receives a copy of PVMG error messages. All messages are transmitted to the virtual machine by the CP Special Message (SMSG) facility. The messages are sent to the PVMPROP virtual machine in the same format as those sent to the PVM Command Response Interface (CRI). For example, the following records would cause all PVMG messages to be sent in encoded form by the CP SMSG command to the PROP user ID:

```
PVMPROP PROP
```

## Specifying the VTAM APPLID to Disconnect When PVMG Fails

Use the SESSDISC record to indicate the VTAM APPLID from which user sessions should be disconnected (rather than stop the sessions) when an unexpected PVMG problem occurs.

## Defining PVMG

SESSDISC N2ECAVSC

will disconnect the APPLID when an unexpected PVMG problem occurs.

## Changing the Default Local PVMG Node Session Termination String

Use the SESSTERM record to change the default session termination string used at the local PVMG node.

For example, to set the session terminate string to #####, specify:

SESSTERM #####

## Specifying VTAM Logon Model Table Entries Used by PVMG

Use the LOGMODE record to specify VTAM logmode table entries that are used by PVMG when establishing interactive sessions.

\* Logon Mode Table Entry Name Keywords and Values:

```
LOGMODE 3277      S3270
LOGMODE 3278-2B   D4B32782
LOGMODE 3278-2E   D4E32782
LOGMODE 3278-3B   D4B32783
LOGMODE 3278-3E   D4E32783
LOGMODE 3278-4B   D4B32784
LOGMODE 3278-4E   D4E32784
LOGMODE 3278-5B   D4B32785
LOGMODE 3278-5E   D4E32785
LOGMODE 3279-2B   D4B32792
LOGMODE 3279-2E   D4E32792
LOGMODE 3279-3B   D4B32793
LOGMODE 3279-3E   D4E32793
LOGMODE 3290      D329001
```

---

## Tailoring the PROFILE GCS File

A PROFILE GCS file is needed to IPL GCS on the PVMG server virtual machine.

Tailor the PROFILE GCSSAMP file that was installed when PVMG was installed. Use the following sample file as a guide.

```
/* */
'CP SET RUN OFF'
'FILEDEF PVMGCONFIG DISK PVMG CONFIG *'
'CP SET IMSG ON'
'GLOBAL LOADLIB PVMG'
'LOADCMD PVMG EFGMAN'
'PVMG INIT'
```

Figure 20. Sample PVMG Profile GCS



## Chapter 7. Developing NAMES Files for MPVM

After MPVM is installed, set up the MPVMSYS NAMES file to reflect the nicknames and characteristics of the sessions the installation will be using. The MPVMSYS NAMES file contains entries, and each entry consists of a nickname and subtags. This chapter provides details on what the system names file should contain.

**Note:** Rename the MPVMSYS NAMESAMP file to MPVMSYS NAMES and use it as sample NAMES file.

Also, an MPVM user can create a personal NAMES file, called MPVM NAMES. Users might need their own file, for instance, when they want to specify nicknames that are not in the MPVMSYS NAMES file, or when they want to override entries defined in MPVMSYS NAMES. The end of this chapter contains more detail about user's name files.

Use the MPVMSYS NAMES file, which should now be as a base for creating session entries. Define entries for target applications and special entries for the overall MPVM environment. The following table describes the sample files supplied on the PVM 2.1.1 LOCALSAM (2C2) disk for MPVM users:

<i>Table 10. PVM 2.1.1 Supplied Sample Files for MPVM users</i>	
File Name	Sample Description
MPVM NAMESAMP	Sample user level MPVM names file.
MPVMSGNS NAMESAMP	Sample system level MPVM Auto-Signon names file.
MPVMSYS NAMESAMP	Sample system level MPVM names file.

Here is an overview of what is contained in this chapter:

- Developing MPVMSYS NAMES file entries that define PVM target applications
- Developing MPVMSYS NAMES file entries that define configuration information for MPVM
- Special, miscellaneous MPVMSYS NAMES file entries
- Verifying the completed MPVM installation
- How individual MPVM NAMES files relate to the system-level MPVMSYS NAMES file.

Plus, starting on page 53, there is a sample MPVMSYS NAMES file that summarizes many of the tags that can be used.

### Creating Entries for Applications

MPVM gets its instructions from the values assigned to entries in the MPVMSYS NAMES file. Each target application must have a nickname; specified by the :nick. tag. Specify the characteristics of the nickname using subtags under the :nick..

### Specifying the Nickname

When users want to access applications through PVM, they specify a nickname from either the MPVM selection screen or the MPVM command or macro. Each nickname entry, specified by the :nick. tag, defines the target application to be accessed.

Create a nickname entry for each application that MPVM users can access. The value specified by the nickname:

## System NAMES File

- Must be unique
- Can be up to eight characters
- Must contain only the characters a-z, A-Z, 0-9, or one of the following special characters:  
  . , \_ : % & @ # ?
- Cannot be the word *CANCEL*
- Cannot start with an escape (default value ?) character

For example, if an application updates customer billing information, it might have a nickname of *BILLING*. The nickname entry in the MPVMSYS NAMES file would look like:

```
:nick.BILLING
```

Even if nicknames contain lower case characters, they are converted to uppercase during processing.

## Specifying Characteristics for Application Nicknames

For MPVM to establish a session with this target application, however, it needs to know more than just a user's nickname. MPVM must have specific session characteristics. PVM gets this information from values associated with the nickname. The values correspond to MPVM variables.

How are these variables set in a NAMES file? With subtags that look the same as the MPVM variable name. There are some examples in the next few sections that describe use of variables.

## Defining Target Applications

Use the **:target.** subtag in the NAMES file to specify the name of a target system. So, for example, if VMSYS1 is the name of a target system, a subtag/value pair could be specified like:

```
:target.VMSYS1
```

:target. must be specified as:

- :target subtag in a nickname entry
- allowing it to default to the :target subtag in the ?default nickname
- allowing it to default to the nickname.

## Defining Other Variables for Applications

There are several other MPVM variables that can be specified for an application nickname. Any MPVM variable that can be set (with a SET command, or directly on the MPVM command or macro) can be specified on a NAMES file subtag. Some of the most common examples are PF key and PA key settings. See the *VM/Pass-Through Facility: Programming Reference* for information on all MPVM variables.

## Specifying Commands to be Invoked

Besides specifying a subtag/value pair that corresponds to setting an MPVM variable, a string can be specified for MPVM to directly start. Do this using a special **:?** tag. This can be used, for example, to have MPVM process a macro when it processes a particular nickname. MPVM processes the command on this tag completely before moving onto the next tag in the NAMES file. For example, **:?.MSG HI THERE!** sends the message HI THERE! to the nickname being processed.

## Specifying custom Subtags

While MPVM is processing an application nickname, it ignores any subtags that begin with a character other than A-Z, a-z, \* or ?. This allows creating custom subtags. For example, a subtag called **:\$DATE.** could be created that specifies when a particular entry was added to the MPVMSYS NAMES file. The MPVM NICKINFO command could be used to search the MPVMSYS NAMES file for this subtag.

## Specifying Nickname Lists

An individual nickname tag could specify the name of a *list* of nicknames rather than contain subtag/value pairs. The list name must be prefix by a \* character on the :nick. tag. The :nick. tag must be followed by a special :\* subtag, which contains the individual nicknames in the list.

For example, suppose there where nicknames PROFS, LOCAL1, and CUSTADR for three applications. A single list name called ALLSYS could be defined to specify the NAMES file entry like this:

```
:nick.*ALLSYS
    :*PROFS LOCAL1 CUSTADR
```

Selecting \*ALLSYS, selects all three applications.

## Summary of Application NAMES File Entries

Here is a summary of the steps to create a NAMES file for MPVM:

1. XEDIT the sample NAMES file for MPVM. (As distributed, this file was named MPVMSYS NAMESAMP; during installation it was renamed it to MPVMSYS NAMES.)
2. Determine all the applications users are permitted to access through MPVM, and where those applications are located.

## System NAMES File

3. For each application listed in step 2 on page 51, do the following steps:

a. Enter these two mandatory tags with associated values:

Tag	Description
-----	-------------

:nick.	A nickname for the application that users will specify, either from the MPVM Session Selection Menu, or as an operand on the MPVM command or macro.
--------	---

:target.	The target system name associated with this session.
----------	--

b. Look through the MPVM variable list and determine what other characteristics to assign to the MPVM session.

4. Place the MPVMSYS NAMES file on the MPVM common disk, and verify that all entries for remote applications work. Each entry nickname should be separately checked to verify that it works properly.

Up to 999 nickname entries can be defined in an MPVMSYS or MPVM NAMES file, with up to 999 subtags per nickname entry.

---

## Defining Configuration Settings—:nick.?CONFIG

The MPVMSYS NAMES file does more than just define information for application nicknames—it includes some special nickname values that define important MPVM characteristics. ?CONFIG is a special value on the nickname tag (:nick.?CONFIG) that is processed during MPVM initialization and defines configuration settings.

## User Initialization Settings

Use the following subtag to define the user ID of the PVM virtual machine:

:pvmid.

The subtag can be in upper, lower, or mixed case. If this subtag is omitted, the user ID of the PVM virtual machine defaults to *PVM*.

---

## Creating Other Special Entries

MPVM allows other special nicknames that start with a question mark (?).

## Setting Global Values for All Sessions—:nick.?INITIAL

?INITIAL is a special value on the :nick. tag that is processed only once, when MPVM is first started, before any sessions are established. Specify subtags for ?INITIAL to set values for all sessions. However, only global variables, which are common to all sessions, can be set. A :?. tag can also be specified to process a given string once, before any sessions are established.

## Setting Default Values for All Sessions—:nick.?DEFAULT

?DEFAULT is a special value on the :nick. tag that is processed whenever a new session is established, before an individual system nickname is processed. ?DEFAULT can be used to set local *and* global default values. A :?. tag can also be specified to process a given string before any new session is established.

## Cleaning Up After the Last Session Has Ended—:nick.?FINAL

?FINAL is a special value on the :nick. tag that is processed whenever the last MPVM session has ended, before the main MPVM module (EFGVIEW) terminates. ?FINAL can be used to undo some things originally set up when establishing the MPVM environment (perhaps on the :?INITIAL tag). The subtag typically used on ?FINAL is the :?.

---

## Sample MPVMSYS NAMES File

The following page shows an example of typical nickname entries and subtags in an MPVMSYS NAMES file.

```
* The ?CONFIG entry contains the PVM virtual machine user ID
:nick.?CONFIG
    :pvmid.PVM
*
*
* The ?INITIAL entry is processed when MPVM is first invoked, providing
* global settings for all MPVM sessions. This example defines the SYSREQ key
* to process the SELECT command (to roll to the next session).
:nick.?INITIAL
    :sysreq.SELECT
*
* The ?DEFAULT entry is processed when each new MPVM session is established.
* This example defines the PF10 key to display the previously-issued command,
* and defines the default target as RMTPVM1.
:nick.?DEFAULT
    :pf10.RETRIEVE
    :target.RMTPVM1
    :nick.?FINAL
    :?.MSG YOU ARE LEAVING MPVM
* APPLICATION DEFINITIONS
*
* This defines a session on the local system.
* The MSG command is processed directly, displaying the indicated message.
:nick.LOCAL1
    :target.RMTPVM1
    :?.MSG This text gets displayed
*
* This example defines a session called PROFS; the PROFS* target application is on a system
* named WESTVM1.
:nick.PROFS
    :target.WESTVM1
*
* This example defines a session called CUSTADDR; the CUSTADDR target application is on a
* system named MVSSYST in an SNA network.
* This example also shows how you can remove the MPVM
* definition for PF10 so that the target application's PF10 definition is used.
:nick.CUSTADDR
    :target.MVSSYST
    :pf10.
*
* This example shows how to specify a nickname list: A user issuing MPVM *ALLSYS
* automatically gets 3 sessions established.
:nick.*ALLSYS
    :*.PROFS LOCAL1 CUSTADDR
```

Figure 21. Sample MPVMSYS NAMES File

---

### Verifying the MPVMSYS NAMES File

After creating the MPVMSYS NAMES file, verify that it works properly.

Log onto another user ID and enter:

MPVM

An MPVM Selection Screen should be displayed. Try each nickname entry to verify that session characteristics are working as intended.

---

### Considerations for Individual User NAMES Files

As mentioned, users can also have a personal NAMES file to define information about their MPVM applications. When MPVM resolves a nickname, the user-level MPVM NAMES file (if it exists) is checked first. If the user-level file does not contain the specified nickname, CMS searches the system-level NAMES file. If a nickname is defined in both the MPVM NAMES file and the MPVMSYS NAMES file, the information in the MPVM NAMES file takes precedence.

A user's NAMES file can include the special ?INITIAL, ?DEFAULT, and ?FINAL nickname entries. Subtags here are added to any subtags in the ?INITIAL, ?DEFAULT, and ?FINAL entries in MPVMSYS NAMES.

User files can have their own ?CONFIG entry and subtags. The only subtag that MPVM will look at, however, is :pvmid.. All other ?CONFIG subtags are ignored.

The MPVM Session Selection Menu includes a convenient feature for users working on their own NAMES file: Go to the command line of the menu and edit the NAMES file (?x mpvm names), then add a nickname entry and file it to cause the new nickname to be added to the menu, in its proper order.

As an alternative to specifying subtag/value pairs in the MPVM NAMES file, users can also specify them:

- As operands on the MPVM command or macro call
- From the command line on the MPVM Session Selection Menu.

## Summary of MPVMSYS NAMES Contents

The following table summarizes all the tags that can be used in MPVM's system-level NAMES file, called MPVMSYS NAMES:

Table 11. Partial Contents of the MPVMSYS NAMES File	
Nickname	Usage and Subtags
: <b>nick</b> . <i>appl_nickname</i>	<p>This is a typical nickname value. <i>appl_nickname</i> is a nickname set up for the application or system that users want to establish sessions with.</p> <p>:<i>varname.value</i>. Sets values for each session. The subtag corresponds directly to an MPVM variable name. Some examples of variables that can be set as a subtag are:</p> <ul style="list-style-type: none"> <li>:target. Node ID for the system MPVM is accessing.</li> <li>:route. PVM secondary route.</li> <li>:cinit. user data for sessions connected through SNA.</li> <li>:logmode. SNA logmode table name that the PVMG server should use.</li> <li>:PF<i>nn</i>. command string processed when the PF<i>nn</i> key is pressed.</li> <li>:columns. number of columns for the session screen.</li> <li>:rows. number of rows for the session screen.</li> <li>:signon. command string processed during an installation-written automatic signon routine.</li> <li>:initcmd command string processed when a session with a target system begins.</li> </ul> <p><b>Note:</b> All local variables that are settable by the MPVM user can be used as subtags in this context. See the <i>VM/Pass-Through Facility: User's Guide</i> for a complete list of MPVM variables.</p> <p>:<b>?.</b><i>executable_string</i> is a special tag telling MPVM to run the string as it is processed. This can also be used with special ?INITIAL and ?DEFAULT nicknames, and with ?FINAL.</p> <p>:<b>\$</b><i>anystring.anymeaning</i> Creates custom tags. Actually, the first character does not have to be a dollar sign; MPVM ignores any character other than A-Z, a-z, or ?.</p>
: <b>nick</b> . <i>*listname</i>	<p><i>listname</i> is a special nickname that points to a list of other nicknames. :<b>nick</b>.<i>*listname</i> must be followed by the :*. subtag, in this format:</p> <p>:*.<i>appl_nickname1 appl_nickname2 ...</i></p>
: <b>nick</b> . <b>?CONFIG</b>	<p>This special nickname value is processed once, when MPVM is first started. ?CONFIG has one special, reserved subtag:</p> <p>:<b>pvmid</b>. defines the user ID of the virtual machine containing the PVM server.</p>
: <b>nick</b> . <b>?INITIAL</b>	<p>This special nickname value is processed once, when MPVM is first invoked. ?INITIAL subtags can be used to set values for all sessions. Only global variables can be set.</p>
: <b>nick</b> . <b>?DEFAULT</b>	<p>This special nickname value is processed whenever a new session is established, before an individual system nickname is processed, and can be used to specify local and global default values.</p>
: <b>nick</b> . <b>?FINAL</b>	<p>This special nickname value is processed whenever the last MPVM session is terminated and MPVM is getting ready to end. This can be used for some final <i>clean up</i> activity.</p>





---

## Chapter 8. Operating the Network

This chapter contains descriptions the PVM commands that are used to operate the PVM network. Operating includes tasks such as starting, querying, and stopping the PVM network, quiescing, resuming, and dropping devices and users, and communicating with the network's users.

To automate the operations of a PVM network, see Chapter 13, "Facilities for Automating PVM and PVMG Operations" on page 145.

The following is based on the network defined in Chapter 5, "Defining A PVM Configuration." The chapter shows sample uses of commands and the output or messages that appear at the PVM operator's display station. All commands are fully described in Appendix B, "PVM Commands" and Appendix D, "PVMG Commands."

---

### Starting PVM

Once the network is defined in the configuration file and the PVM profile and PROFILE EXEC are created, enter the RUNPVM command. The RUNPVM command loads and starts the PVM module, which uses the configuration file to define the system setup. Once PVM is up, the PVM profile is called. The RUNPVM command can be started from the CMS PROFILE EXEC entered from a CMS command line.

If the default names are used for the configuration file (PVM CONFIG), the PVM profile (PROFILE PVM), and the PVM module (PVM MODULE), load and start the PVM module by issuing:

```
runpvm
```

However, in this scenario, to process RMTPVM1 CONFIG (the configuration file for RMTPVM1) and RMTPVM1 PVM (the PVM profile for RMTPVM1) and to load and start the PVM module, enter

```
runpvm * rmtpvml rmtpvml
```

## Operating the Network

When PVM starts, it produces messages similar to the following:

```
DMSLI0740I Execution begins...
03/31/93 09:41:38 DVMBFR040I VM Pass-Through Facility 2.1.1000, built 03/30/93
15:27:08, is ready
03/31/93 09:41:40 DVMCON050I Your EXEC command is complete
03/31/93 09:41:40 DVMCON050I *****
03/31/93 09:41:40 DVMCON050I * PVM Profile Exec for node RMTTPVM1 - RMTTPVM1 PVM *
03/31/93 09:41:40 DVMCON050I *****
03/31/93 09:41:40 DVMCON050I START LINE 200
03/31/93 09:41:40 DVMCON050I START LINE 201
03/31/93 09:41:40 DVMCON050I START LINE 500
03/31/93 09:41:40 DVMCON050I START NODE RMTTPVM5B
03/31/93 09:41:40 DVMCON050I START NODE PVMG1
03/31/93 09:41:40 DVMCON050I START GRAF 450
03/31/93 09:41:40 DVMCON050I START GRAF 451
03/31/93 09:41:40 DVMCON050I START GRAF 452
03/31/93 09:41:40 DVMCON050I START GRAF 453
03/31/93 09:41:40 DVMCON050I START GRAF 454
03/31/93 09:41:40 DVMCON068I GRAF 450 DEFINED
03/31/93 09:41:40 DVMNET318I GRAF450 is starting
03/31/93 09:41:40 DVMCON068I GRAF 451 DEFINED
03/31/93 09:41:40 DVMNET318I GRAF451 is starting
03/31/93 09:41:40 DVMCON068I GRAF 452 DEFINED
03/31/93 09:41:40 DVMNET318I GRAF452 is starting
03/31/93 09:41:40 DVMCON068I GRAF 453 DEFINED
03/31/93 09:41:40 DVMNET318I GRAF453 is starting
03/31/93 09:41:40 DVMCON068I GRAF 454 DEFINED
03/31/93 09:41:40 DVMNET318I GRAF454 is starting
03/31/93 09:41:40 DVMCON068I DEVICE RESET
03/31/93 09:41:40 DVMCON068I DEVICE RESET
03/31/93 09:41:40 DVMCON068I DEVICE RESET
03/31/93 09:41:40 DVMCON068I DEVICE RESET
03/31/93 09:41:40 DVMCON068I DEVICE RESET
```

Now, suppose RMTVPVM1 CONFIG contains errors. After issuing the following command,

```
runpvm * rmtvpvm1 rmtvpvm1
```

it will produce messages similar to the following:

```
DMSLI0740I Execution begins...
DVMDIR016E The LINK type is not valid
DVMDIR010E LINK 300 VSEAPL 327X
DVMDIR011E Invalid configuration type record
DVMDIR010E PORT 0 3278-2 RMTVPVM1 USERA
DVMDIR011E Invalid configuration type record
DVMDIR010E PORT 1 3278-2 RMTVPVM1 0351
DVMDIR011E Invalid configuration type record
DVMDIR010E PORT 8 3278-2 RMTVPVM1 USERB
DVMDIR011E Invalid configuration type record
DVMDIR010E PORT 9 328X RMTVPVM1 0360
DVMDIR012E The configuration record is out of order
DVMDIR010E AUTHORIZ OPR
DVMDIR012E The configuration record is out of order
DVMDIR010E AUTHORIZ JONES RMTVPVM2
DVMDIR821E SELECT node is not valid
DVMDIR010E SELECT VSEAPL
03/31/93 10:26:20 DVMBFR040I VM Pass-Through Facility 2.1.1000, built 03/30/93
15:45:18, is ready
03/31/93 10:26:22 DVMCOS001I Your EXEC command is complete
03/31/93 10:26:22 DVMCON050I *****
03/31/93 10:26:22 DVMCON050I * PVM Profile Exec for node RMTVPVM1 - RMTVPVM1 PVM *
03/31/93 10:26:22 DVMCON050I *****
03/31/93 10:26:22 DVMCON050I START LINE 200
03/31/93 10:26:22 DVMCON050I START LINE 201
03/31/93 10:26:22 DVMCON050I START LINE 300
03/31/93 10:26:22 DVMCON052E Invalid address "300" on START LINE
03/31/93 10:26:22 DVMCON050I START LINE 500
03/31/93 10:26:22 DVMCON050I START NODE RMTVPVM5B
03/31/93 10:26:22 DVMCON050I START NODE PVMG1
03/31/93 10:26:22 DVMCON050I START GRAF 450
03/31/93 10:26:22 DVMCON050I START GRAF 451
03/31/93 10:26:22 DVMCON050I START GRAF 452
03/31/93 10:26:22 DVMCON050I START GRAF 453
03/31/93 10:26:22 DVMCON050I START GRAF 454
03/31/93 10:26:22 DVMCON068I GRAF 450 DEFINED
03/31/93 10:26:22 DVMNET318I GRAF450 is starting
03/31/93 10:26:22 DVMCON068I GRAF 451 DEFINED
03/31/93 10:26:22 DVMNET318I GRAF451 is starting
03/31/93 10:26:22 DVMCON068I GRAF 452 DEFINED
03/31/93 10:26:22 DVMNET318I GRAF452 is starting
03/31/93 10:26:22 DVMCON068I GRAF 453 DEFINED
03/31/93 10:26:22 DVMNET318I GRAF453 is starting
03/31/93 10:26:22 DVMCON068I GRAF 454 DEFINED
03/31/93 10:26:22 DVMNET318I GRAF454 is starting
03/31/93 10:26:22 DVMCON068I DEVICE RESET
03/31/93 10:26:22 DVMCON068I DEVICE RESET
03/31/93 10:26:22 DVMCON068I DEVICE RESET
03/31/93 10:26:22 DVMCON068I DEVICE RESET
03/31/93 10:26:22 DVMCON068I DEVICE RESET
```

The following list contains the errors in the configuration file and corrections that need to be made.

- DVMDIR016E The LINK type is not valid  
With the APAR for VM66733, 327X link type are obsolete and are no longer supported.
- DVMDIR011E Invalid configuration type record  
With the APAR for VM66733, 327X link type and associated PORT statements are obsolete and are no longer supported.
- The configuration record is out of order

## Operating the Network

The AUTHORIZ OPR and AUTHORIZ JONES RMTVM2 records are out of order. The configuration file records were arranged as follows:

```
:  
ROUTE DEFAULT RMTVM3  
AUDIT DATA RECORD  
* The configuration file records to this point are required to  
* define all routes from RMTVM1 to the other nodes.  
* The following records might apply to any PVM node.  
AUTHORIZ OPR  
AUTHORIZ JONES RMTVM2  
SUBSET LISTFILE  
LNODE * AUTO  
:
```

All AUTHORIZ records must appear before the AUDIT DATA RECORD in the configuration file. Move the AUTHORIZ OPR and AUTHORIZ JONES RMTVM2 before the AUDIT DATA RECORD record. The configuration file records should be arranged as follows:

```
:  
ROUTE DEFAULT RMTVM3  
* The configuration file records to this point are required to  
* define all routes from RMTVM1 to the other nodes.  
* The following records might apply to any PVM node.  
AUTHORIZ OPR  
AUTHORIZ JONES RMTVM2  
SUBSET LISTFILE  
AUDIT DATA RECORD  
LNODE * AUTO  
:
```

- SELECT node is not valid

The SELECT VSEAPL record is not valid since the 3270x link type is no longer supported as of APAR VM66733.

The following list contains the error statements from the PVM profile.

- Invalid address 300 on START LINE

Since address 300 was not valid in the LINK record in the configuration file, the START LINE 300 record could not execute.

## Querying the System

A PVM operator can determine the status of the system, lines, devices, and users.

To see the information about the system, such as, the number of links on the system, the link names, group names, or the number of users on the system, enter:

```
q system
```

This will produce messages similar to the following:

```
03/31/93 09:43:41 DVMCON050I Q SYSTEM
03/31/93 09:43:41 DVMQRY080I The local node is RMTPVM1, Users 0, PVM 2.1.1000,
built 03/30/93 15:27:08
03/31/93 09:43:41 DVMQRY081I Link RMTPVM2A Type CTCA      Address  200      Users
0 CONNECT Group RMTPVM2
03/31/93 09:43:41 DVMQRY081I Link RMTPVM2B Type CTCA      Address  201      Users
0 CONNECT Group RMTPVM2
03/31/93 09:43:41 DVMC39081I Link RMTPVM5A Type 3088      Address  500/501  Users
0 DOWN  Group RMTPVM5
03/31/93 09:43:41 DVMAPE081I Link RMTPVM5B Type APPC      Resource EAGLE   Users
0 DOWN  Group RMTPVM5
03/31/93 09:43:43 DVMVTC081I Link PVMG1 Type IUCV Path 0 Users 0 ACTIVE
```

These messages identify all the defined links on the local PVM node RMTPVM1, the group names, the type of each link, the address of each link, the number of users on each link, the status of each link (only the CTCA links, line 200 and 201 forming group RMTPVM2, are connected), the number of users connected to RMTPVM1 (0), the level of PVM Version 2, and the time and date the PVM system was built.

For information about a specific line, enter:

```
q line 200
```

This will produce messages similar to the following:

```
03/31/93 09:44:16 DVMCON050I Q LINE 200
03/31/93 09:44:16 DVMQRY081I Link RMTPVM2A Type CTCA      Address  200      Users
0 CONNECT Group RMTPVM2
```

These messages provide the same information as `q system` but just for the specified line.

If a user dials into RMTPVM1, messages similar to the following will be displayed:

```
03/31/93 09:44:50 DVMUGR150I User GRAF450 connected to node RMTPVM1 Address WG7
51308
03/31/93 09:44:50 DVMAPI180I Logical device L031 created for RMTPVM1 GRAF450, ad
dress WG751308
```

## Operating the Network

These messages indicate that a device whose ID is GRAF450 and whose address is WG751308 (VTAM logical unit name) is connected to RMTPVM1 at logical address L031. To obtain additional information about this display device, enter:

```
q user graf450
```

This produces messages similar to the following:

```
03/31/93 09:45:03 DVMCON050I Q USER GRAF450
03/31/93 09:45:03 DVMQRY088I RMTPVM1 GRAF450 Device WG751308 3278 Model 2 H is
in session at RMTPVM1 L031, Bytes In 1101 Out 134
```

These messages indicate that device GRAF450 is a 3278 Model 2 display station.

Now, suppose two other users enter the DIAL command—one dials directly into RMTPVM1 and the other dials into RMTPVM1 through RMTPVM2. To display the status of each active link and each user session, enter:

```
q sys q
```

```
03/31/93 10:11:45 DVMCON050I Q SYS Q
03/31/93 10:11:45 DVMQRY080I The local node is RMTPVM1, Users 2, PVM 2.1.1000,
built 03/30/93 15:27:08
03/31/93 10:11:45 DVMQRY088I RMTPVM1 GRAF451 Device 017E 3279 Model 3 HPC is in
session at RMTPVM1 L03B, Bytes In 1101 Out 200
03/31/93 10:11:45 DVMQRY088I RMTPVM1 GRAF450 Device WG751308 3278 Model 2 H is
in session at RMTPVM1 L031, Bytes In 1101 Out 134
03/31/93 10:11:45 DVMQRY081I Link RMTPVM2A Type CTCA      Address 200      Users
1 CONNECT Group RMTPVM2
03/31/93 10:11:45 DVMQRY088I RMTPVM2 GRAF460 Device 0D4D 3277 Model 2 is in
session at RMTPVM1 L03C, Bytes In 1095 Out 0
03/31/93 10:11:46 DVMQRY081I Link RMTPVM2B Type CTCA      Address 201      Users
0 CONNECT Group RMTPVM2
```

Now, suppose a user, whose ID is SMITH, enters the PASSTHRU command to get to RMTPVM1. This will produce messages similar to the following:

```
03/31/93 10:13:21 DVMAPP180I Logical device L02B created for RMTPVM1 SMITH, addr
ess WG752308
03/31/93 10:13:21 DVMUIN150I User SMITH connected to node RMTPVM1 L02B Address W
G752308
```

To display additional information about SMITH, enter:

```
q user smith
```

This will produce messages similar to the following:

```
03/31/93 10:13:32 DVMCON050I Q USER SMITH
03/31/93 10:13:32 DVMQRY088I RMTPVM1 SMITH Device WG752308 3278 Model 2 H is in
session at RMTPVM1 L02B, Bytes In 1101 Out 134
```

These messages indicate that user SMITH is connected to RMTPVM1 from a 3278 Model 2 display station.

Now to see the status of the system again, enter:

```
q system
```

This will produce messages similar to the following:

```
03/31/93 10:14:08 DVMCON050I Q SYSTEM
03/31/93 10:14:08 DVMQRY080I The local node is RMTPVM1, Users 5, PVM 2.1.1000,
built 03/30/93 15:27:08
03/31/93 10:14:08 DVMQRY081I Link RMTPVM2A Type CTCA      Address  200
Users  1 CONNECT  Group RMTPVM2
03/31/93 10:14:08 DVMQRY081I Link RMTPVM2B Type CTCA      Address  201
Users  0 CONNECT  Group RMTPVM2
03/31/93 10:14:08 DVMC39081I Link RMTPVM5A Type 3088      Address  500/501
Users  0 DOWN     Group RMTPVM5
03/31/93 10:14:08 DVMQRY081I Link RMTPVM5B Type APPC      Resource EAGLE
Users  0 DOWN     Group RMTPVM5
03/31/93 10:14:39 DVMVTC081I Link PVMG1 Type IUCV Path 0 Users 0 ACTIVE
```

---

## Defining a Link Dynamically

To add a new link to a group without shutting down PVM, enter:

```
define rmtpvm2c new type ctca line 202 group rmtpvm2
```

This will produce messages similar to the following:

```
03/31/93 10:14:47 DVMCON050I DEFINE RMTPVM2C NEW TYPE CTCA LINE 202 GROUP RMTPVM
2
03/31/93 10:14:47 DVMCON001I Your DEFINE command is complete
```

To display the status of each active link and each user session, enter:

```
q sys q
```

If user SMITH2 has entered MPVM and started two sessions, query information for those sessions will also be displayed.

This will produce messages similar to the following:

```
03/31/93 10:16:28 DVMCON050I Q SYS Q
03/31/93 10:16:28 DVMQRY080I The local node is RMTVPV1, Users 5, PVM 2.1.1000,
built 03/30/93 15:27:08
03/31/93 10:16:28 DVMQRY088I RMTVPV1 SMITH Device WG752308 3278 Model 2 H is in
session at RMTVPV1 L02B, Bytes In 1101 Out 134
03/31/93 10:16:28 DVMMPG088I RMTVPV1 SMITH2 *SES0002 Device L014F 3278
Model 2 HC is in session at RMTVPV1 L182, Bytes In 1718 Out 110
03/31/93 10:16:28 DVMMPG088I RMTVPV1 SMITH2 *SES0001 Device L014F 3278
Model 2 HC is in session at RMTVPV1 L181, Bytes In 1718 Out 110
03/31/93 10:16:28 DVMQRY088I RMTVPV1 GRAF451 Device 017E 3279 Model 3 HPC is in
session at RMTVPV1 L03B, Bytes In 1101 Out 200
03/31/93 10:16:28 DVMQRY088I RMTVPV1 GRAF450 Device WG751308 3278 Model 2 H is
in session at RMTVPV1 L031, Bytes In 1101 Out 134
03/31/93 10:16:28 DVMQRY081I Link RMTVPV2A Type CTCA      Address  200      Users
1 CONNECT  Group RMTVPV2
03/31/93 10:16:28 DVMQRY088I RMTVPV2 GRAF460 Device 0D4D 3277 Model 2  is in
session at RMTVPV1 L03C, Bytes In 1095 Out 0
03/31/93 10:16:28 DVMQRY081I Link RMTVPV2B Type CTCA      Address  201      Users
0 CONNECT  Group RMTVPV2
03/31/93 10:16:28 DVMQRY081I Link RMTVPV2C Type CTCA      Address  202      Users
0 CONNECT  Group RMTVPV2
```

A new CTCA link has been added to group RMTVPV2 from RMTVPV1 to RMTVPV2.

---

## Displaying the Path of a Session

Use the SHOW command to display the path of a particular session. For example,

```
show path rmtvpv2
```

shows the path PVM took to get to RMTVPV2. This will produce messages similar to the following:

```
03/31/93 10:17:07 DVMCON050I SHOW PATH RMTVPV2
03/31/93 10:17:08 DVMCOR922I The path to node RMTVPV2 is through node RMTVPV1
03/31/93 10:17:08 DVMNEV127E RMTVPV2 RMTVPV3 LINK IS DOWN
FROM RMTVPV3: Your SHOW command is complete
```



---

## Sending Messages

Three commands can be used to send users messages: LMSG, MSG, and PMSG. Use the LMSG message to send messages to all PVM users associated with the local PVM node or a specified line or group. For example,

```
lmsg rmtpvm1 Are you coming to Monday's meeting?
```

sends the message to all PVM users connected to RMTTPVM1. This will produce messages similar to the following:

```
03/31/93 10:17:34 DVMCON050I LMSG RMTTPVM1 ARE YOU COMING TO MONDAY'S MEETING?
03/31/93 10:17:34 DVMCOU001I Your LMSG command is complete
```

Use the MSG command to send messages to a remote station or to a specified VM user at a specified node. For example,

```
msg rmtpvm2 hayes Are you coming to Tuesday's meeting?
```

sends the message to HAYES at RMTTPVM2. This will produce messages similar to the following:

```
03/31/93 10:19:42 DVMCON050I MSG RMTTPVM2 HAYES ARE YOU COMING TO TUESDAY'S MEETING?
03/31/93 10:19:42 DVMCOU001I Your MSG command is complete
```

Use the PMSG command to send messages to all PVM users connected to the local PVM node. For example,

```
pmsg all Are you coming to Wednesday's meeting?
```

sends the message to all PVM users connected to RMTTPVM1. This will produce messages similar to the following:

```
03/31/93 10:20:39 DVMCON050I PMSG ALL ARE YOU COMING TO WEDNESDAY'S MEETING?
03/31/93 10:20:39 DVMCOU001I Your PMSG command is complete
```

---

## Issuing CMS Subset Mode Commands

Since the configuration file contains the record SUBSET LISTFILE, the CMS files on the A-disk can be displayed by entering:

```
listfile
```

If the configuration file does not contain the SUBSET LISTFILE record, prefix the LISTFILE command with CMS to process the command.

---

### Quiescing, Resuming, and Dropping Links, Devices, and Users

To quiesce line 200, enter:

```
quiesce line 200
```

This will produce messages similar to the following:

```
03/31/93 10:22:18 DVMCON050I QUIESCE LINE 200
03/31/93 10:22:18 DVMCON001I Your QUIESCE command is complete
```

To check the status of the active links on RMTVPV1, enter:

```
q sys q
```

This will produce messages similar to the following:

```
03/31/93 10:22:28 DVMCON050I Q SYS Q
03/31/93 10:22:28 DVMQRY080I The local node is RMTVPV1, Users 5, PVM 2.1.1000,
built 03/30/93 15:27:08
03/31/93 10:22:28 DVMQRY088I RMTVPV1 SMITH Device WG752308 3278 Model 2 H is in
session at RMTVPV1 L02B, Bytes In 1101 Out 134
03/31/93 10:22:28 DVMMPG088I RMTVPV1 SMITH2 *SES0002 Device L014F 3278
Model 2 HC is in session at RMTVPV1 L182, Bytes In 1718 Out 110
03/31/93 10:22:28 DVMMPG088I RMTVPV1 SMITH2 *SES0001 Device L014F 3278
Model 2 HC is in session at RMTVPV1 L181, Bytes In 1718 Out 110
03/31/93 10:22:28 DVMQRY088I RMTVPV1 GRAF451 Device 017E 3279 Model 3 HPC is in
session at RMTVPV1 L03B, Bytes In 1101 Out 200
03/31/93 10:22:28 DVMQRY088I RMTVPV1 GRAF450 Device WG751308 3278 Model 2 H is
in session at RMTVPV1 L031, Bytes In 1101 Out 134
03/31/93 10:22:28 DVMQRY081I Link RMTVPV2A Type CTCA      Address 200      Users
1 QUIESCE  Group RMTVPV2
03/31/93 10:22:28 DVMQRY088I RMTVPV2 GRAF460 Device 0D4D 3277 Model 2 is in ses
sion at RMTVPV1 L03C, Bytes In 1400 Out 12
03/31/93 10:22:28 DVMQRY081I Link RMTVPV2B Type CTCA      Address 201      Users
0 CONNECT  Group RMTVPV2
03/31/93 10:22:28 DVMQRY081I Link RMTVPV2C Type CTCA      Address 202      Users
0 CONNECT  Group RMTVPV2
```

Since there is a user connected on line 200, the line is put in quiesce state rather than being dropped. If the user session ends, then line 200 would be dropped.

However, if the user session does not end line 200 can be *connected* again by entering:

```
resume line 200
```

To confirm that the system is back to the same status (line 200 is CONNECTed) as it was before issuing the QUIESCE command, enter:

```
q sys q
```

This will produce messages similar to the following:

```
03/31/93 10:22:38 DVMCON050I Q SYS Q
03/31/93 10:22:38 DVMQRY080I The local node is RMTVPV1, Users 5,
03/31/93 10:22:38 DVMQRY088I RMTVPV1 SMITH Device WG752308 3278 Model
2 H is in session at RMTVPV1 L02B, Bytes In 1101 Out 134
03/31/93 10:22:38 DVMMPG088I RMTVPV1 SMITH2 *SES0002 Device L014F 3278
Model 2 HC is in session at RMTVPV1 L182, Bytes In 1718 Out 110
03/31/93 10:22:38 DVMMPG088I RMTVPV1 SMITH2 *SES0001 Device L014F 3278
Model 2 HC is in session at RMTVPV1 L181, Bytes In 1718 Out 110
03/31/93 10:22:38 DVMQRY088I RMTVPV1 GRAF451 Device 017E 3279 Model
3 HPC is in session at RMTVPV1 L03B, Bytes In 1101 Out 200
03/31/93 10:22:38 DVMQRY088I RMTVPV1 GRAF450 Device WG751308 3278
Model 2 H is in session at RMTVPV1 L031, Bytes In 1101 Out 134
03/31/93 10:22:38 DVMQRY081I Link RMTVPV2A Type CTCA Address 200
Users 1 CONNECT Group RMTVPV2
03/31/93 10:22:38 DVMQRY088I RMTVPV2 GRAF460 Device 0D4D 3277 Model
2 is in session at RMTVPV1 L03C, Bytes In 1400 Out 12
03/31/93 10:22:39 DVMQRY081I Link RMTVPV2B Type CTCA Address 201
Users 0 CONNECT Group RMTVPV2
03/31/93 10:22:39 DVMQRY081I Link RMTVPV2C Type CTCA Address 202
Users 0 CONNECT Group RMTVPV2
03/31/93 10:22:39 DVMVTC081I Link PVMG1 Type IUCV Path 0 Users 0 ACTIVE
```

To place the display device GRAF450 into quiesce mode, enter:

```
quiesce graf 450
```

This will produce messages similar to the following:

```
03/31/93 10:22:52 DVMCON050I QUIESCE GRAF 450
03/31/93 10:22:52 DVMCON001I Your QUIESCE command is complete
```

The user who dialed into RMTVPV1 can complete the session. However, once the user session is over, the user is dropped and the display support ends.

After issuing QUIESCE, to prevent the user being dropped or the display support ended after the user session completes, enter

```
resume graf 450
```

This will produce messages similar to the following:

```
03/31/93 10:23:16 DVMCON050I RESUME GRAF 450
03/31/93 10:23:16 DVMCON001I Your RESUME command is complete
```

Now, the display device GRAF450 remains available even if the user session completes.

To end the display support for device GRAF450, enter

```
drop graf 450
```

This will produce messages similar to the following:

## Operating the Network

```
03/31/93 10:23:30 DVMCON050I DROP GRAF 450
03/31/93 10:23:30 DVMCON001I Your DROP command is complete
```

Any active user session on display GRAF450 is also ended.

To end SMITH user's session, enter

```
drop user rmtpvml smith
```

This will produce messages similar to the following:

```
03/31/93 10:24:03 DVMCON050I DROP USER RMTVPM1 SMITH
03/31/93 10:24:03 DVMCON001I Your DROP command is complete
03/31/93 10:24:03 DVMAPP181I Logical device L02B ended for RMTVPM1 SMITH, address WG752308,
03/31/93 10:24:03 DVMAPP181I Bytes In 134 Out 1101
03/31/93 10:24:03 DVMUIN151I User SMITH dropped from node RMTVPM1 L02B address WG752308,
03/31/93 10:24:03 DVMUIN151I Bytes In 1101 Out 134
```

SMITH's session at L02B is ended. Then SMITH is dropped from RMTVPM1.

To drop line 200 connecting RMTVPM1 to RMTVPM2, enter:

```
drop line 200
```

This will produce messages similar to the following:

```
03/31/93 10:24:23 DVMCON050I DROP LINE 200
03/31/93 10:24:23 DVMCON001I Your DROP command is complete
03/31/93 10:24:23 DVMCTC501I Link RMTVPM2A Blocks In 11 Out 10, Bytes In 692 Out 2285, Delay 189, Average 68
03/31/93 10:24:23 DVMCTC314I Link RMTVPM2A is terminating via command
03/31/93 10:24:23 DVMNEU105W Link RMTVPM2A Line 0200 Status is DOWN
```

These messages confirm that line 200 to RMTVPM2 is down.

---

## Using the PVM Programmable Operator (PVMPROP)/Command Response Interface (CRI)

Use the PVMPROP programmable operator and the PVM Command Response Interface to write applications that monitor PVM operations. When an problem or error occurs, PVM can send the message to the PVMPROP user ID using the CP MSG command. An application program running on the PVMPROP user ID can act as a PVM operator and respond to the message or send instructions to a human PVM operator.

For more information about PVMPROP and the Command Response Interface, see Chapter 13, "Facilities for Automating PVM and PVMG Operations."

---

## Stopping PVM

There are two ways to stop the PVM system: the SHUTDOWN command or the QUIT command. The QUIT command stops the system immediately without notifying the users. Therefore, before issuing the QUIT command, first try the SHUTDOWN command with the IMMED option. When using this SHUTDOWN command, users receive messages informing them that their session are being terminated.

To stop PVM, enter:

```
shutdown immed
```

This will produce messages similar to the following:

```
03/31/93 10:24:51 DVMCON050I SHUTDOWN IMMED
03/31/93 10:24:51 DVMNET313I PVM system shutdown has started
03/31/93 10:24:51 DVMUGR319I Task GRAF454 is terminating
GRAF 454 DETACHED
03/31/93 10:24:51 DVMCON068I GRAF 454 DETACHED
03/31/93 10:24:53 DVMUGR319I Task GRAF453 is terminating
GRAF 453 DETACHED
03/31/93 10:24:53 DVMCON068I GRAF 453 DETACHED
03/31/93 10:24:53 DVMUGR319I Task GRAF452 is terminating
GRAF 452 DETACHED
03/31/93 10:24:53 DVMCON068I GRAF 452 DETACHED
03/31/93 10:24:53 DVMCTC501I Link RMTTPVM2B Blocks In 8 Out 10, Bytes In 564 Out
654, Delay 191, Average 67
03/31/93 10:24:54 DVMCTC314I Link RMTTPVM2B is terminating via command
03/31/93 10:24:54 DVMNEU105W Link RMTTPVM2B Line 0201 Status is DOWN
03/31/93 10:24:54 DVMCTC501I Link RMTTPVM2C Blocks In 4 Out 4, Bytes In 240 Out 2
40, Delay 192, Average 59
03/31/93 10:24:54 DVMCTC314I Link RMTTPVM2C is terminating via command
03/31/93 10:24:54 DVMNEU105W Link RMTTPVM2C Line 0202 Status is DOWN
```

---

## Starting PVMG

Once PVMG is installed and added to GCS, starting PVMG is simply a matter of IPLing GCS, or issuing RUNPVMG from the PROFILE EXEC.

---

## Querying PVMG

A PVMG operator can determine the status of the system, sessions, nodes, and users. The sections that follow provide examples of how to use the QUERY command.

### Querying the System

To see the information about the system, such as, the number of sessions on the system, line status, or the number of users on the system, enter:

```
pvmg q system
```

This will produce messages similar to the following:

## Operating the Network

```
03/31/93 22:20:53 EFGQSQ080I VM Pass-Through Gateway Services 2.1.1000,  
Users 0, Node PVMG1, built 03/31/93 22:16:25  
03/31/93 22:20:53 EFGQCP081I Link RMTpvm1 Type IUCV Users 0 CONNECT
```

These messages identify all the defined links on the local PVMG node PVMG1, the number of users on each link, the level of PVMG Version 2 and the time and date the PVMG system was built.

## Monitoring User Activity

When a PVM user accesses an SNA application, such as N2N6AVSC, a message similar to the following displays:

```
03/31/93 22:21:43 EFGUSG102I Session start from RMTpvm1 SMITH to PVMG1 GRAF0001,  
Application N2NBASAM, Logical Unit N2ECAP01
```

For information about a specific PVM user, enter:

```
pvmg q user rmtpvm1 smith
```

This will produce messages similar to the following:

```
03/31/93 22:21:50 EFGQCP088I RMTpvm1 SMITH Device 1D9 3279 Model 2 HC is  
in session at PVMG1 GRAF0001  
03/31/93 22:21:50 EFGQCP088I Application N2NBASAM, Logical Unit N2ECAP01,  
Bytes In 1701 Out 110
```

These messages provide the same information as `q` system but just for the specified user.

Suppose an SNA user (from SNA application AL3E0Q) accesses a PVM node, such as RMTpvm5, a message similar to the following displays:

```
03/31/93 22:21:59 EFGUSG102I Session start from PVMG1 AL3E0Q to RMTpvm5  
L120
```

---

## Displaying the Path of a PVMG Session

Use the `SHOW` command to display the path of a particular session. For example,

```
pvmg show path rmtpvm2
```

shows the path from the PVMG node to the RMTpvm2. This will produce messages similar to the following:

```
03/31/93 22:22:04 EFGOCP922I The path to node RMTPVM1 is through node PVMG1
```

## Sending Messages

Three commands that send users messages: LMSG, MSG, and PMSG. Use the LMSG message to send messages to all users whose sessions access PVMG through VTAM, or whose sessions access VTAM through PVMG. For example,

```
pvmg lmsg rmtpvm1 Are you coming to Monday's meeting?
```

sends the message to all users connected to RMTPVM1.

Use the MSG command to send messages to a remote station or to a specified VM user ID at a specified node. For example,

```
pvmg msg rmtpvm2 hayes Are you coming to Tuesday's meeting?
```

sends the message to HAYES at RMTPVM2.

Use the PMSG command to send messages to all PVMG users connected to the local PVMG node. For example,

```
pvmg pmsg all Are you coming to Wednesday's meeting?
```

sends the message to all PVMG users connected to PVMG1.

## Dropping Users and Nodes

To quiesce the link to node RMTPVM1 enter:

```
pvmg quiesce node RMTPVM1
```

Now, to check the status of the active links on PVMG1, enter:

```
pvmg q sys q
```

This will produce messages similar to the following:

```
03/31/93 22:22:29 EFGOQS080I VM Pass-Through Gateway Services 2.1.1000,
Users 2, Node PVMG1, built 03/31/93 22:16:25
03/31/93 22:21:50 EFGOCP088I RMTPVM1 SMITH Device 1D9 3279 Model 2 HC is
in session at PVMG1 GRAF0001
03/31/93 22:21:50 EFGOCP088I Application N2NBASAM, Logical Unit N2ECAP01,
Bytes In 1701 Out 110
03/31/93 22:22:29 EFGO05088I PVMG1 AL3E0R Device AL3E0R 3278 Model 2 H
is in session at RMTPVM5 L120, Bytes In 1423 Out 1964
03/31/93 22:22:29 EFGOCP081 Link RMTPVM1 Type IUCV Users 2 Quiesce
```

Since there are users connected on RMTPVM1, the link is put in quiesce state rather than being dropped. Once the users' sessions end, then link RMTPVM1 would be dropped.

However, if the user session does not end, to make link RMTPVM1 *connected* again, enter:

```
pvmg resume node RMTPVM1
```

## Operating the Network

To verify that the system is back to the same status (RMTPVM1 is CONNECTed) as it was before issuing the QUIESCE command, enter:

```
pvmg q sys q
```

This will produce messages similar to the following:

```
03/31/93 22:23:29 EFGQSQ080I VM Pass-Through Gateway Services 2.1.1000,  
Users 2, Node PVMG1, built 03/31/93 22:16:25  
03/31/93 22:21:50 EFGQCP088I RMTPVM1 SMITH Device 1D9 3279 Model 2 HC is  
in session at PVMG1 GRAF0001  
03/31/93 22:21:50 EFGQCP088I Application N2NBASAM, Logical Unit N2ECAP01,  
Bytes In 1701 Out 110  
03/31/93 22:23:29 EFGQ05088I PVMG1 AL3E0R Device AL3E0R 3278 Model 2 H  
is in session at RMTPVM5 L120, Bytes In 1423 Out 1964  
03/31/93 22:23:29 EFGQCP081 Link RMTPVM1 Type IUCV Users 2 Connect
```

To quiesce the local PVMG server, enter:

```
pvmg quiesce system
```

Any users with sessions through PVMG1 can complete the session. However, once the users' sessions are over, the users are dropped.

After issuing QUIESCE, to prevent the users from being dropped after the user sessions complete, enter

```
pvmg resume system
```

The PVMG1 server remains available even if the user's sessions complete.

To end the session for user SMITH, enter:

```
pvmg drop user pvmg1 graf0001
```

This will produce messages similar to the following:

```
03/31/93 22:26:03 EFGUST103I Session ended from RMTPVM1 SMITH to PVMG1 GRAF0001,  
Application N2NBASAM, Logical Unit N2ECAP01  
03/31/93 22:26:03 EFGQSQ001I Your DROP command is complete
```

SMITH's SNA session ends, and is dropped from link RMTPVM1.

To end the server support for link RMTPVM1, enter

```
pvmg drop RMTPVM1
```

This will produce messages similar to the following:

```
03/31/93 22:25:30 EFGCCT105W Link RMTPVM1 Status is DOWN
```



---

## Stopping PVMG

Issue the SHUTDOWN command. If there are no active sessions, the PVMG server will shutdown immediately. If there are active sessions, the operator can choose to wait for the sessions to end, or can enter the following command:

```
pvmg shutdown immed
```

Users will receive messages informing them that their session will be terminated. The PVMG server then shuts down.

This will produce messages similar to the following:

```
03/31/93 22:30:23 EFG0ED3131 PVMG System Shutdown has Started
03/31/93 22:30:23 EFG0PE626I The PVMG command processor has terminated
03/31/93 22:30:23 EFGCTC626I The PVMG link control task has terminated
03/31/93 22:30:23 EFGUSH626I The PVMG VTAM session control task has terminated
03/31/93 22:30:23 EFGMAI100I PVMG is terminating
```

**Note:** Shutdown without the IMMED option allows active sessions to continue, but will not allow new sessions to start. When the last session ends, shutdown of the PVMG server completes.



## Chapter 9. Using CVIEW

This chapter offers information that can help you with the daily operation and administration of the CVIEW virtual machine.

### Starting the CVIEW Virtual Machine

You can start the CVIEW virtual machine either automatically or manually. Once the CVIEW virtual machine is started, the consult and conference sessions are available for use.

### Automatically Starting CVIEW

To automatically start the CVIEW virtual machine from another virtual machine, ensure that the:

- VM/ESA directory contains an AUTOLOG1 or equivalent user ID.
- AUTOLOG1 virtual machine has Privilege Class A or B and this command (with no variable data) is specified in its PROFILE EXEC:  
CP AUTOLOG CVIEW PASSWORD
- CVIEW virtual machine directory contains an IPL CMS entry
- CVIEW PROFILE EXEC contains an EXEC DXGCVIEW command.

For a complete description of the CP AUTOLOG command, see the *z/VM: CP Commands and Utilities Reference*, SC24-6268.

Once the CVIEW virtual machine is automatically logged on, it runs in a disconnected mode.

### Manually Starting CVIEW

You can start the CVIEW virtual machine manually using these steps:

1. Clear the VM/ESA system logo from the screen.
2. Enter:  
L CVIEW
3. Enter the password when prompted.
4. If an IPL CMS was not specified in the directory for the CVIEW virtual machine, enter:  
IPL CMS
5. Enter a null line to run the PROFILE EXEC.
6. If an EXEC DXGCVIEW command was not specified in the PROFILE EXEC for the CVIEW virtual machine, when the IPL completes enter:  
DXGCVIEW
7. After CVIEW is running, you can enter:  
CP DISC  
to disconnect the CVIEW virtual machine.

Once CVIEW is running, users can enter the DIAL CVIEW command from their own display stations to start consult and conference sessions.

---

## CVIEW Commands

CVIEW has both *general* and *privileged* commands. The general commands can be used by any CVIEW user to obtain operating status and information about the CVIEW virtual machine. Users defined as privileged (during customization) and users logged on to the CVIEW virtual machine console can use the privileged commands. Privileged commands administer and control the operation of the CVIEW virtual machine.

The commands that follow can be entered from the CVIEW virtual machine console. However, if you want to enter commands from a virtual machine console other than the CVIEW virtual machine console use the CP SMSG command. For example, a general user who wants to enter the CVIEW HELP command from his virtual machine console enters:

```
SMSG userid HELP
```

where *userid* is the user ID of the CVIEW virtual machine that will receive the command. (HELP is a CVIEW general command).

For a complete description of the CP SMSG command, see the *z/VM: CP Commands and Utilities Reference*, SC24-6268.

---

## CMS



### Purpose

Lets you enter CMS commands within the CVIEW virtual machine.

### Authority

Privileged

### Operands

*command*

Is the CMS command to be processed.

**EXEC** *command*

Specifies that the command to be processed is an exec. *command* is the file name of the EXEC.

### Usage Notes

1. The CVIEW virtual machine does not perform other functions while a CMS command is being processed.
2. CMS commands that process slowly can effect users engaged in CVIEW sessions.

---

## CP

```
▶▶—CP—command—◀◀
```

### Purpose

Lets you enter CP commands within the CVIEW virtual machine.

### Authority

Privileged

### Operands

*command*

Is the CP command to be processed.

### Usage Notes

1. The CP LOGOFF command is not detected by CVIEW; if entered, it will log off the CVIEW virtual machine.

---

## DROP

```
▶▶—DROP—┬—GRAF nnn—▶◀
          └—SESSION xxxxxxxx—
          └—LDEV Lnnn—▶◀
```

### Purpose

Stops an active GRAF device by number or a CVIEW session by number, name, or logical device address. The CVIEW session ends if the:

- GRAF address is assigned to the moderator's display station.
- Session has only two users.

### Authority

Privileged

### Operands

**GRAF** *nnn*

Indicates the 3270 display station that has dialed the CVIEW virtual machine. The *nnn* is the virtual address assigned by the CVIEW virtual machine to a 3270 display station. Valid GRAF addresses are 501 through 530. Use QUERY GRAF to find the active GRAF address for a display station.

**SESSION** *xxxxxxxx*

Indicates an active consult or conference session. The *xxxxxxxx* is an unlisted consultant's identification (up to eight alphanumeric characters), a conference

# PASSWORD

name (up to eight alphanumeric characters) or a listed consultant's ID (a number from 1 to 12).

## LDEV *Lnnn*

Indicates a consult or conference session by the logical device address assigned by CP for communications between CP and a CVIEW session. *Lnnn* is the LDEV address assigned by CP to a CVIEW session. Use the QUERY ACTIVE command to obtain the LDEV address.

---

# HELP

▶▶—HELP—◀◀

## Purpose

Displays on-line information about CVIEW commands. Privileged users receive help information for all privileged and general commands. Other users receive help information for the general commands.

## Authority

General

---

# PASSWORD

▶▶—PASSWORD—

ADD *number password*

DELETE *number*

CHANGE *number newpassword*

DISPLAY *number*

◀◀

## Purpose

Maintains passwords for consultants that are listed in the CONSULT IDNAME file and appear on the GIVE and RECEIVE panels. You can also maintain consultant passwords by editing the CONSULT IDNAME file.

## Authority

Privileged

## Operands

### ADD

Assigns a password to a consultant with no current password.

#### *number*

Is a value from 2 to 12 that corresponds to the consultant in the CONSULT IDNAME file.

#### *password*

Is the password to be assigned to the specified consultant. The password can be up to eight characters long.

## DELETE

Removes a consultant's password from the CONSULT IDNAME file.

## CHANGE

Assigns a different password to a consultant.

*number*

Is a value form 2 to 12 that corresponds to the consultant in the CONSULT IDNAME file.

*newpassword*

Is the new password to be assigned to the specified consultant. The new password can be up to eight characters long.

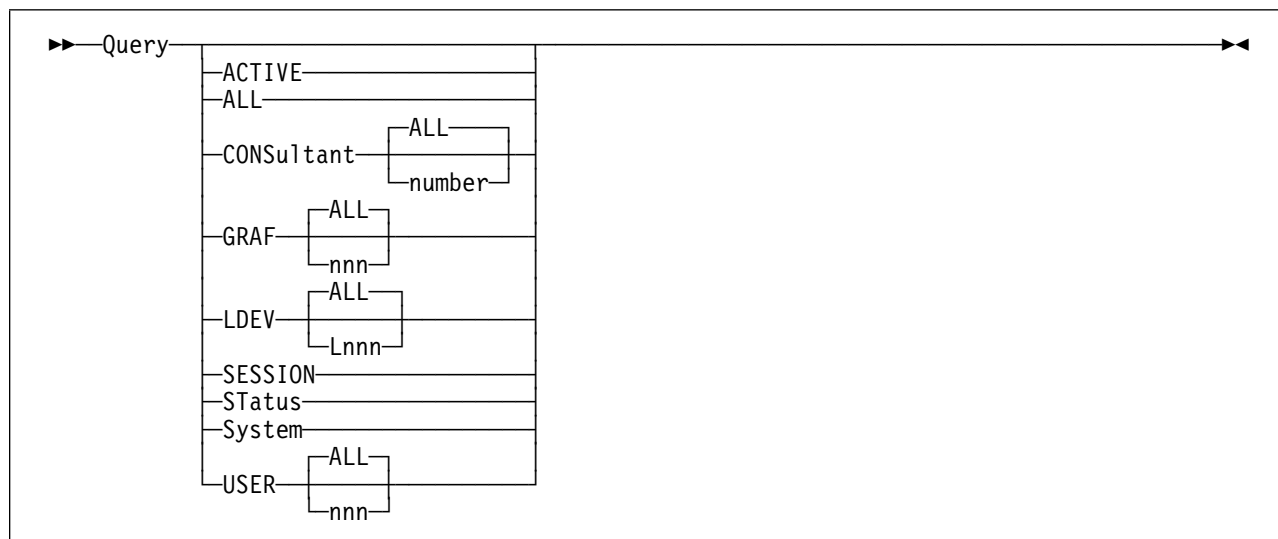
## DISPLAY

Lets a privileged user look at a consultant's password.

*number*

Is a value from 2 to 12 that corresponds to the consultant in the CONSULT IDNAME file.

# QUERY



## Purpose

Gets information and status about many components of the CVIEW virtual machine.

## Authority

General

## Operands

### Query

Used alone it returns the version, release, and modification level of the CVIEW virtual machine, if the CVIEW virtual machine is running.

### ACTIVE

Returns the session name or the consultant's ID number, the session LDEV address, the display station type and model as assigned by CVIEW, and the display input PF key setting for all currently active consult and conference sessions. The QUERY ACTIVE command is a synonym for and interchangeable with the QUERY SESSION command.

### ALL

Returns the same information as the QUERY ACTIVE command plus all the active GRAF addresses and the display station type and model of each GRAF. Also, ALL returns all customization values in effect.

The System option is a synonym for and is interchangeable with the ALL option.

### CONSultant

Returns information about consultants.

#### *number*

Returns a busy or not busy status for a specific consultant listed in the CONSULT IDNAME file. It also states how many recipients are waiting for the consultant's help. Number 2 through 12 is the ID of a consultant in the ID NAME file.

### ALL

Returns the same information that CONSultant number displays, for every consultant listed in the ID NAME file by consultant number. ALL is the default parameter.

### GRAF

Returns the display station type and model and the session type if the GRAF is active. If the specified GRAF address is not active, the command returns a message indicating this.

#### *nnn*

Is the virtual address, assigned by the CVIEW virtual machine, of a 3270 display station. Valid GRAF addresses are 501 through 530.

### ALL

Returns the display station type and model and the session type for every GRAF that is active. ALL is the default parameter.

### LDEV

Returns the session name and type, the display station type and model as assigned by CVIEW, and the setting of the display-input PF key.

#### *Lnnn*

Is the LDEV address assigned by CP to a CVIEW session. Use the QUERY ACTIVE command to obtain the LDEV address.

### ALL

Returns the session name and type for all active LDEVs, along with their display station types and models as assigned by CVIEW, and the settings of the display-input PF key. ALL is the default parameter.

### SESSION

Returns the same information as the QUERY ACTIVE command.



**Status**

Returns a message indicating whether the CVIEW virtual machine is running.

**System**

Returns the same information as the QUERY ALL command. The S option is a synonym for and interchangeable with the ALL option.

**USER**

Returns the same information as QUERY GRAF.

*nnn*

Is the virtual address, assigned by the CVIEW virtual machine, of a 3270 display station. Valid GRAF addresses are 501 through 530.

**ALL**

Returns the same information as the QUERY GRAF ALL command. ALL is the default parameter.

---

## SHUTDOWN

►►—SHUTDOWN—◄◄

**Purpose**

Stops the CVIEW virtual machine if no GRAFs or sessions are active. The CVIEW virtual machine will not stop if there are any active GRAFs or sessions.

**Authority**

Privileged

**Usage Notes**

1. Use the SHUTDOWN command when you want to avoid affecting any active users and sessions. (See the STOP command.)

---

## STOP

▶—STOP—◀

### Purpose

Stops the CVIEW virtual machine. All active sessions and users are immediately ended.

### Authority

Privileged

### Usage Notes

1. When you enter a STOP command, no information is logged in the CVIEW LOG file. (See the SHUTDOWN command).

---

## Establishing a Session

When the CVIEW operator has started the CVIEW virtual machine, consult and conference sessions can be established. As a session participant, you **do not** log on the CVIEW user-ID. Instead, you enter the 'DIAL CVIEW' command to set up and establish a session.

CVIEW provides menu and data-entry panels to guide you through the setup process. There are help panels for each setup panel; to see one press the PF key assigned to the HELP function. You will see several menu and data-entry panels in the course of setting up a CVIEW session.

When the consult or conference session starts, the logo for the VM/ESA system CVIEW is running under is displayed to all session participants. At this point, one participant with an active keyboard can clear the system logo and log on any user-ID defined in the system's directory or can "pass through" to another system node if desired.

To establish a consult or conference session, do the following:

1. Clear the VM/ESA system logo.
2. Enter: DIAL CVIEW

The CVIEW HOME panel, shown in Figure 22 on page 83 appears.

HOME	COOPERATIVE VIEWING FACILITY (CVIEW)					
-----						
Enter an item number.						
Consult Session						
1 Receive assistance						
2 Give assistance						
Conference Session						
3 Attend a conference						
4 Moderate a conference						
Press ENTER after you enter an item number.						
-----						
PF: 1= Help	2=	3= Quit	4=	5=	6=	
PF: 7=	8=	9=	10=	11=	12=	
===>						

Figure 22. The HOME Panel Indicates the Start of Setup Mode

3. Select your role for the session from the HOME panel. Each role has specific characteristics:

#### Consult Session

- **Recipient's role — receive assistance**

A recipient always has an active keyboard and can enter data and commands during the consult session. If the consultant assigns a display-input PF key, the recipient can use it. If the recipient presses the PA1 key, the host or application program processes it.

- **Consultant's role — give assistance**

A consultant always has an active keyboard and can enter data and commands during the consult session. A consultant can assign and use a display-input PF key. When the consultant presses the PA1 key, the CONSULT SESSION CONTROL panel appears and the consult session is suspended.

#### Conference Session

- **Attendee's role — attend a conference**

An attendee always has an *inactive keyboard*, so he cannot enter data and commands during the conference session. He can, however, see the data and commands entered by the moderator and presenters, and can talk to other participants by telephone. If the moderator assigns him presenter status, he then has an active keyboard and can enter data and commands.

- **Moderator's role — moderate a conference**

A moderator always has an active keyboard and can enter data and commands during the conference session. A moderator can assign and use a display-input PF key. When the moderator presses the PA1 key, the CONFERENCE CONTROL panel appears and the conference session is

suspended. The moderator controls the session and can assign presenter status to any of the attendees.

See the discussions of consult and conference sessions, later in this chapter, for detailed descriptions of the activities associated with each role.

- 4. After you select a role from the HOME panel, additional setup panels are displayed to identify the consult or conference session. See Appendix I, “CVIEW Product Panels” on page 335, for all the setup-mode panels.
- 5. When setup is complete, the session starts: the logo of the VM/ESA system CVIEW is running under (similar to the logo in Figure 23 is displayed to all participants).

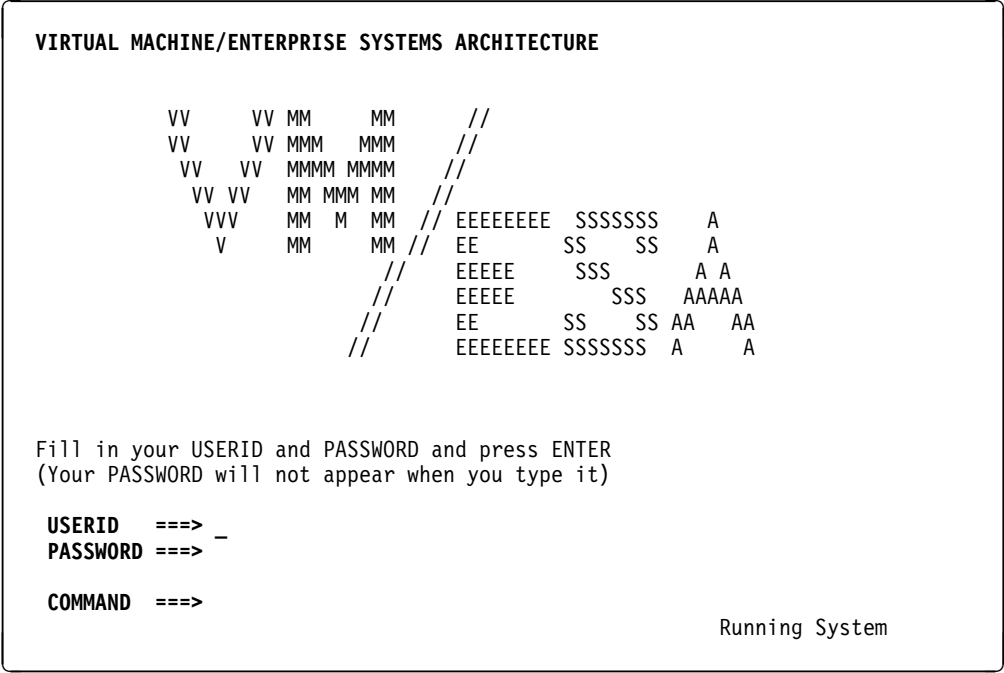


Figure 23. The VM/ESA System Logo. Indicates that the CVIEW session has started.

- 6. One session participant with an active keyboard clears the VM/ESA system logo logs on to the system, and proceeds as for a usual interactive session.  
For a consult session, either the recipient or the consultant can log on.  
For a conference session, the moderator or a presenter can log on.
- 7. If the notify option is in effect for your system, the NOTIFY panel appears for the first time shortly after one user logs on to the system. It will appear again at preset intervals as the interactive session progresses.

Figure 24 shows the NOTIFY panel, which indicates the beginning of notify mode.

```
NOTIFY                                CVIEW - NOTIFY
-----
This Notify panel is being displayed by the Cooperative Viewing Facility
(CVIEW) to make you aware you are running under the control of
CVIEW.

You are currently in a CVIEW session, someone else is watching the same
screen images you are displaying.

If you press the QUIT or HOME PF key while this panel is displayed, the
session will be terminated.

If you were not aware you were using this facility, contact your Computer
Center Management.

Every participant who receives a Notify panel must press ENTER for the
session to continue

Press ENTER to continue.
-----
PF:  1= Help      2=          3= Quit    4= Home    5=          6=
PF:  7=          8=          9=          10=         11=         12=

===>
```

Figure 24. The NOTIFY Panel

If a NOTIFY panel appears on your screen, you must press the ENTER key for the session to continue. If you were not aware that your interactive session is executing under the control of CVIEW, and you do not wish to continue under these circumstances, press the PF key assigned to either the HOME or the QUIT function and the entire CVIEW session will end.

- 8. When the interactive session is finished, one user with an active keyboard logs off the system and the HOME panel is displayed to all session participants. Another session can be established or you can press the PF key assigned to the QUIT function to be dropped out of setup mode.

## Managing a CVIEW Session

During setup and session modes, consult and conference sessions are controlled by the consultant and moderator. The consultant and moderator provide information to session participants during setup to correctly identify the session. In addition, the consultant and moderator can suspend a session and enter control mode to change certain session characteristics. The next few pages describe the things each user can do to manage a CVIEW session.

### Consult Sessions

During setup mode, the consultant guides the recipient through the setup panels to provide the proper session identification. Consult-session identification can be either a listed consultant ID number or a name (if the consultant is not listed on the GIVE and RECEIVE panels). A listed consultant may also be required to enter a password. If the consultant is not listed, he must tell the recipient the exact session name to enter. The consult session starts when both users complete all the setup panels displayed to them.

A **recipient** can do the following:

- Return to the HOME panel during setup mode by pressing the HOME PF key
- Leave CVIEW during setup mode by pressing the QUITPF key
- Enter data or commands during the consult session
- Use the display-input PF key assigned by the consultant
- Send the PA1 key function to the host or an application program for processing, by pressing the PA1 key
- Continue a consult session by pressing ENTER in response to the NOTIFY panel
- End a consult session by pressing the QUIT or HOME PF key in response to the NOTIFY panel
- End a consult session by logging off.

A **consultant** can do the following:

- Return to the HOME panel during setup mode by pressing the HOME PF key
- Leave CVIEW during setup mode by pressing the QUIT PF key panel
- Enter data or commands during the consult session
- Assign a display-input PF key during setup mode
- Use the display-input PF key during the consult session
- Press the PA1 key to suspend the consult session and access the CONSULT SESSION CONTROL panel to:
  - Change or cancel the display-input PF key assignment
  - Send the PA1 key function to the host or an application program for processing, by pressing the PA1 key
  - End the consult session by pressing the HOME or QUIT PF key.
- Continue a consult session by pressing ENTER in response to the NOTIFY panel
- End the consult session by pressing the QUIT or HOME PF key in response to the NOTIFY panel
- End the consult session by logging off.

## Conference Sessions

During setup mode for a conference session, the moderator guides attendees through the setup panels to provide the proper session identification. The moderator selects a session name and a password if one is desired. The moderator tells the attendees the exact session identification to enter. The moderator can assign presenter status to any of the attendees and drop attendees before or after starting the conference session. The moderator starts the conference by pressing the SESSION PF key when the CONFERENCE PARTICIPANTS panel is displayed.

A conference session **attendee**, whose keyboard is inactive, can do the following:

- Return to the HOME panel during setup mode by pressing the HOME PF key
- Leave CVIEW during setup mode by pressing the QUIT PF key

- Leave the conference session by pressing PA1. This does not disrupt the conference in progress if at least two participants remain.
- Continue a conference session by pressing ENTER in response to the NOTIFY panel
- End a conference session by pressing the QUIT or HOME PF key in response to the NOTIFY panel
- Join a conference already in progress except for a user with an incompatible display station type and model cannot join a conference session already in progress. See “Joining a Conference Session in Progress” on page 88

A conference session **presenter** can do the following:

- Return to the HOME panel during setup mode by pressing the HOME PF key
- Leave CVIEW during setup mode by pressing the QUIT PF key
- Enter data of commands during the conference session
- Use the display-input PF key assigned by the moderator
- Send the PA1 key function to the host or an application program for processing, by pressing the PA1 key
- Continue a conference session by pressing ENTER in response to the NOTIFY panel
- End a conference session by pressing the QUIT or HOME PF key in response to the NOTIFY panel
- End a conference session by logging off.

A conference session **moderator** can do the following:

- Return to the HOME panel during setup mode by pressing the HOME PF key
- Leave CVIEW during setup mode by pressing the QUIT PF key
- Enter data or commands during the conference session
- Assign a display-input PF key during setup mode
- Use the CONFERENCE PARTICIPANTS panel to:
  - Start a conference session by pressing the SESSION PF key during setup mode
  - Change an attendee's status to presenter
  - Change a presenter's status to attendee
  - Drop an attendee who is not on the NOTIFY panel
  - Return to a conference session by pressing the SESSION PF key
  - End the session by pressing the HOME or QUIT PF key
  - End the session by dropping the last attendee.
- Use the display-input PF key during the conference session
- Press the PA1 key to suspend the conference session and access the CONFERENCE CONTROL panel. From the CONFERENCE CONTROL panel the moderator can:
  - Change or cancel the display-input PF key assignment

- Send the PA1 key function to the host or an application program for processing
- End the session by pressing the HOME or QUIT PF key
- Display the CONFERENCE PARTICIPANTS panel.
- Continue a conference session by pressing ENTER in response to the NOTIFY panel
- End a conference session by pressing the QUIT or HOME PF keys in response to the NOTIFY panel
- End a conference session by logging off.

## Device Type and Model Assignment

When a consult or conference session is being established, CVIEW determines the display station type and model to be used in the session. Its purpose is to provide all session participants with identical screen images while in session mode. Because a variety of display stations with different screen sizes are supported by CVIEW, it must assign a display station type and model to the session that accommodates the different display stations participating in the session. The display station type and model assigned to the session is determined as session mode starts.

The following table shows the display station type and model selected for a session when two types and models are compared.

	<b>3277-2</b>	<b>3278-2</b>	<b>3278-3</b>	<b>3278-4</b>	<b>3278-5</b>
3277-2	3277-2	3277-2	3277-2	3277-2	3277-2
3278-2	3277-2	3278-2	3278-2	3278-2	3277-2
3278-3	3277-2	3278-2	3278-3	3278-3	3277-2
3278-4	3277-2	3278-2	3278-3	3278-4	3277-2
3278-5	3277-2	3277-2	3277-2	3277-2	3278-5

For example, if a consultant has a 3278-4 and the recipient has a 3278-2, CVIEW assigns a display station type and model of 3278-2 to the consult session.

## Joining a Conference Session in Progress

The following table illustrates whether a display station wanting to join a conference session will be allowed to join if the moderator has started the session.

The horizontal header represents the display station type and model assigned to the conference session.

The leftmost column represents the type and model of a display station that wants to join a conference after the session has begun.



	3277-2	3278-2	3278-3	3278-4	3278-5
3277-2	Yes	No	No	No	No
3278-2	Yes	Yes	No	No	No
3278-3	Yes	Yes	Yes	No	No
3278-4	Yes	Yes	Yes	Yes	No
3278-5	Yes	No	No	No	Yes

For example, a conference session has been started by the moderator and CVIEW has assigned the display station type and model as a 3278-3. A user wants to join the conference in progress from a 3278-5. The user will not be allowed to join the conference in progress from that display station. The user could join the conference in progress from a 3278-3 or a 3278-4.

The 3278-5 has a wider screen than the other display station types and models supported by CVIEW Version 2. If the entire 3278-5 screen width is needed in a session, it can be used only if all display stations participating in the session are 3278-5.

---

## CVIEW PROFILE EXEC

CVIEW includes a sample profile exec, named DXGPROF EXEC. The exec is loaded to the CVIEW 191 A-disk during installation. The control statements contained in the sample profile exec are as follows:

```
&TRACE OFF
* 5684-100 COPYRIGHT IBM CORP 1993
* LICENSED MATERIAL - PROGRAM PROPERTY OF IBM
EXEC DXGCVIEW
&EXIT &RETCODE
```

To use the sample profile exec, copy the file with the replace option and specify the new filename and filetype as PROFILE EXEC. **Do not** erase or rename the DXGPROF EXEC file on the CVIEW 191 A-disk.

---

## CVIEW Log File

CVIEW includes a logging file that records information about consult and conference session activities. The CVIEW virtual machine writes records to the CVIEW LOG file sequentially as activity occurs, until the CVIEW A-disk is full. When the A disk is full, no more information is logged in the CVIEW LOG file. The person responsible for administering the CVIEW virtual machine should periodically review and purge the file.

You can use the information in the CVIEW LOG file to generate utilization reports for your establishment. The file contains variable-length records. The fields within each record contain the following:

Positions	Field Content
1 - 5	Type of session activity
7 - 16	Type of session
18 - 25	Session identification

Positions	Field Content
27 - 36	User's role in session
38 - 45	Date activity occurred
47 - 54	Time activity occurred
56 - 61	Display-station model number as assigned by CVIEW for GRAF or session
63 - end	One of the following: <ul style="list-style-type: none"> <li>• CVIEW virtual GRAF address and the real address</li> <li>• CVIEW virtual GRAF address and the logical device address</li> <li>• CVIEW virtual GRAF addresses for the GRAF devices in a conference session.</li> </ul>

The following are sample records from a CVIEW LOG file:

Record <....+....1....+....2....+....3....+....4....+....5....  
Number +....6....+....7....+....8....+....9....+....10...+....>

```

00001  JOIN  CONFERENCE SALES    Moderator  02/13/93 15:48:32
        3277-2 501 = LDEV E06

00002  JOIN  CONFERENCE SALES    Attendee   02/13/93 15:50:34
        3278-2 502 = GRAF 391

00003  JOIN  CONFERENCE SALES    Attendee   02/13/93 15:50:45
        3278-3 503 = GRAF 392

00004  START CONFERENCE SALES    Session    02/13/93 16:35:18
        3277-2 GRAFS=  501  502  503

00005  STOP  CONFERENCE SALES    Session    02/13/93 16:35:39
        3277-2 GRAFS=  501  502  503

00006  JOIN  CONSULT    2        Recipient  03/07/93 10:56:17
        3278-2 501 = GRAF 383

00007  JOIN  CONSULT    2        Consultant 03/07/93 10:56:26
        3278-3 502 = GRAF 344

00008  JOIN  CONSULT    2        Session    03/07/93 10:56:27
        3278-2 GRAFS=  502  501

00009  STOP  CONSULT    2        Session    03/07/93 10:58:31
        3278-2 GRAFS=  502  501

00010  JOIN  CONFERENCE HISTORY  Attendee   03/07/93 10:59:00
        3278-4 501 = GRAF 383

00011  JOIN  CONFERENCE HISTORY  Moderator  03/07/93 10:59:06
        3278-5 502 = GRAF 344

00012  START CONFERENCE HISTORY  Session    03/07/93 10:59:26
        3277-2 GRAFS=  502  501

00013  STOP  CONFERENCE HISTORY  Session    03/07/93 11:04:55
        3277-2 GRAFS=  502  501

```

---

## Part 2. Determining Problems

<b>Chapter 10. Collecting Problem Information</b> . . . . .	93
Trace Facility for the PVM Component . . . . .	94
The Dump Facility . . . . .	97
Analyzing Dump Information . . . . .	98
Diagnosing PVMG Problems . . . . .	101



## Chapter 10. Collecting Problem Information

### DMTI

Diagnosis, Modification, and Tuning Information is provided to help diagnose PVM and its components.

**WARNING:** Do not use this Diagnosis, Modification, and Tuning Information as a programming interface.

This chapter points out what problem determination aids are available for use with PVM, PVMG, and MPVM, and how these aids are invoked. The problem determination aids are intended for use by:

- IBM Support Personnel
- System programmers with a knowledge of PVM structure
- System administrators and operators with the assistance of IBM Support Personnel.

Dumps and traces are PVM problem determination aids. Dumps provide information about PVM when an error causes PVM or a component of PVM to abnormally terminate. Dumps are automatically produced when PVM or the component fails.

Traces are used when there are symptoms of a PVM problem but PVM continually recovers. In these cases, dumps are not produced, and traces provide information used to solve a problem. A system administrator or operator has to issue commands to PVM to produce traces.

Some types of problems that may occur with PVM include:

- Users can't access a target system.

Check the PVM configuration file to make sure that the target system is set up and defined correctly. Hardware problems may be preventing the connection to the target system.

- PVM has very slow response time.

Check to make sure that all of the PVM performance options are being used. "Enhancing PVM Performance" on page 107 describes these performance options.

- PVM has stopped running or abnormally terminated.

Use the PVM trace and dump facilities to try to find the problem.

If a solution to a problem cannot be found, contact IBM Support Personnel. Be prepared to answer questions about:

- The release and service levels of the VM system that PVM is running in
- Other products that are running in the VM system
- The release and service level of PVM
- The PVM configuration.

Create a table showing the PVM network configuration. Having this information available for the IBM Support Personnel can sometimes help get PVM problems fixed more quickly.

**Note:** Help for all PVM messages with identifiers is available through the z/VM HELP Facility. Only messages without identifiers, and Compiler messages are documented in this book.

---

## Trace Facility for the PVM Component

PVM provides tracing for:

- Line driver I/O
- Line driver data transfer
- Session or MPVM user data transfers
- System functions.

Some forms of the TRACE command can be issued before a line starts. The type of tracing specified by the TRACE options is invoked when the line is started at a later time. These TRACE commands can be placed in the PVM profile or can be entered from the PVM user ID:

- TRACE LINE IO
- TRACE LINE DATA
- TRACE LINE WRAP
- TRACE NODE IO
- TRACE NODE DATA
- TRACE NODE WRAP.

(See page 228 for the exact syntax for the TRACE command.)

Later, if a line develops a problem, having TRACE invoked in this way can make it easier to determine the source of that problem.

See Appendix B, "PVM Commands" for a description of the TRACE command and its options.

## System Services Tracing

System services tracing on an IUCV diagnose function is performed once control is returned to the support task upon an IUCV diagnose function, and a condition code has been saved. The event will be recorded in the PVM system trace. This information will include the IUCV function type, user ID and path ID of the MPVM user, and the length of data sent or received from the MPVM user. If the IUCV diagnose function ends with an error, a PVM message is produced to record the error.

## TRACE USER Command

Data for one or all user sessions can be traced using PVM's TRACE USER command. See "TRACE" on page 228 for information about the PVM TRACE command.

If the *vmid* is a VM user ID of a MPVM user, the data for all of that user's sessions will be traced if the *nodeid* is the origin node ID of the MPVM user. Data for one session can also be traced by specifying the destination user ID and node ID in the *nodeid* and *vmid* parameters of the TRACE parameters.

## Tracing Line Driver I/O Activity

When the IO and ON options of the TRACE command are used, line driver I/O activity tracing begins automatically when the associated line driver is started, and continues as long as the line driver is active. When the WRAP option is used, this information is recorded in an internal wraparound trace table. If the line driver terminates abnormally because of an unexpected line problem, the line driver uses the CP DUMP command to automatically print out the trace table contents. This includes a record of the events occurring on that line just prior to the failure.

The format of each entry of the trace table when the WRAP option is used is as follows:

<i>Table 12. Format of Trace table entries</i>		
Offset	Length	Contents
0	8	CSW
8	1	Sense byte (if CSW indicates UC)
9	3	Channel Program Address
12	4	Identifier (NCTC, SIM, C381, or C382)
16	8	First CCW
24	16	Data associated with first CCW
40	8	Second CCW
48	16	Data associated with second CCW
64	8	Third CCW
72	16	Data associated with third CCW
88	8	Fourth CCW
96	16	Data associated with fourth CCW
<b>Note:</b> Each entry contains at least 1 and up to 4 CCWs.		

A different trace table format is used by the 3088 line driver when attached only to the FICON channel. Two sense bytes are returned on a unit check, and this trace table accommodates those two sense bytes.

The format of each entry of the FICON channel trace table when the WRAP option is used is as follows:

<i>Table 13 (Page 1 of 2). Format of FICON CTC Trace table entries</i>		
Offset	Length	Contents
0	8	CSW
8	4	Channel Program Address
12	4	Identifier (S38R, S38W)
16	2	Sense bytes
18	8	First CCW
26	18	Data associated with first CCW
42	10	Second CCW
50	18	Data associated with second CCW
66	10	Third CCW
74	18	Data associated with third CCW
90	10	Fourth CCW

Table 13 (Page 2 of 2). Format of FICON CTC Trace table entries		
Offset	Length	Contents
98	18	Data associated with fourth CCW
<b>Note:</b> Each entry contains at least 1 and up to 4 CCWs.		

Use the LINE or NODE options of the TRACE command whenever there appears to be a communication problem on a PVM link. All I/O activity on the line is recorded and indicates problems in teleprocessing hardware and protocol. Interpreting TRACE results requires a knowledge of IBM channel commands and basic teleprocessing protocol.

When the IO or DATA options of the TRACE command, output is written to a designated virtual printer. When the WRAP option is used, output is placed in a wraparound trace table. For more information on using the TRACE command, see page 228.

### Line Driver Data Transfer Tracing

Chronology and contents of data buffers passed to and from a line driver can be traced by using the TRACE command with the LINE and DATA, or NODE and DATA options. This is useful when a problem is suspected with a data transfer between PVM components or between systems in a network. The line driver dumps the contents of each data buffer it sends or receives across the telecommunication line. The duration of the trace is controlled by the ON and OFF parameters of the TRACE command. The TRACE command is described in Appendix B, "PVM Commands."

### PCCF Tracing

PCCF activity can be recorded by using the TRACE PCCF command or by using the START PCCF command with the TRACE option. PCCF tracing can be started only for active PCCF tasks. To stop a PCCF trace, use the OFF parameter of the TRACE command. The TRACE command is described in Appendix B, "PVM Commands."

### Tracing System Activity

Use the TRACE command with the SYSTEM option to assist in problem source determination when internal PVM trouble is suspected. This combination issues a chronological trace of the activities of key PVM system components. The trace facility records requests for service, passed data (if any), execution of services, and status information. The sequence of the trace records is also a map that shows the transfer of control between various components.

The output resulting from the TRACE command can be directed to the PVM user ID, or to a virtual printer. The output, however, can be lengthy so it often cannot be displayed on the PVM user ID's display station. The TRACE command is described in Appendix B, "PVM Commands."



## The Dump Facility

The PVM component does basic error checking during its processing. If a problem, such as a program check, causes PVM to abnormally terminate, the system produces a full dump of the type specified in the DUMP record of the configuration file. Abnormal termination of some PVM component parts cause partial dumps. These dumps are automatically produced by PVM. Comment fields of the dump output include date and time stamps in the following format:

mm/dd/yy hh:mm:ss

In addition to producing a dump, PVM writes informational messages to the PVM user ID and to the optional audit file (if the AUDIT record is included in the configuration file).

## Full Dumps

### Multitasking Supervisor

The PVM multitasking supervisor does basic checking during its processing. If it detects a situation that makes it impossible to continue, it issues a multitasker abort (DVMABRT macro with a special error code parameter). Error message ABR000E (with corresponding error code) is issued, a dump (of the type specified in the PVM CONFIG file) is produced for nonzero return codes, and control returns to CMS.

Most invalid SVC parameter lists give an error return code to the calling task.

Any program check within PVM results in a full virtual storage dump. PVM tries to continue processing without the aborted task. If that is not possible, PVM aborts.

### Session Manager

The session manager issues error messages for invalid commands, and terminates the processing of an event if it cannot perform that task. It then waits for the next event.

### Buffer Manager

The buffer manager aborts if it cannot get initial free storage to build the buffer pool, if it finds an internal error within the buffer pool, if a task releases a buffer belonging to another task, or if the (passed) buffer address is invalid. The buffer manager abort procedure is as follows:

- Issue the error message describing the problem
- Store the current registers in the TREGs field of the current task control blocks (TCB)
- Request the appropriate type dump
- Suspend the aborted task is suspended
- Try to continue processing without the suspended task.

### Partial Dumps

#### Line Drivers

PVM line drivers check telecommunication line protocol and attempt to recover from errors. If error recovery fails, the line driver dumps its TCB along with its internal (wraparound) trace table, and then terminates. The session manager restarts the line driver if a connection was previously established.

The networking line drivers maintain a block sequence counter and restart the connection if a block is found to be missing.

The line drivers keep track of all errors and writes the error statistics to the PVM user ID and to the optional audit file at termination time.

#### User and PCCF Support Tasks

These PVM tasks check for error conditions and terminate the session, or sessions in the case of an MPVM user, if any error cannot be handled. Such an error causes a dump of related control blocks and associated buffers as well as termination of the task.

---

### Analyzing Dump Information

Use the z/VM Dump Tool to analyze system failures and then manage problem information and status. Use these facilities to interactively retrieve and examine information from PVM storage dumps to help track and report PVM problems.

### Other PVM Problem Determination Actions

PVM's MPVM user support task will do basic error checking during its normal processing. If a problem occurs, the MPVM support task produces a dump to the trace printer. Sessions and the user task(s) will be terminated when unrecoverable errors occur at the session or task level. Information useful for IBM service personnel is included in the dump that results.

### Tracing MPVM

If there is a problem with MPVM, MPVM's EFGVIEW module can be traced, either at a certain point in its processing, or from the time it is first invoked.

#### Tracing Pieces of EFGVIEW Execution

If users are encountering problems with the MPVM user module (EFGVIEW), setting on MPVM trace variables can help understand what processing is taking place. These variables are of the form TRACE $n$ , where  $n$  is a number from 1 to 7 (8 — 10 are reserved). Each TRACE $n$  variable corresponds to a particular type of tracing operation.

In the trace output file for EFGVIEW, MPVM uses keywords to categorize the tracing information being reported. Here are the types of tracing:

Number	Keyword	Type of Debug Performed
1	FLOW	Shows entry into and return from every routine
2	STATUS	Gives status information at various critical decision points
3	MESSAGE	Logs all messages
4	BUFFER	Reports storage buffer management statistics

Number	Keyword	Type of Debug Performed
5	IUCV	Shows IUCV parameter lists and data flow
6	CONSOLE	Shows all console interrupts and commands
7	IMAGE	Step through processing of the MPVM routine that handle screen images
8	Reserved	
9	Reserved	
10	Reserved	

**Note:** The TRACE output may contain sensitive data, including passwords.

MPVM provides three shortcuts for turning all currently-defined TRACE $n$  variables for EFGVIEW on or off:

1. The MPVM macro TRACES
2. A ?TRACES parameter on the MPVM command
3. A CMS global variable. Issue the command:

```
GLOBALV SELECT MPVM SETP TRACE ON
```

For more information on tracing the EFGVIEW user interface module, see the *VM/Pass-Through Facility: Programming Reference*.

The EFGVIEW module checks, at various points in its processing, which (if any) trace variables are set on. If one or more of MPVM's trace flags are set on, MPVM writes tracing information to the virtual printer.

For example, suppose when a user typed in a certain string, EFGVIEW abnormally ended with an operation exception. Follow these steps to trace the problem:

1. Use MPVM to establish a session with the system, and log on to a normal CMS user ID.
2. Spool the virtual printer to the reader with the following command:  

```
?CP SP00L E *
```
3. Get to the last known spot before the error occurred.

4. Issue the following command to turn on all trace flags:  
`?TRACES ON`  
(Or turn individual trace flags on, using a command such as `?SET TRACE1 ON`.)
5. Then issue the command that caused the error.
6. Exit MPVM's EFGVIEW module. (Issuing `?TEMPDISC` or `?SESSTERM *` accomplishes this.)

This will produce a trace file in the virtual reader. If the virtual printer was closed, the trace file will have an identifier of *(none) (none)*; Otherwise, the module closes it automatically, and the resulting file is named MPVM TRACE. This file can be browsed and it will be useful for IBM personnel to track the problem.

**Note: The trace output file could contain sensitive information, such as passwords. Receive the file to a minidisk and edit any sensitive data before giving it to the problem determination personnel.**

To trace EFGVIEW early in its execution, before the TRACE variables are set, use a special parameter when issuing the MPVM command. For example, to turn on EFGVIEW tracing from the very start of a session:

```
mpvm ?TRACES * msys
```

## TRACEn Subcommand

Use the TRACEn subcommand to control the tracing options for the EFGVIEW module, which runs on each MPVM user's virtual machine. For each trace option set ON, MPVM will perform that type of trace. Use the TRACE macro to turn all MPVM traces ON or OFF.

When at least one trace option is ON, the output (MPVM TRACE) is sent to the virtual printer when MPVM exits.

The TRACEn subcommand enables tracing these categories:

- Entry into and return from every routine
- Status information at various critical decision points
- All messages are logged
- Get\_Buffer and Return\_Buffer management
- IUCV parameter lists and data flow
- Console interrupts and commands
- Screen\_Image routine.

## Tracing the EFGVIEW Module

The EFGTRACE exec can also be used as a front end for tracing the EFGVIEW module.

As supplied, EFGTRACE simply invokes the specified module. However, editing EFGTRACE to include a call to another tracing tool, will allow using EFGTRACE to trace the entire EFGVIEW module. Here are the steps to accomplish this:

1. Edit the EFGTRACE exec file to include a call the designated tool.
2. Issue the CMS command `GLOBALV SELECT MPVM SET TRACE ON`.
3. Issue the following from the CMS command line:

When the MPVM command is invoked, and it sees that the CMS global variable TRACE variable is on, it invokes the EFGTRACE exec to trace the EFGVIEW module.

## Logging Messages and Screen Images

Error messages and screen images can be sent to output files.

The MPVM ERRLOG message file is placed on the filemode A minidisk.

Messages and screen images can be directed to error log files using the variables ERRLOG and ERRIMAGE. Note that screen images cannot be logged without messages.

### Notes:

1. If messages are being logged to one of these files, and the disk becomes full, logging stops and message 272W is issued.
2. When logging display images, field attributes are included, but non-displayable fields are not included.

For identification purposes, MPVM messages are issued with a header that has this format:

*EFGabcnnns*

where:

### **EFG**

is the product prefix for MPVM

*abc*

are three characters denoting where the message was issued. If the message was issued from an MPVM macro, these are BSG. If the message was issued from an MPVM internal file, these are the fourth, fifth, and sixth characters of that file's name; see the next section for details.

*nnn*

is the message number

*s* is the severity code

---

## Diagnosing PVMG Problems

PVM provides several ways to diagnose PVMG problems. This section describes methods of PVMG problem determination.

## PVMG Trace Facility

The PVMG trace facility provides tracing for:

- System functions
- Line driver I/O.

Trace records created by the PVMG TRACE command are directed to internal PVMG wrap tables or to GTRACE tables, depending upon the type of trace specified, and trace output destination specified on the TRACE command.

## Tracing PVMG System Activity

Use the TRACE command with the SYSTEM option to assist in problem source determination when internal PVMG trouble is expected. This command combination causes a chronological trace of the activities of key PVMG system components. The PVMG trace facility records requests for service, passed data (if any), execution of services, and status information.

The output resulting from the TRACE command can be directed to:

- An internal PVMG trace table
- A GTRACE table.

To use the GTRACE option, first enter the ETRACE command so that trace events are entered into the GTRACE table.

The TRACE command is described in “TRACE” on page 281.

## PVMG I/O Tracing

Use the TRACE command with the Node *nodeid* and ON option to trace line driver activity for a specific *nodeid*. Tracing begins when line driver activity is started. Trace output is written to an internal PVMG trace I/O wrap table. (A PVMG server may have multiple I/O wrap tables.)

The TRACE command is described in “TRACE” on page 281.

## Other PVMG Problem Determination Actions

PVMG will do basic error checking during its processing. If a problem such as:

- An unexpected post
- An error occurring when posting data to a target application
- A data buffer segment is received out of order

then the session is terminated.

For errors at the task level, PVMG will produce a dump of part or all of the PVMG virtual machine's storage, and PVMG terminates.

For errors regarding a user session, PVMG abnormally terminates the active task and specifies that a dump will be sent to the user's virtual reader. When information is dumped to the user's virtual reader, console messages are written.

## The PVMG Dump Facility

If PVMG terminates abnormally, a dump will be produced. The dump will be a:

- VMDUMP

## Analyzing PVMG Dump Information

Use the Dump Viewing Facility of z/VM to analyze PVMG system failures and then manage problem information and status. Use this facility to interactively retrieve and examine information from PVMG storage dumps to help track and report PVMG problems.

**DMTI end**





## Part 3. Modifying PVM

<b>Chapter 11. Tuning PVM Performance</b>	107
Enhancing PVM Performance	107
Installing MPVM in a Shared Segment	108
Developing Exits	114
<b>Chapter 12. National Languages on the PVM System</b>	129
National Language Overview	129
Changing the Backup Language	130
Changing Message Destinations	130
Changing the PVM Translate Table	131
Creating and Using a Message Repository	132
Rules for Making a custom Repository	134
<b>Chapter 13. Facilities for Automating PVM and PVMG Operations</b>	145
PVMPROP - A Programmable Operator Interface for PVM and PVMG	145
The Command Response Interface	146
PVMPROP/CRI Message Protocol	146
Using Command Prefixes	149
Sample Application	151
<b>Chapter 14. TCP/IP Line Driver Overview</b>	153
The PVM TCP/IP Redirector Task	154
The PVM TCP/IP Line Driver Task	154
A Word About Sockets	155
Tracing Considerations	155
Configuration File Example	155
<b>Chapter 15. Application-to-Application Overview</b>	159
PCCF Data Flow	159
Application-to-Application Communication Sample Program	161
<b>Chapter 16. Customizing CVIEW</b>	165
Listed Consultants — CONSULT IDNAME	165
Program Function Key Assignments — DXGKEYS LIST	166
The Customizing EXEC — DXGINIT	167



## Chapter 11. Tuning PVM Performance

### Enhancing PVM Performance

The PVM system programmer has to weigh the benefits of the various options discussed below against the effect on the overall system performance to determine which options should be chosen for the installation.

Certain CP options enhance PVM performance. One or more of these options can provide adequate performance on heavily loaded systems. PVM spends most of its time waiting for an interrupt to service. When an interrupt occurs, the virtual machine should be dispatched quickly for good user response time.

### SET RESERVE Command

On systems with high paging load, PVM will very likely be paged out when an interrupt comes in. The SET RESERVE command, however, lets most PVM active pages remain in real storage.

The number of reserve pages can be estimated by sampling the working set of the running machine or can be roughly calculated by:

---


$$\begin{aligned} \text{RES} = & 8 + 2(\text{CTCA}) + 5(\text{APPC}) \\ & + 2(\text{IUCV}) + 3(\text{DIAL}) + 2(\text{PASS}) + 2(\text{APP}) \\ & + 2(\text{PCCF}) + 3(\text{MPVM}) + 5(\text{TCPIP}) \end{aligned}$$


---

*where:*

#### **CTCA**

is 1 if any CTCA line drivers are active.

#### **APPC**

is 1 if any APPC line drivers are active.

#### **IUCV**

is 1 if any IUCV line drivers are active.

#### **DIAL**

is 1 if any attached display support tasks are active.

#### **PASS**

is 1 if any users are using the PASSTHRU command.

#### **APP**

is 1 if any PVM users are accessing the local CP system.

#### **PCCF**

is 1 if any PCCF sessions are active.

#### **MPVM**

is 1 if any users are using the MPVM command.

## Installing MPVM in a Shared Segment

### TCPIP

is 1 if any TCPIP line drivers are active.

This algorithm provides a rough estimate. A larger value based on the individual activity and number of users may be required.

## Dispatching Priority

A priority can be specified on the USER or SHARE control statement for PVM in the VM directory. CP uses the specified value, or the default value (100), to determine the PVM dispatching priority. The system operator can alter this value, for a given IPL, by using the CP SET SHARE command.

---

## Installing MPVM in a Shared Segment


If MPVM will be running in a large number of virtual machines, CP paging may be reduced by installing MPVM in a shared segment.

By installing MPVM in a shared segment, there is one copy in shared storage, rather than individual copies in each user's virtual machine.

## Purpose

The EFGDCSS EXEC, loaded to the P684100E's 400 minidisk, will assist in the installation of a shared segment. The EXEC will prompt for the necessary information required to generate a shared segment for a particular environment. The majority of installations can press the *enter* key in response to prompts thereby taking the default installation values for the shared segment generation method (SEGGEN, SAVESEG, or SAVESYS), segment location above or below 16 megabyte line, segment name, and segment page address. The syntax of EFGDCSS follows.

**Note:** PVM supports the SEGGEN method of building shared segments. Alternatively, PVM can use the VMSES/E support for building the MPVM shared segment. For further details on VMSES/E support, see the Program Directory.

EFGDCSS

## Usage Notes

1. EFGDCSS must be issued from a user ID with the appropriate CP CLASS of CLASS E, or its equivalent.
2. The user ID issuing EFGDCSS must have a virtual machine size that includes the entire saved segment. For example, if a command is issued to load the MPVM segment at 5 megabytes, then the user ID that issues the command must have a virtual machine of at least 6 megabytes, since the MPVM shared segment requires .7 megabytes.
3. If using the SEGGEN method of installing a shared segment, then a SYSTEM *segid* will be created on the A-disk and must be copied to the system 190 disk to complete the shared segment installation.
4. Once the shared segment has successfully installed, the command will issue a good completion message. Proceed to the "MPVM Shared Segment Verification" on page 109.

## MPVM Shared Segment Verification

Once the EFGDCSS EXEC has been run and the MPVM shared segment installation has completed, verify the shared segment using the following procedures.

1. Log onto a user ID on the same system as the shared segment with a virtual machine storage size less than or equal to the location of where the MPVM shared segment has been loaded.
2. Issue the MPVM command, the MPVM selection menu should appear.
3. Ensure that the shared segment is being used by using the following procedure.

- a. Logon to a second user ID with class "E" privilege or equivalent, and issue the following command:

```
Q NSS USERS segname
```

Where *segname* is the file name of the PSEG file used for the SEGGEN shared segment generation method, or the shared segment entry name for the SAVESEG or SAVESYS shared segment generation method (depending upon which generation method was used when the EFGDCSS exec was executed).

The output will contain any user IDs currently using the MPVM shared segment.

## Changing the Names of MPVM Shared Segments

The names of MPVM shared segments are hardcoded in the EFGVSTUB MODULE. They are:

- MPVM, for the saved segment less than or equal to 16 megabytes
- MPVMHIGH, for the saved segment above 16 megabytes.

## Purpose

To support custom names for the low or high segments, use the EFGZAP exec. This allows an installation to have multiple (and different) copies of the MPVM shared segment. This capability is useful when testing new MPVM shared segments before they are placed in production. The syntax of the EFGZAP exec follows.

```
»»—EFGZAP—————««
```

## Usage Notes

Before running the EFGZAP exec, verify there is a copy of the EFGVSTUB MODULE available on a minidisk that is in R/W mode so that the EFGVSTUB MODULE can be updated.

### Adding Common MPVM Commands and Macros to an Installation Shared Segment

Load the installation shared segment (default name is CMSINST) with the most commonly used MPVM commands and macros to improve response time.

**Note:** If the SEGGEN method of building shared segments was not used, do the steps that follow.

1. Issue the following command to build CMSINST as a physical shared segment:

```
DCSSGEN fn ft fm CMSINST
```

where *fn ft fm* identifies a file that lists all the commands and macros to load into the saved segment. A file named CMSINST MPVMLIST, supplied on P684100E's 404 minidisk, contains a sample list of these files; to use that file, the command would be:

```
DCSSGEN CMSINST MPVMLIST fm CMSINST
```

The DCSSGEN command—available on MAINT's 193 disk—requires that the executing user ID have authority to issue SAVESYS (class E or equivalent).

2. After loading these often-used macros and commands into the CMSINST installation shared segment, ensure that the CMSINST copies of the macros are found before the versions on MAINT's 38F minidisk. Update the VM system profile to include the following line:

```
SET INSTSEG ON fm
```

where *fm* is a file mode that is accessed ahead of MAINT's 38F minidisk.

If the CMSINST segment is changed, MPVM can be set up to automatically warn users that they must re-IPL CMS to pick up the latest changes.

To accomplish this, the person doing the DCSSGEN must do these steps:

1. Edit the sample file \$CMSINST EXEC on the MPVM common disk (38F) and increment the *level = nn* in that file.
2. Save the \$CMSINST EXEC file along with the others in the CMSINST segment. (The name \$CMSINST EXEC gets added to the CMSINST MPVMLIST used for the DCSSGEN.)

The supplied sample SYSIPROF MPVM macro contains provisions to automatically warn MPVM users, if necessary, to re-IPL to get up-to-date information.

### Saving MPVM Common Disk Information in a Segment

Minidisk file directory information for the MPVM common disk (by default, the PVM user ID's 199 disk) can be stored in a shared segment. All MPVM users who link and access the common disk then share the same *minidisk file directory*, providing faster disk access time and reduced storage usage.

To place this information in a shared segment, do these steps:

1. Get logged on to the MAINT user ID.
2. Create a discontinuous shared segment (DCSS), one segment in size, called *MPVMDISK* to contain the MPVM Common Disk file directory information using the DEFSEG command.
3. LINK to MPVM common disk in R/W mode.

**Note:** The MPVM common disk is the P684100E's 403 disk.

4. Issue the following command to write the segment name on the disk label record:

```
SAVEFD INIT linkaddr label segname
```

where *linkaddr* is the address used in the LINK in step 3 on page 110, *label* is the CMS minidisk label of MPVM common disk and *segname* is the name of the MPVM shared segment (as in MPVMDISK).

**Note:** The SAVEFD command requires class E authority (or equivalent).

5. Issue the following command to save file status tables on the segment:

```
SAVEFD SAVE linkaddr label segname
```

where *linkaddr* is the address used in the link in step 3 on page 110, *label* is the CMS minidisk label of MPVM Common disk, and *segname* is the name of the MPVM shared segment (as in MPVMDISK).

**Note:** The file directory information segment must be resaved (by linking to the disk and then issuing SAVEFD SAVE) any time a file is changed on the disk, or any time the disk is accessed in R/W mode and then released.

For more information on the SAVEFD command and defining segments, see the *z/VM: CP Planning and Administration* publication.

## Fine-Tuning the PVM Server

Some privileged CP commands can be used to tune a PVM server virtual machine:

- Issuing CP's SET SHARE command, which specifies the processor's dispatching priority in relation to other virtual machines.

The format of the command is *SET SHARE userid level*. The priority *level* to supply depends on the particular system, but a value of 1000 is probably a good start. (A priority of 1000 is higher than what an *average* system user gets, 100.)

**Note:** Be sure not to give the PVM server a higher share than is assigned to the GCS and VTAM user IDs.

## Checking Response Time for User Commands

The RTM macro, which is part of the MPVM user interface, can record response times for every MPVM user interaction with a target application. For more information on the RTM macro, see the *VM/Pass-Through Facility: Programming Reference*.

## Customizing MPVM Sessions

**PSPI**

### CAUTION

Only an experienced system programmer should attempt to write exit routines for PVM. The writing of an exit routine requires a thorough knowledge of system programming, of PVM programming conventions, and of PVM internals. Without having this knowledge, attempting to write exit routines or install new exit points runs the risk both of seriously degrading the performance of the system and causing system failure.

Customizing PVM, therefore, is intended to be accomplished primarily by experienced system programmers.

Also, be aware that exit routines involving user display stations, PVM, or target sessions could affect extended data streams. Any alteration to extended data streams can produce errors. Programmers should be aware if data streams are involved. Test exit routines thoroughly.

MPVM provides several ways to set up sessions to meet specific needs.

- Write automatic signon EXECs
- Write EXECs and macros, which are called by MPVM during its session processing, to tailor MPVM to specific user needs
- Supply a meaning for reserved MPVM variable slots
- Write macros that are triggered by certain application conditions, or by the pressing of *hot keys*.

These capabilities are described in the *VM/Pass-Through Facility: Programming Reference*.

## Sample MPVM Exits

Currently, MPVM puts out all messages as fullscreen 3270 Erase/Write datastreams. This exit will put each message in linemode via the REXX SAY command.

---

```
/* Sample exit9 macro with filename of exit9 and filetype of mpvm */
trace 0
if mpvmvalu('msgmode') then
  do
    say mpvmvalue('lastmsg')
    'set lastmsg'
  end
exit 0
```

---

Figure 25. Sample exit for issuing messages

This sample exit1 macro causes an MPVM session to terminate with the '####' string (like passthru)



---

```

/*Sample exit1 macro with filename of exit1 and filetype of mpvm*/
trace 0
'EXTRACT /LASTINP/'
if length(mpvmlastinp.1) = 10      &,
    right(mpvmlastinp.1,4) = '####'
    then 'SESSTERM'
exit 0

```

---

*Figure 26. Sample exit for using ####*

To have these exits automatically set when ever MPVM is used, create an MPVM macro with filename of startup as shown in the following example:

---

```

/*exit9 is called just before MPVM issues a message to the terminal*/
/*exit1 is called anytime the keyboard input is read from the terminal*/
'set exit9 exit9'
'set exit1 exit1'

```

---

*Figure 27. Sample STARTUP MPVM file*

## Developing Exits

A system programmer can make changes to PVM by developing exits. There are three areas of PVM processing that can be customized with exits. These areas, and the PVM modules associated with them, are:

**Session control exits** — These exits can make use of external reference points for session initiation or termination and for command and command response processing. The session control exits are associated with module DVMNUE.

**Networking management exits** — Exits applied here can be used to collect data about a user's session. The networking management exits are associated with module DVMNME.

**CMS user exits** — These exits provide a means for collecting data on a user's I/O activity. The CMS user exits are associated with module DVMCUE, which is part of the DVMUSI module.

### Exit Point and Module Summary

The following table provides a reference to the IBM-defined exit points and the PVM and DVMUSI modules from which they are called. The MPVM component support exits. For information about developing exits for MPVM, see the *VM/Pass-Through Facility: Programming Reference*.

Exit Point	Name	Module
DVMNUE1	Session Initiation Exit	DVMNUE
DVMNUE2	Session Termination Exit	DVMNUE
DVMNUE3	Origin Node Session Request Exit	DVMNUE
DVMNUE4	Intermediate Node Session Request Exit	DVMNUE
DVMNUE5	Command and Response Exit	DVMNUE
DVMNUE6	CINIT Session Initiation Data Exit	DVMNUE
DVMNME		
DVMNMEL	Link Statistics Exit	DVMNME
DVMNMES	Session Statistics Exit	DVMNME
DVMCUEBI	Before Initialization Exit	DVMCUE
DVMCUEAI	After Initialization Exit	DVMCUE
DVMCUEBR	Before Read Exit	DVMCUE
DVMCUEAR	After Read Exit	DVMCUE
DVMCUEBW	Before Write Exit	DVMCUE
DVMCUEAW	After Write Exit	DVMCUE
DVMCUEBF	Before Final Exit	DVMCUE

**Exit routines are never required as part of standard PVM processing.** A module supplied by IBM supports these areas with dummy exit routines that immediately return control to the caller. To provide additional processing for any of these exits, replace and assemble the source code for the corresponding routines. Also, either rebuild the:

- PVM load module using the PVMBLD command (for the session control and the networking management exits)

- DVMUSI module that executes in the user's CMS virtual machine using the PVMGEN command (for the CMS user exits).

**Note:** For PVM 2.1.1, see the Program Directory for applying local modifications. See “Rebuilding PVM for Exits” on page 128 on how to apply exit routines.

See the Appendix H, “PVM Access Security Exits” on page 319 for reference information about the supplied sample PVM access security exits.

## Programming Note

Registers 10 — 13 contain the following information for session and management control exits:

- Register 10 is used as the base addressing register
- Register 11 contains the address of the system control block (DVMSYSB)
- Register 12 contains the address of the session manager's (DVMNET's) task control block (DVMTCB)
- Register 13 contains a pointer to the register stack area.

**Registers 11 and 12 (and the data areas they point to) must not be modified. Registers 10 and 13 may be modified (using only the DVMENR macro), but must be restored (using the DVMRETN macro).**

Any other registers used (including 10 and 13) must be restored to their original values.

To save working registers at entry use the DVMENR macro. To restore the registers upon exiting, use the DVMRETN macro.

For MPVM users, messages pointed to by R1 are truncated to 35 bytes upon return from the session control exits, due to an internal PVM restriction.

Some of the exits discussed in this section may have one or more of the following control blocks addressed by an input register. These control blocks provide data that the exits may use, but must not be altered:

- DVMALT see “DVMALT (Active Link Table Entry) Format” on page 302
- DVMAUT see “DVMAUT (Active User Interface Link Descriptor) Format” on page 304
- DVMBUF see “DVMBUF (Internal Buffer Header)” on page 306
- DVMSYSB see “DVMSYSB (PVM Multitasker System Control Block) Format” on page 312
- DVMTCB for session manager see “DVMTCB” on page 315

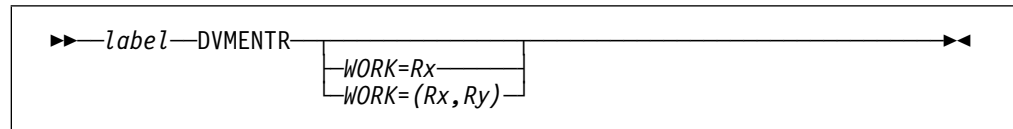
## PVM Macros Used in the Sample Access Security Exits

Three PVM macros are used by the sample access security exits. This section contains a brief description of how they are used.

## DVMENTR

DVMENTR is used to generate the standard entry protocol for subroutines in PVM. It saves various registers on a register stack for later restoration by the DVMRETN macro.

### Format



### Operands

#### label

the routine entry name

#### WORK=R<sub>x</sub>

WORK is the macro variable. R<sub>x</sub> is the register to be saved.

#### WORK=(R<sub>x</sub>,R<sub>y</sub>)

WORK is the macro variable. R<sub>x</sub>,R<sub>y</sub> is the register range to be saved.

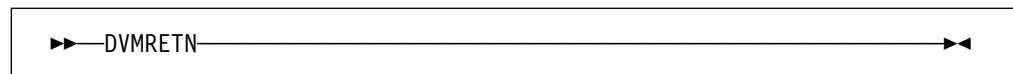
### Usage Notes

1. If the WORK keyword is included, the register or register range specified will be saved on the stack and restored by the DVMRETN macro.
2. Registers always saved by the macro:
  - R10 (used as a base addressing register)
  - R14 (return address register)
  - R13 is *bumped* by four times the number of registers saved including R10, R14, and the number specified on the WORK option.

## DVMRETN

DVMRETN is used to generate the standard return protocol for subroutines in PVM. It restores the registers saved by the DVMENTR macro, (including R10, R13, and R14, and any specified on the WORK option of the DVMENTR macro), then returns to the subroutine's caller using the address specified in R14.

### Format



## DVMXTRN

DVMXTRN is used to map internal subroutine names into external linkedit names.

### Format

```
▶▶—label—DVMXTRN—entry————▶▶
```

### Operands

#### **label**

is the internal subroutine name.

#### **entry**

is the external linkedit name.

## Session Control Exits

The DVMNUE module provides for six session control exit routines. These routines are described in the following sections.

## Session Initiation Exit (DVMNUE1)

Session control exit, DVMNUE1, is provided upon initiation of a PVM session.

This routine is invoked at session initiation to allow tracking sessions in which the target node is the:

- Local node

The exit routine may be programmed to allow session initiation to proceed or not.

Note that execution of either session control exit DVMNUE3 (local requests) or DVMNUE4 (nonlocal requests) precedes execution of this exit.

### **Register Contents on Entry to the User Routine**

Register	Contents																								
<b>R1</b>	Address of a 39-byte area to provide an explanatory message if the session is to be aborted																								
<b>R2</b>	Address of a parameter list containing: <table> <tr> <th>Byte</th><th>Contents</th></tr> <tr> <td><b>00-07</b></td><td>Originating node ID</td></tr> <tr> <td><b>08-15</b></td><td>Originating user ID</td></tr> <tr> <td><b>16-19</b></td><td>Zero</td></tr> <tr> <td><b>20-21</b></td><td>zero.</td></tr> <tr> <td><b>22-23</b></td><td>User display station address (hexadecimal)</td></tr> <tr> <td><b>24-25</b></td><td>User display station class and type</td></tr> <tr> <td><b>26</b></td><td>User display station device model (hexadecimal)</td></tr> <tr> <td><b>27-34</b></td><td>Destination node ID</td></tr> <tr> <td><b>35-42</b></td><td>EBCDIC real terminal address</td></tr> <tr> <td><b>43-50</b></td><td>Origin <i>vmid</i> if this is an MPVM session or PCCF session</td></tr> <tr> <td><b>51-58</b></td><td>Destination <i>userid</i></td></tr> </table>	Byte	Contents	<b>00-07</b>	Originating node ID	<b>08-15</b>	Originating user ID	<b>16-19</b>	Zero	<b>20-21</b>	zero.	<b>22-23</b>	User display station address (hexadecimal)	<b>24-25</b>	User display station class and type	<b>26</b>	User display station device model (hexadecimal)	<b>27-34</b>	Destination node ID	<b>35-42</b>	EBCDIC real terminal address	<b>43-50</b>	Origin <i>vmid</i> if this is an MPVM session or PCCF session	<b>51-58</b>	Destination <i>userid</i>
Byte	Contents																								
<b>00-07</b>	Originating node ID																								
<b>08-15</b>	Originating user ID																								
<b>16-19</b>	Zero																								
<b>20-21</b>	zero.																								
<b>22-23</b>	User display station address (hexadecimal)																								
<b>24-25</b>	User display station class and type																								
<b>26</b>	User display station device model (hexadecimal)																								
<b>27-34</b>	Destination node ID																								
<b>35-42</b>	EBCDIC real terminal address																								
<b>43-50</b>	Origin <i>vmid</i> if this is an MPVM session or PCCF session																								
<b>51-58</b>	Destination <i>userid</i>																								
<b>R11</b>	Address of the system control block (DVMSYSB). Do not modify.																								
<b>R12</b>	Address of session manager's (DVMNET's) task control block (DVMTCB). Do not modify.																								
<b>R13</b>	Address of register save stack area																								

## Session Request Exits (DVMNUE3 and DVMNUE4)

Register	Contents
R14	Return address
R15	Entry address.

The listed control blocks are contained in the PVM macro library (DVMMAC MACLIB).

### *Register Contents on Return to PVM*

Register	Contents
R3-R9	Save on entry using the DVMENTR macro; restore on exit using the DVMRETN macro.
R10-13	R11 and R12 must not be modified. R10 and R13 must be restored to their original value.
R15	Must contain one of the following return codes:  0 Tells PVM to continue the session. 4 Tells PVM to terminate the session. 8 Tells PVM to terminate the session.  If R15 does not contain zero on return, an explanatory message must be returned in the 39-byte area (pointed to by R1 on entry to the exit routine) which is displayed on the user's screen.

## Session Termination Exit (DVMNUE2)

The second session control exit to optional user routines, DVMNUE2, is provided at session termination.

This routine is invoked at session termination to allow tracking the termination of sessions in which the target node is:

- The local node

**Register Contents on Entry to the User Routine:** Register usage and the parameter list passed are very similar to that described above for session initiation. The only difference is that R1 does not contain the address of an area for returning a reason the session cannot continue. Since the session is now completed, the area is not needed.

**Register Contents on Return to PVM:** Register usage and the parameter list return to PVM is also very similar to that described above for session initiation. The only difference is that for the session termination exit, zero must be returned in R15.

## Session Request Exits (DVMNUE3 and DVMNUE4)

The third and fourth session control exits, DVMNUE3 and DVMNUE4, are provided for session requests originating at the local node and for session requests originating outside the local node. They occur when the node receives the session request.

The Origin Node Session Request Exit (DVMNUE3) routine is invoked at session initiation to allow tracking of any or all session requests originating from the local node. The exit may be programmed to allow the request to proceed or not. It can also alter the requested destination.

The Intermediate Node Session Request Exit (DVMNUE4) routine is invoked at session initiation to allow tracking of session requests originating from another node. The exit may be programmed to allow the request to proceed or not. It may also alter the requested destination.

## Register Contents on Entry to the User Routine

Register	Contents																								
R1	Address of a 39-byte area to provide a reason if the session is to be aborted																								
R2	Address of a parameter list containing: <table> <tr> <th>Byte</th><th>Content</th></tr> <tr> <td>00-07</td><td>Originating node ID</td></tr> <tr> <td>08-15</td><td>Originating user ID</td></tr> <tr> <td>16-19</td><td>Unused</td></tr> <tr> <td>20-21</td><td>Zero</td></tr> <tr> <td>22-23</td><td>User display station address (hexadecimal)</td></tr> <tr> <td>24-25</td><td>User display station device class and type</td></tr> <tr> <td>26</td><td>User display station device model (hexadecimal)</td></tr> <tr> <td>27-34</td><td>Destination node ID</td></tr> <tr> <td>35-42</td><td>EBCDIC real terminal address</td></tr> <tr> <td>43-50</td><td>Origin <i>vmid</i> if this is an MPVM or PCCF session</td></tr> <tr> <td>51-58</td><td>Destination <i>userid</i></td></tr> </table>	Byte	Content	00-07	Originating node ID	08-15	Originating user ID	16-19	Unused	20-21	Zero	22-23	User display station address (hexadecimal)	24-25	User display station device class and type	26	User display station device model (hexadecimal)	27-34	Destination node ID	35-42	EBCDIC real terminal address	43-50	Origin <i>vmid</i> if this is an MPVM or PCCF session	51-58	Destination <i>userid</i>
Byte	Content																								
00-07	Originating node ID																								
08-15	Originating user ID																								
16-19	Unused																								
20-21	Zero																								
22-23	User display station address (hexadecimal)																								
24-25	User display station device class and type																								
26	User display station device model (hexadecimal)																								
27-34	Destination node ID																								
35-42	EBCDIC real terminal address																								
43-50	Origin <i>vmid</i> if this is an MPVM or PCCF session																								
51-58	Destination <i>userid</i>																								
R11	Address of the system control block (DVMSYSB). Do not modify.																								
R12	Address of session manager's (DVMNET's) task control block (DVMTCB). Do not modify.																								
R13	Address of the register save stack area																								
R14	Return address																								
R15	Entry address.																								

The listed control blocks are contained in the PVM macro library (DVMMAC MACLIB).

## Register Contents on Return to PVM

Register	Contents
R3-R9	Save on entry using the DVMENTR macro; restore on exit using the DVMRETN macro.
R10-13	R11 and R12 must not be modified. R10 and R13 must be restored to their original value.
R15	<p>Must contain one of the following return codes:</p> <ul style="list-style-type: none"> <li><b>0</b> Tells PVM to allow the session to continue.</li> <li><b>4</b> Tells PVM to try another destination (because the destination has been altered by the installation within the exit routine. If the destination has not been altered, the session is terminated.)</li> </ul> <p><b>Note:</b> The capability to alter node names allows for use of synonym node names. An example is an installation which has two MVS systems running CICS accessible through PVM. The installation wants the ability to change which MVS system the users actually access for a specific CICS application. The installation can setup a synonym node name for the application, for example, CICSAPP1. A session control exit can be designed to recognize the request for <i>node</i> CICSAPP1, choose which real node the user is to access, and then alter the parameter list to point to the new (real) node as the destination.</p> <ul style="list-style-type: none"> <li><b>8</b> Tells PVM to terminate the session.</li> </ul> <p>If R15 does not contain zero on return, an explanatory message must be returned in the 39-byte area (pointed to by R1 on entry to the exit routine) which is displayed on the user's screen.</p>

### Command and Response Exit (DVMNUE5)

The fifth session control exit, DVMNUE5, is provided before PVM processes any commands or command responses originating at another node.

This routine is invoked before PVM processes any commands or command responses originating at another node. The exit may be programmed to allow the command or response to continue or not. It may also change the destination of the command or response. If the target node is the origin node, this exit is not entered.

#### *Register Contents on Entry to the User Routine*

Register	Contents												
R1	Address of a 39-byte area to provide a reason if the session is to be aborted.												
R2	Address of a parameter list containing: <table> <tr> <th>Byte</th><th>Content</th></tr> <tr> <td>00-07</td><td>Originating node ID</td></tr> <tr> <td>08-15</td><td>Originating user ID</td></tr> <tr> <td>16-19</td><td>Address of the buffer containing the command or command response</td></tr> <tr> <td>20-26</td><td>Unused</td></tr> <tr> <td>27-34</td><td>Destination node ID.</td></tr> </table>	Byte	Content	00-07	Originating node ID	08-15	Originating user ID	16-19	Address of the buffer containing the command or command response	20-26	Unused	27-34	Destination node ID.
Byte	Content												
00-07	Originating node ID												
08-15	Originating user ID												
16-19	Address of the buffer containing the command or command response												
20-26	Unused												
27-34	Destination node ID.												
R3	Address of DVMALT control block for the outgoing node. If the value in R3 is not positive, then the destination node is unknown. The link referenced by R3 may not actually be available. The condition code from the route lookup routine is in the high byte of R6. An <i>SPM R6</i> instruction restores the condition code. A CC=0 indicates the path is available. A CC=1 indicates the node is unknown, and a CC=2 indicates the path is not connected. Do not modify.												
R4	Address of the DVMALT control block for incoming link												
R5	Address of the message packet (DVMBUF). Do not modify.												
R11	Address of the system control block (DVMSYSB). Do not modify.												
R12	Address of session manager's (DVMNET's) task control block (DVMTCB). Do not modify.												
R13	Address of the register save stack area												
R14	Return address												
R15	Entry address.												

Access the listed control blocks by using the PVM macro library (DVMMAC MACLIB).

Buflen is a halfword in the Internal Buffer Header (DVMBUF) data area containing the length of the command. Bufdata contains the start of the command. Check the BUFSPEC field in the Internal Buffer Header before rejecting the message. If the reply bit in BUFSPEC is on, do not reject the message. For further information on DVMBUF, see "DVMBUF (Internal Buffer Header)" on page 306.

#### *Register Contents on Return to PVM*

Register	Contents
R3-R9	Save on entry using the DVMENTR macro; restore on exit using the DVMRETN macro.
R10-13	R11 and R12 must not be modified. R10 and R13 must be restored to their original value.



Register	Contents
<b>R15</b>	<p>Must contain one of the following return codes:</p> <p><b>0</b> Tells PVM to process the command or command response normally.</p> <p><b>4</b> Tells PVM to try another destination (because the destination has been altered by the installation within the exit routine.)</p> <p><b>8</b> Tells PVM to return the reason code as a response.</p> <p>If R15 does not contain zero on return, an explanatory message must be returned in the 39-byte area (pointed to by R1 on entry to the exit routine) which is returned as a response to the original command.</p>

## CINIT Session Initiation Data Exit (DVMNUE6)

The sixth session control exit, DVMNUE6, is provided for examining the CINIT data that will be passed from an MPVM user for a session request to an SNA application that requires CINIT data.

This exit is provided for examining CINIT data that will be passed for a session control request to an SNA application that requires CINIT data. The routine will be invoked at session initiation time by the local node. The CINIT data can be altered and added to by the exit, but cannot exceed a length of 255 bytes. The exit can be programmed to allow the request to proceed or not proceed.

### *Register Contents on Entry to the CINIT Exit*

Register	Contents																						
<b>R1</b>	Address of a 39-byte area to provide a reason if the session is to be aborted.																						
<b>R2</b>	<p>Address of a parameter list containing:</p> <table> <tr> <th>Byte</th><th>Content</th></tr> <tr> <td><b>00-07</b></td><td>Originating node ID</td></tr> <tr> <td><b>08-15</b></td><td>Originating session ID (in form of *SESnnnn)</td></tr> <tr> <td><b>16-19</b></td><td>Unused</td></tr> <tr> <td><b>20-21</b></td><td>Zero</td></tr> <tr> <td><b>22-23</b></td><td>User display station address (hexadecimal)</td></tr> <tr> <td><b>24-25</b></td><td>User display station device class and type</td></tr> <tr> <td><b>26</b></td><td>User display station device model (hexidecimal)</td></tr> <tr> <td><b>27-35</b></td><td>Destination node ID</td></tr> <tr> <td><b>35-42</b></td><td>EBCDIC real terminal address</td></tr> <tr> <td><b>43-50</b></td><td>MPVM user real VM ID</td></tr> </table>	Byte	Content	<b>00-07</b>	Originating node ID	<b>08-15</b>	Originating session ID (in form of *SESnnnn)	<b>16-19</b>	Unused	<b>20-21</b>	Zero	<b>22-23</b>	User display station address (hexadecimal)	<b>24-25</b>	User display station device class and type	<b>26</b>	User display station device model (hexidecimal)	<b>27-35</b>	Destination node ID	<b>35-42</b>	EBCDIC real terminal address	<b>43-50</b>	MPVM user real VM ID
Byte	Content																						
<b>00-07</b>	Originating node ID																						
<b>08-15</b>	Originating session ID (in form of *SESnnnn)																						
<b>16-19</b>	Unused																						
<b>20-21</b>	Zero																						
<b>22-23</b>	User display station address (hexadecimal)																						
<b>24-25</b>	User display station device class and type																						
<b>26</b>	User display station device model (hexidecimal)																						
<b>27-35</b>	Destination node ID																						
<b>35-42</b>	EBCDIC real terminal address																						
<b>43-50</b>	MPVM user real VM ID																						
<b>R3</b>	Address of start of CINIT data																						
<b>R4</b>	Length of CINIT data																						
<b>R11</b>	Address of the system control block (DVMSYSB). Do not modify.																						
<b>R12</b>	Address of session manager's (DVMNET's) task control block (DVMTCB). Do not modify.																						
<b>R13</b>	Address of the register save stack area																						
<b>R14</b>	Return address																						
<b>R15</b>	Entry address.																						

The listed control blocks are contained in the PVM macro library (DVMMAC MACLIB).

### *Register Contents on Return to PVM*

## Statistics Exit (DVMNMEL)

Register	Contents
<b>R0-R3, R5-R9</b>	Save on entry using the DVMENTR macro; restore on exit using the DVMRETN macro.
<b>R4</b>	The original or updated CINIT length
<b>R10-13</b>	R11 and R12 must not be modified. R10 and R13 must be restored to their original value.
<b>R15</b>	<p>Must contain one of the following return codes:</p> <ul style="list-style-type: none"><li><b>0</b> Tells PVM to allow the session to continue</li><li><b>4</b> Tells PVM to try another destination (because the destination has been altered by the installation within the exit routine. If the destination has not been altered, the session is terminated.)</li></ul> <p>The capability to alter node names allows for use of synonym node names. An example is an installation which has two MVS systems running CICS accessible through PVM. The installation wants the ability to change which MVS system users actually access for a specific CICS application. The installation can setup a synonym node name for the application, for example, CICSAPP1. A session control exit can be designed to recognize the request for "node" CICSAPP1, choose which real node the user is to access, and then alter the parameter list to point to the new (real) node as the destination.</p> <ul style="list-style-type: none"><li><b>8</b> Tells PVM to terminate the session</li></ul> <p>If R15 does not contain zero on return, an explanatory message must be returned in the 39-byte area (pointed to by R1 on entry to the exit routine) which is returned as a response to the original command.</p>

## Network Management Exits

The DVMNME module provides for two network management exit routines:

**Link Statistics Exit (DVMNMEL)** — This routine is called at:

- PVM startup
- Link startup
- Link connect
- Link disconnect
- Link deactivation
- On a periodic basis for a connected link.

The exit routine records the statistics in either a disk file, a spool file, or through the CP ACCOUNT facility.

**Session Statistics Exit (DVMNMES)** — This routine records session statistics at session termination. The exit is **not** called when the input byte count passed to the DVMNMEL exit is zero. The routine is entered at each node along the session path.

The exit routine records the statistics in either a disk file, a spool file, or through the CP ACCOUNT facility.

## Link Statistics Exit (DVMNMEL)

The network statistics exit, DVMNMEL, is provided at:

- PVM startup
- Link startup
- Link connect
- Link disconnect
- Link deactivation
- On a periodic basis for a connected link.

**Register Contents on Entry to the User Routine**

Register	Contents
<b>R0</b>	<p>Contains a value that specifies when this exit was called.</p> <p><b>0</b> Indicates this is the initial startup of PVM. In this case, there is no data in the parameter list.</p> <p><b>1</b> Indicates a link startup. At this point, PVM has processed the START LINE command and is ready to attach a line driver task, and the line has not entered the connect state. Only the first three fields of the parameter list are valid at this call.</p> <p><b>2</b> The link has been connected. At this point, the link has gone from the active state to the connect state. Only the first three fields of the parameter list are valid at this call.</p> <p><b>3</b> The link has gone from the connect state to the active state.</p> <p><b>4</b> The link has gone from the active state to the inactive or down state. Only the first three fields of the parameter list are valid at this call.</p> <p><b>5</b> The link is in the connect state. This entry is invoked on a periodic basis set by the STATTIME record in the configuration file.</p>
<b>R1</b>	<p>Address of a parameter list containing:</p> <p><b>00-03</b> Address of the DVMALT control block for the link</p> <p><b>04-05</b> Virtual address of the line (hex)</p> <p><b>06-07</b> Real address of the line (hex).</p> <p>The following fields are available only if R0 is 3 or 5:</p> <p><b>08-11</b> Input block count</p> <p><b>12-15</b> Output block count</p> <p><b>16-19</b> Input byte count</p> <p><b>20-23</b> Output byte count</p> <p><b>24-25</b> Extra records per block on input</p> <p><b>26-27</b> Extra records per block on output</p> <p><b>28-29</b> Link delay time in milliseconds (network links only).</p>
<b>R3</b>	Address of the DVMALT control block. Do not modify.
<b>R11</b>	Address of the DVMSYSB control block. Do not modify.
<b>R12</b>	The task control block of the running task. Do not modify.
<b>R14</b>	Return address
<b>R15</b>	Entry address.

The listed control blocks are contained in the PVM macro library (DVMMAC MACLIB).

The *extra records* fields indicate the degree of blocking on the link. If a transmission block contains four message packets, then the block count is incremented by one, and the extra records field is incremented by 3. Only network links have the capability of blocking records.

The *link delay time* field indicates how long it took a special time stamped record to be sent down the link and returned by the other side. This field is available only on network links.

**Register Contents on Return to PVM**

Register	Contents
<b>R10-R13</b>	R11 and R12 must not be modified. R10 and R13 must be restored to their original value.

## Session Statistics Exit (DVMNMES)

The session statistics exit, DVMNMES, records session statistics at session termination. This exit is not called if the input byte count from the Link Statistics Exit is zero.

### *Register Contents on Entry to the User Routine*

Register	Contents
R1	Address of a parameter list containing: <b>00-07</b> Originating node ID <b>08-15</b> Originating user ID <b>16-23</b> Destination node <b>24-31</b> Destination user ID <b>32-35</b> Session start time (top 32 bits of TOD clock) <b>36-39</b> Input byte count <b>40-43</b> Output byte count <b>44-45</b> Origin device real address <b>46-49</b> Real device class, type, model, and features <b>50-57</b> Origin VMID if it's an IUCV session, or blank <b>58-65</b> Destination VMID if this is an IUCV session, or blank.
R11	Address of the DVMSYSB control block. Do not modify.
R12	The task control block of the running task. Do not modify.
R14	Return address
R15	Entry address.

The listed control blocks are contained in the PVM macro library (DVMMAC MACLIB).

### *Register Contents on Return to PVM*

Register	Contents
R10-R13	R11 and R12 must not be modified. R10 and R13 must be restored to their original value.

## CMS User Exits

There are seven user exits in the DVMCUE module of DVMUSI. These exits support a CMS user virtual machine participating in a PVM PASSTHRU session. The CMS User exits are:

- Before Initialization (DVMCUEBI)
- After Initialization (DVMCUEAI)
- Before Read (DVMCUEBR)
- After Read (DVMCUEAR)
- Before Write (DVMCUEBW)
- After Write (DVMCUEAW)
- Before Final (DVMCUEBF).

For each exit routine:

- R0-12 may be altered and need not be restored.
- R13 must be restored to its original value.
- R15 is used for linkage. Use R15 for the exit return code. Clear R15 for a normal return.

On return, R0 and R1 have significance for some routines.

## Before Initialization Exit (DVMCUEBI)

This exit, DVMCUEBI, is provided to allow for any user initialization activity.

### *Register Contents on Entry to the User Routine*

Register	Contents
R14	Return address
R15	Entry address.

### *Register Contents on Return to PVM*

Register	Contents
R13	Must be restored to its original value.

This exit does not process return codes.

## After Initialization Exit (DVMCUEAI)

This exit, DVMCUEAI, is scheduled after the PVM CMS application module, DVMUSI, has completed its initialization. The PASSTHRU command parameters have been processed and validated. These parameters cannot be changed by this exit routine. A session has not yet been established with the PVM system.

This exit is always scheduled prior to any of the read and write exits.

### *Register Contents on Entry to the User Routine*

Register	Contents																																												
R1	Address of a parameter list containing: <table> <tr> <th>Bytes</th><th>Meaning</th></tr> <tr> <td>00-07</td><td>Node ID of the target session node (if entered with PASSTHRU command)</td></tr> <tr> <td>08-09</td><td>Reserved</td></tr> <tr> <td>10-15</td><td>Reserved</td></tr> <tr> <td>16-23</td><td>The PVM user ID</td></tr> <tr> <td>24-31</td><td>Temporary disconnect character string</td></tr> <tr> <td>32-33</td><td>Temporary disconnect character string length</td></tr> <tr> <td>34-39</td><td>Reserved</td></tr> <tr> <td>40-47</td><td>Session termination character string</td></tr> <tr> <td>48-49</td><td>Session termination character string length</td></tr> <tr> <td>50-55</td><td>Reserved</td></tr> <tr> <td>56-59</td><td>Notepad buffer address (if Notepad specified, otherwise empty)</td></tr> <tr> <td>60-63</td><td>Reserved</td></tr> <tr> <td>64-67</td><td>Console input buffer address</td></tr> <tr> <td>68-71</td><td>Reserved</td></tr> <tr> <td>72</td><td>Display station model</td></tr> <tr> <td>73-79</td><td>Reserved</td></tr> <tr> <td>80-87</td><td>Secondary route, if specified</td></tr> <tr> <td>88-90</td><td>I/O verification (<i>ON</i> or blanks)</td></tr> <tr> <td>91-95</td><td>Reserved</td></tr> <tr> <td>96-100</td><td>User requested language, if specified</td></tr> <tr> <td>101-103</td><td>Reserved</td></tr> </table>	Bytes	Meaning	00-07	Node ID of the target session node (if entered with PASSTHRU command)	08-09	Reserved	10-15	Reserved	16-23	The PVM user ID	24-31	Temporary disconnect character string	32-33	Temporary disconnect character string length	34-39	Reserved	40-47	Session termination character string	48-49	Session termination character string length	50-55	Reserved	56-59	Notepad buffer address (if Notepad specified, otherwise empty)	60-63	Reserved	64-67	Console input buffer address	68-71	Reserved	72	Display station model	73-79	Reserved	80-87	Secondary route, if specified	88-90	I/O verification ( <i>ON</i> or blanks)	91-95	Reserved	96-100	User requested language, if specified	101-103	Reserved
Bytes	Meaning																																												
00-07	Node ID of the target session node (if entered with PASSTHRU command)																																												
08-09	Reserved																																												
10-15	Reserved																																												
16-23	The PVM user ID																																												
24-31	Temporary disconnect character string																																												
32-33	Temporary disconnect character string length																																												
34-39	Reserved																																												
40-47	Session termination character string																																												
48-49	Session termination character string length																																												
50-55	Reserved																																												
56-59	Notepad buffer address (if Notepad specified, otherwise empty)																																												
60-63	Reserved																																												
64-67	Console input buffer address																																												
68-71	Reserved																																												
72	Display station model																																												
73-79	Reserved																																												
80-87	Secondary route, if specified																																												
88-90	I/O verification ( <i>ON</i> or blanks)																																												
91-95	Reserved																																												
96-100	User requested language, if specified																																												
101-103	Reserved																																												
R14	Return address																																												
R15	Entry address.																																												

### *Register Contents on Return to PVM*

Register	Contents
<b>R13</b>	Must be restored to its original value.
<b>R15</b>	One of the following return codes: <b>0</b> Normal return <b>4</b> Reset Notepad function and initially bypass the Before Read exit. <b>8</b> Terminate, go to Before Final exit. <b>Other</b> Continue processing, ignore any other return code setting.

## Before Read Exit (DVMCUEBR)

This exit, DVMCUEBR, is scheduled prior to checking the console stack to see if there is any data.

### *Register Contents on Entry to the User Routine*

Register	Contents
<b>R14</b>	Return address
<b>R15</b>	Entry address.

### *Register Contents on Return to PVM*

Register	Contents
<b>R13</b>	Must be restored to its original value
<b>R15</b>	One of the following return codes: <b>0</b> Normal return <b>4</b> Bypass next read. Exit routine is returning an input record. Bypass checking for entry on the console. Routine should set R1 to point to the input record and R0 should contain the input record length. Go to the After Read exit. <b>8</b> Terminate, go to the Before Final exit. <b>Other</b> Continue processing and ignore any other return code setting.

## After Read Exit (DVMCUEAR)

This exit, DVMCUEAR, is scheduled after input has been read from the CMS virtual console or when a return code of 4 is set from the Before Read exit.

### *Register Contents on Entry to the User Routine*

Register	Contents
<b>R1</b>	Address of a parameter list containing: <b>00-03</b> Input record length <b>04-07</b> Input record address <b>08-11</b> One of the following literals is encoded: NORM or SPEC. NORM is used for NORMal readings from the console. SPEC is used for SPECial types of input. Special input types result from reading the input requested by the Before Write Exit.
<b>R14</b>	Return address
<b>R15</b>	Entry address.

**Register Contents on Return to PVM**

Register	Contents
<b>R13</b>	Must be restored to its original value.
<b>R15</b>	One of the following return codes: <b>0</b> Normal return <b>4</b> Return to the Before Read exit processing <b>8</b> Terminate, go to Before Final exit. <b>Other</b> Continue processing and ignore any other return code setting.

**Before Write Exit (DVMCUEBW)**

This exit, DVMCUEBW, is scheduled prior to writing to the CMS virtual console.

If Notepad is specified with the PASSTHRU command and the After Initialization exit turns off file writes, then the Notepad image goes into a buffer area. If file writes are not turned off, then the Notepad image is also written to the file PASSTHRU DATA.

**Register Contents on Entry to the User Routine**

Register	Contents
<b>R1</b>	Address of a parameter list containing: <b>00-03</b> Output record length <b>04-07</b> Output record address.
<b>R14</b>	Return address
<b>R15</b>	Entry address.

**Register Contents on Return to PVM**

Register	Contents
<b>R13</b>	Must be restored to its original value.
<b>R15</b>	One of the following return codes: <b>0</b> Normal return <b>4</b> Bypass write operation. The Notepad facility is called to update the Notepad buffer (pointed to by bytes 56-59 of the DVMCUEAI exit parameter list). Update the Notepad file (if active), then schedule the After Write exit. <b>8</b> Terminate, go to Before Final exit. <b>12</b> Bypass write operation. Perform a read from the virtual console. The read command to use is stored in the first byte position of the output record (the address of which is at bytes 04-07 of the parameter list). The read command is a Read Modified (X'06') or a Read Buffer (X'02'). <b>Other</b> Continue processing and ignore any other return code setting.

**After Write Exit (DVMCUEAW)**

This exit, DVMCUEAW, is scheduled:

- After writing to the CMS virtual console
- After a final response VMCF external interrupt
- For each segment except the last one from PVM
- As a general entry prior to entering into the wait states for external interrupts or console I/O interrupts.

### *Register Contents on Entry to the User Routine*

Register	Contents
R14	Return address
R15	Entry address.

### *Register Contents on Return to PVM*

Register	Contents
R13	Must be restored to its original value.
R15	One of the following return codes: <div style="margin-left: 20px;"> <b>0</b> Normal return  <b>4</b> Perform another write. The exit returns a pointer to a 40 byte VMCF parameter list in R1; which defines a VMCF Send operation from PVM. The DVMUSI module then performs a VMCF Receive using that parameter list. This results in either the transmission of data from PVM or a VMCF return code indicating no further data exists.  <b>8</b> Terminate, go to Before Final exit.  <b>Other</b> Continue processing and ignore any other return code setting. </div>

## Before Final Exit (DVMCUEBF)

This exit, DVMCUEBF, is scheduled prior to terminating a user session in the CMS virtual machine.

### *Register Contents on Entry to the User Routine*

Register	Contents
R14	Return address
R15	Entry address.

### *Register Contents on Return to PVM*

Register	Contents
R13	Must be restored to its original value
R15	On entry back to CMS, whatever is in R15 is used as a return code.

## Rebuilding PVM for Exits

**Note:** See the Program Directory for applying local modifications.

**PSPI end**



---

## Chapter 12. National Languages on the PVM System

PVM supports several national languages. This chapter describes how to take advantage of these languages.

---

### National Language Overview

The PVM product is shipped with mixed-case American English and uppercase English languages. The system default language is specified on the LANG record in the PVM and PVMG configuration files.

PASSTHRU and MPVM users see PVM messages and menus in the same language being used in their CMS session, provided that PVM is enabled for that language. If a language identifier is specified on the MPVM command, MPVM checks whether it is enabled for the language, and if so, selects it as the default language. If the PVM server is not enabled for this language, a warning message is issued. This means warning messages from the PVM server are issued in its default language. All other messages are in the user's selected language. PVM, MPVM and PVMG users can select the language to be used for a session or for a command response.

PVM messages and menus can be translated into local languages or dialects in order to meet the needs of the users. See “Creating and Using a Message Repository” on page 132 for information on how to do this.

### Languages Available Through IBM

All languages are identified by a 1-to-5 character language ID. The language ID for uppercase English, the default language, is *UCENG*. The IBM-provided languages are:

**AMENG** American English

**GER** German

**KANJI** Japanese

**UCENG** Uppercase English

**Note:** Some languages have character sets that require special hardware. Be sure that all display devices in the configuration can properly display the character set of any new language added to the system. If they cannot, it may be possible to create a new language repository that uses the necessary code page.

### Backup National Language

In some instances PVM (and PVMG) cannot use the specified national language when sending or displaying messages and menus. In these cases, PVM and PVMG use a *backup* language.

PVM and PVMG use the backup language:

- As the system national language when an installation does not use the LANG record in the configuration file
- When communicating with PVM systems that are at an earlier release level with respect to the messages being sent

## Changing Message Destinations

- When sending messages which have been added to PVM between releases to other systems (for example, messages added to PVM by a service update),
- Sending messages to local display stations which do not support the double-byte character set of the national language that would otherwise be used by default.

PVM uses uppercase American English (UCENG) as the backup language. See “Changing the Backup Language” for instructions on how to change the PVM backup language.

---

## Changing the Backup Language

When changing the PVM, MPVM, PASSTHRU (DVMUSI), or PVMG backup language, first consider these points:

- The backup language should be the language that is most universally understood by all of the users throughout the PVM network.
- The backup language must be an enabled PVM or PVMG language. It should also be the same language at every system in the PVM network.
- Do not use a language that contains double-byte characters as the backup language unless all display stations in the PVM network support those characters.

Follow the steps described below to change the backup language. The steps are performed while logged on to the MAINT user ID.

1. Enter these two commands:

```
pvmcmp dvmnlangid net
pvmcmp efgnlangid net
```

where *nlangid* is the language ID of the new backup language.

**Note:** Use the CTL option of the PVMCMP command if there are any update files for PVM message repositories or for the PVMDEST FILE.

2. Enter these two commands:

```
pvmcmp dvmolangid nat
pvmcmp efgolangid nat
```

where *olangid* is the language ID of the old backup language.

**Note:** Use the CTL option of the PVMCMP command if there are any update files for PVM message repositories or for the PVMDEST FILE.

**Note:** See the Program Directory regarding building PVM 2.1.1 and applying local modifications.

---

## Changing Message Destinations

PVM and PVMG use a file named PVMDEST FILE to determine the destinations for PVM messages. (The PVMDEST FILE is on the PVM source staging minidisk.) Records in this file map a message number to keywords describing the message destinations. Keywords in the file include:

OP      Indicates that the message is sent to the CP operator (the OPID configuration file record)

**CONS** Indicates that the message is sent to the PVM console (PVM user ID) or to the PVMG console (PVMG user ID)

For PVM, **CONS** also indicates that the message is sent to the audit file.

**ORIG** Indicates that the message is sent to the command originator for command responses.

**PROP** Indicates that the message is sent to the PVMPROP virtual machine (PVMPROP configuration file record)

For example, the PVMDEST FILE record:

0209	OP	CONS	PROP
------	----	------	------

Indicates that message 209 is sent to the CP operator, the PVM user ID, and to the PVMPROP user ID (if one exists).

The PVMDEST FILE record:

1209	OP	CONS	PROP
------	----	------	------

Indicates that message 1209 is sent to the CP operator, the PVMG user ID, (messages numbered 1000 - 1999 are PVMG messages) and to the PVMPROP user ID (if one exists).

To change the destinations for a PVM or PVMG message, follow these steps:

1. Change message entry in the PVMDEST FILE.
2. Compile all of the message repositories used by the PVM MODULE (that is, all of the *DVMlangid* REPOS files). Use the PVMCMP command described on page 210. Use the *NAT* option for all repositories, except for the backup language repository; use the *NET* option when compiling the repository for the backup language. (Create an update file using CMS UPDATE. Use the CTL option of the PVMCMP command.)

**Note:** See the Program Directory regarding building PVM and applying local modifications.

---

## Changing the PVM Translate Table

PVM uses a 256-byte translate table to translate lowercase American English alphabetic characters into uppercase when reading input from a PVM menu. To modify the translation algorithm for the needs of another language, or to accommodate the use of a different code page, follow these steps:

1. Change the translate table located at the label DVMTRN in the DVMTBL ASSEMBLE file. (The DVMTBL ASSEMBLE file can be found on the PVM source staging minidisk.)
2. Recompile DVMTBL ASSEMBLE file using VMFASM. Enter:  
vmfasm dvmtbl pvm

where `pvm` indicates the PVM CNTRL file that is distributed with PVM.

**Note:** See the Program Directory regarding building PVM 2.1.1 and applying local modifications.

---

## Creating and Using a Message Repository

The PVM, PVMG, PASSTHRU (DVMUSI) and MPVM message repositories can be changed or new ones can be created. Change the message repository to:

- Change the wording of messages or menus
- Translate messages to another language or dialect
- Change the format of time stamps on messages.
- Change only certain characters so that they display properly with the local code page.

For example, in the United Kingdom, copying the `nnnAMENG` repository and altering the copy allows PVM to display characters properly using the United Kingdom's code page.

The steps involved with creating and then using a new message repository or changing a PVM message repository are:

1. Print and read *Section One* of the READTHIS FIRST file found on 5684100E's 2C2 minidisk.
2. Create a message repository file using XEDIT
3. Compile the repository
4. Verify the accuracy of the message file
5. Make the repository file available.

## Creating a Message File

There are two sets of message repositories in PVM — one for the PVM MODULE and the PVMG MODULE, and one for the DVMUSI and EFGVIEW MODULEs that run in a PVM user's CMS session when the PASSTHRU or MPVM commands are used. Messages with identifiers for PVM and PASSTHRU (DVMUSI) range from 0-999. Messages with identifiers for MPVM and PVMG range from 1000 - 1999. Repositories containing definitions for messages and screens used by the PVM MODULE and PVMG MODULE are named `DVMlangid` REPOS, where *langid* is a 1-to-5-character language identifier. Repositories containing definitions for messages used by the DVMUSI MODULE and EFGVIEW MODULE are named `EFGlangid` REPOS.

To create a message repository file, follow certain notation rules. These rules are explained in "Rules for Making a custom Repository" on page 134.

One of the IBM-supplied message files can be used as a reference to create a message file. (Use the CMS UPDATE command to make sure that the most current level of the IBM-supplied message file is used.)

## Compiling the Message File

Note that some message text may be identical with two different messages. For example, messages 0071 and 1071 contain identical message texts. Ensure that new message text for messages that contain identical message text (like 0071 and 1071) is identical in any new message repository.

Compile the message file using the PVMCMP command. Enter:

```
pvmcmp repository
```

where *repository* is the filename of the message repository.

**Note:** Use the CTL option of the PVMCMP command if there are any update files for the file PVMDEST FILE.

PVMCMP displays error messages on the display station and in the listing file for any invalid statement it finds in the message file. If there are any errors, XEDIT the source file and correct any mistakes. The listing file has the same filename as the repository file and a filetype of *\$LISTING*.

When a message repository does not contain syntax errors, PVMCMP creates an output file with the same filename as the input file and a filetype of *TEXT*. This TEXT file contains the machine readable form of the message file that the PVM message processor recognizes.

See page 210 for details on the PVMCMP command.

## Verifying the Message File

To verify that a message and screen definitions are functionally accurate:

1. The *DVMLangid \$COMPnnn* file should be identical with the IBM-supplied *DVMUCENG \$COMPnnn* file.
2. The *EFGLangid \$COMPnnn* file should be identical with the IBM-supplied *EFGUCENG \$COMPnnn* file.
3. The *DVMLangid \$SYNCnnn* and the *EFGLangid \$SYNCnnn* files should be identical.

Use the CMS COMPARE command to verify that these files are identical. See "PVMCMP" on page 210 for more information about the files created when a message file is compiled.

## Making the Message File Available

Once the message file has been compiled, make the message file available for use. To do this, perform these steps:

1. For each language file created, update the DVMLNG ASSEMBLE file found on the PVM source staging minidisk. Add the language ID to the line in the file that looks like this:

```
LANGS IDS=(AMENG,UCENG,GER,KANJI)
```

For example, to add the language identified by XYZ, specify:

```
LANGS IDS=(AMENG,UCENG,GER,KANJI,XYZ)
```

2. Recompile DVMLNG ASSEMBLE file using VMFASM. Enter:

```
vmfasm dvm1ng pvm
```

where pvm indicates the PVM CNTRL file.

3. For each language installed, add a line to the PVMMOD EXEC file for PVM 2.1.1 in the form *DVModlangid*.

**Note:** See the Program Directory regarding building PVM 2.1.1 and applying local modifications.

---

## Rules for Making a custom Repository

There are five types of records in a repository file:

- Comments
- Master
- Time stamps
- Message and screen definitions
- Control lines.

Blank lines are also allowed in a repository file.

Control lines should not be changed when creating a repository. The control lines are used to increment the compatibility level for subsequent releases and updates. Compatibility levels are used by PVM and PVMG for message level synchronization in the network. For example, if a PVM Version 2 node needs to send a PVM Version 2 message to a PVM 1.4 node, the message is sent to the PVM 1.4 node in the network (backup) language. If the message is a PVM 1.4 message, the message is sent in encoded format.

## Comment Record

A comment record begins with an asterisk in column 1. Comment records can be placed anywhere in the message repository and should describe what is in the file. Use columns 1 through 63 for comments.

When creating a file using XEDIT, make the comment record in the following format:

```
====      .
====      .
==== * This is a sample comment record
      |...+...1...+...2...+...3...+...4...+...5...+...6...
====      .
====      .
```

## Master Record

The master record defines the characters that have special meanings in the repository's message and screen records. This record is mandatory. Use columns 1 through 63 for the master record. Also, it must be the first line in the repository other than blank lines or comments. All characters used on the master record must be a single-byte character whose hexadecimal value is between X'40' and X'FE', inclusive. When creating a file using XEDIT, make the master record in the following format:

```

===== * * * Top of File * * *
===== a bcd y z
          |...+....1....+....2....+....3....+....4....+....5....+....6...
=====
          .
=====
          .
=====
          .

```

- a** Defines a character that identifies a substitution variable in a message record, or indicates the beginning of a highlighted field in a screen record. The character cannot be a digit and must be followed by at least one blank character.

This character is called the *start variable* character for message definitions and the *start highlighting* character for screen definitions.

#### **bcd**

Defines the characters indicating the severity of numbered messages. For example, in the IBM-supplied mixed-case and uppercase American English message files, these are:

Code	Message Type
E	Error
I	Information
W	Warning
S	Severe Error (MPVM only).

At least one severity character must be specified. Duplicate characters and blank characters between message types are not allowed. The last message type specified must be followed by at least one blank character.

- y** Specifies the character that is used at the end of a line when it is necessary to continue the definition of a message or screen record on an additional line in the repository. This character must be followed by at least one blank. The character selected must not be a digit.

This character is called the *continuation* character.

- z** Indicates the position of a required blank in a message record or the end of a highlighted field in a screen record. The character specified must not be a digit. This character is optional.

This character is called the *end highlighting* character in screens, and the arbitrary blank character in messages.

Here is a sample master record:

```
$ EIW # ¢
```

## Time Stamp Records

There are two time stamp records. Time stamp records are used only in the *DVMIlangid* message files. When creating a file using XEDIT, make the time stamp records in the following format:

```

=====
=====
===== AMSTAMP ----- format -----
===== PMSTAMP NN ----- format -----
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
=====
=====
=====

```

### AMSTAMP

Defines the time stamp format for PVM time stamps issued before noon. The character string *AMSTAMP* must be in columns 1-7.

### PMSTAMP

Defines the time stamp format for PVM time stamps issued after noon. The character string *PMSTAMP* must be in columns 1-7.

### NN

Indicates what type of clock should be used for time stamps. One of two clock types must be specified on the PMSTAMP record. **NN** can be:

**12** Telling PVM to use a 12-hour clock.

**24** Telling PVM to use a 24-hour clock.

### format

Specifies the format of the time stamp. This field starts in column 9 for AMSTAMP and in the first nonblank character column following the **NN** field for PMSTAMP. A format is required on the AMSTAMP record and can be omitted on the PMSTAMP record. If a format is not specified on a PMSTAMP record, then the format specified on the AMSTAMP record is used. Leading, trailing, and excess intervening blanks are removed from the format, and the display length of the format, after removal of these blanks, must not exceed 40 characters. The format must not proceed beyond column 63.

The following keywords can be used to specify a time stamp format. The keywords must appear in uppercase when used in the time stamp record. At least one keyword must be specified.

Keyword	Use
<b>MO</b>	The month (2 digits)
<b>DD</b>	The day (2 digits)
<b>YY or YYYY</b>	The year. If YY is specified, the 2-digit year will be preceded by 19 (that is, the equivalent of coding 19YY).
<b>HH</b>	The hour (2 digits)
<b>MI</b>	The minute (2 digits)
<b>SS</b>	The second (2 digits)

**Note:** Each keyword can be used only once in the time stamp record. Except for keywords, all characters used in the time stamp format appear in PVM time stamps. Do not use character strings in time stamp records that exactly match a keyword. For example, if PMSTAMP is specified as follows:

PMSTAMP 12 HOUR: HH MINUTE: MI SECOND: SS



PVM reports that this record has an error in it because the *MI* keyword is used twice; once in *MINUTE:* and then again after *MINUTE:*. The following PMSTAMP record would not contain any errors:

PMSTAMP 12 Hour: HH Minute: MI Second: SS

because the *MI* keyword appears only once.

## Message and Screen Records

Message and screen records define three types of information:

**Numbered messages** messages that are displayed with headers that contain a three digit message identifier.

**Unnumbered messages** messages that are displayed without headers, usually in the message display areas of PVM display menus but occasionally embedded in numbered messages.

**Screen items** items that collectively define PVM display menus.

Each of these records in a repository file contains five fields. When creating a file using XEDIT, make every message and screen record in the following format:

```

=====
=====
===== NNNNVLLS ----- text -----
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
=====
=====
=====

```

### NNNN

is the message or screen identification number, in columns 1-4. Repository numbers:

**0000-1999** are used for numbered messages. Use a 4 digit message number in a message repository, but only the last three digits are displayed in the message header. (Message numbers 000 - 999 are for PVM and (DVMUSI), and messages 1000 - 1999 are for PVMG and MPVM. PVM and DVMUSI use the prefix DVM. PVMG and MPVM use the EFG prefix.)

**2000-6999** are not used.

**7000-7999** are used for unnumbered messages whose output length is limited to 80 bytes. These include the messages that must fit into the message display areas of PVM display menus. Each message in the 7000 to 7999 range must have a counterpart message in the 8000-8999 range.

**8000-8999** are used for unnumbered messages whose output length is limited to 48 bytes. These are messages that must fit into the display screens of PVM Release 3 and earlier. Each message in the 8000 to 8999 range must have a counterpart message in the 7000-7999 range. PVM always uses the 8000-8999 message when it must send a message to any remote node that is at an earlier release level with respect to that message.

**9000-9999** used for screen records. The 4 digit numbers for screen definitions are not displayed.

## **VV**

Is the message format, in columns 5-6. This field is for a message that can be in several versions. If a message has just one format, do not need anything, and the format field defaults to *01*. *00* cannot be used as a format number. This must be blank for screen records.

## **LL**

is the line number of the message, in columns 7-8. This field is used to display the text for a single message format on more than one line. Messages that spread across more than one line must have sequential, consecutive line numbers.

If a single format of a message has only one line, do not type a line number. The line number defaults to *01*. *00* cannot be used as a line number.

Both digits must be specified. The range of valid line numbers is *01* to *15*.

The line number must be blank for records defining either PVM unnumbered messages or screen records.

## **s**

is the severity code, in column 9. The severity codes defined in the IBM-supplied mixed-case and uppercase American English message files are:

<i>Code</i>	<i>Message Type</i>
E	Error
I	Information
W	Warning
S	Severe Error (MPVM only).

## **text**

is the message text, starting in column 11. PVM removes any leading, trailing, and excess intervening blanks in the text field. Specify characters up to column 63; use the continuation character defined in the master record to continue on the next line of the message file. To make a continuation line, use the continuation character as the last nonblank character in the text field. The continuation line must be the next record in the repository that is neither a comment nor a blank record. Columns 1-10 must be blank on the continuation record.

The continuation character must not be used as the first nonblank character in the text.

Also, substitution variables can be used in text definitions; see “Substitution Variables in Messages” on page 139 for more information.

The display lengths of messages, after substitution for variables, is limited as follows:

**Messages 0001-1999** Must not exceed 140 characters including the message header. There are some exceptions to this rule listed in “Special Considerations for Message and Screen Records” on page 139.

**Messages 7000-7999** Must not exceed 80 characters.

**Messages 8000-8999** Must not exceed 48 characters.

The following table Table 14 on page 139 summarizes the contents of message and screen definition records.

<i>Table 14. Message and Screen Definition Record Summary</i>			
Columns	Numbered Messages	Unnumbered Messages	Screen Items
1-4	Identification number (0000-1999)	Identification number (7000-8999)	Identification number (9000-9999)
5-6	Version number	Version number	Blank
7-8	Line number	Blank	Blank
9	Severity code character	Blank	Blank
10	Blank	Blank	Blank
11-63	Message Text	Message Text	Screen Text
<b>Note:</b> Columns 1-10 must be blank in continuation lines.			

## Substitution Variables in Messages

PVM uses substitution variables in message repository files for information it does not know until execution time. Substitution variables are replaced with actual values by the PVM message handler at execution time. Modules that call the message handler to construct a message pass an argument list that contains the values for the substitution variables. Substitution variables cannot be used in screen definition records.

A substitution variable is a five-character string in the following format:

AMMNN

**A** Is the start variable character defined in the repository's master record.

**MM**

Identifies the parameter that is to replace this particular substitution variable. For example, *02* identifies the second parameter in the argument list. The valid range of values for MM is *01* to *51*. Both digits must be specified. Parameter positions cannot be skipped (for example, if position *03* is used, then there must also be positions *01* and *02*).

**NN**

Indicates the maximum length in characters of the replacement string. The valid range of values for NN is *01* to *99*.

For example, if *\$* is defined as the start variable character, then *\$0205* defines a variable that is to be replaced by the second parameter in the argument list to a maximum length of five characters.

If the text of a message is changed, or a message is translated into another language, all of the substitution variables used in the original message must be present in the new message. The substitution variables themselves cannot be changed. They can, however, be placed in any order within the text.

## Special Considerations for Message and Screen Records

- Highlighting can enhance the appearance and readability of PVM display screens. To use highlighting, define an end highlighting character in the master record.

To define a highlighted field, use the start highlighting character followed by the field of characters that is to be highlighted, followed by the end highlighting character. The start highlighting character does not count toward the overall

length limit of the screen item if it is the first character in a text definition. The end highlighting character does not count toward the overall length limit of the screen item if it is the last character in a text definition.

Highlighting works only for screen records 9001, 9074, and 9075. Highlighting characters in any other screen record are converted to blanks.

For example, suppose \$ and ¢ are defined as the start highlighting and end highlighting characters. Specifying \$ENTER¢ in the text definition highlights the word *ENTER* in the screen display.

Certain lines on the PVM Selection and Initial Menus are highlighted without regard to highlighting characters in the screen item definitions. When this occurs, any highlighting characters are converted to blanks.

Highlighting cannot be used in message records.

- Some PVM messages have special restrictions. These messages and their restrictions are listed in the following table.

Message	Restrictions
0000	<ul style="list-style-type: none"> <li>• Must be defined with only one version and as single-line messages.</li> <li>• The text definition must contain exactly one substitution variable.</li> <li>• The length of the text of this message after variable substitution cannot exceed 79 characters.</li> </ul>
0010, 1010, 0050, 1050, 0068, 1068	<ul style="list-style-type: none"> <li>• Must be defined with only one version and as single-line messages.</li> <li>• Do not specify any text definition on the message record for these messages.</li> </ul>
0127, 1127	<ul style="list-style-type: none"> <li>• Must be defined with only one version and as single-line messages.</li> <li>• The text definition for this message must contain one substitution variable. Additional variables and characters are not allowed.</li> </ul>

- When creating a new message repository, there are limits to the display lengths of screen records. The display length limits for screen records are:

	Display Length Limit				
	4	5	27	67	79
<b>Screen Record Identifier</b>	9010-9017	9018-9020	9002-9007, 9073, 9076	9008	9000-9001, 9009, 9074-9075, 9100-9114

- Screen records 9101-9114 define the text of the PVM I/O error recovery menu. Each screen record defines a single line of text. The text consists of a single paragraph beginning with record 9101. The records for unneeded display lines must be defined with blank text fields. The first record with a blank text field indicates the end of the paragraph.
- Blanks between words in screen record 9009 are not removed by PVM and are left as defined. Leading and trailing blanks are removed.
- Screen record 9021 defines the five characters that are in the PVM Selection Menu to indicate node types. The text field of this record must contain exactly five nonblank, single-byte characters. Duplicate characters are not allowed.

- Screen record 9022 defines a padding string that is used in the PVM Selection Menu to enhance the readability of certain lines. The text field is limited to a maximum of two, nonblank characters. PVM English language repositories have → defined as the padding string. The padding string is used in screen records 9002-9008.

If only one character is specified as a padding string, it is treated as though the same character is specified twice. For example, = is treated as if == had been specified.

- Records in the range 7510-7519 are reserved for use in the MPVM Selection Menu (which is built at MPVM initialization). Records 8510-8519 are reserved as counterpart messages.

The PVM compiler doesn't check to see that there are no variables in these records. It is the programmer's responsibility to ensure that no variables in these messages are used for screen lines by MPVM.

## Control Lines

Control line records are used in message repositories to separate messages for different versions and releases of PVM. Messages must remain in the correct control section of the file. The control line sections are used for message level synchronization in the PVM network. For example, any messages in the section below

CONTROL CL3

are for PVM Version 2.

Messages that may be added for service are added after the

CONTROL SERVICE

record.

## Message Repository Samples

Figure 28 illustrates the records defining three sample messages.

```
|...+...1...+...2...+...3...+...4...+...5...+...6...
$ EIW # ¢
.
.
.
09993201I Port $0102 is a $0204, Model $0302, currently in #
      session.
09993202I It is a color terminal with extended highlighting
09993203I and programmable      symbols. The last status rece#
      ived
09993204I from this terminal was $0404.

7006      $0108 $0208 link is already in use.
8006      $0108 $0208 LINK IS ALREADY IN USE.
```

Figure 28. Example Message Records

When it is displayed, sample message 999I could look like this:

```

DVMQRY999I Port 20 is a 3279, Model 3, currently in session.
DVMQRY999I It is a color terminal with extended highlighting
DVMQRY999I and programmable symbols. The last status received
DVMQRY999I from this terminal was 4050.

```

Figure 29, and Figure 30, are samples of the screen definition records for the PVM Selection, Initial, and I/O Error Recovery Menus.

```

|...+....1....+....2....+....3....+....4....+....5....+....6...
9000  VM/PASS-THROUGH FACILITY
9001  You Can Select a Node With the Cursor and Press #
      $ENTER
9002  Destination
9003  Language
9004  End Session
9005  Verify
9006  Port
9007  Route
9008  Your Identification
9009  PF8= Scroll          CLEAR key = Top Sc#
      reen                PA1= Exit
9010  PF1=
9011  PF2=
9012  PF3=
9013  PF4=
9014  PF5=
9015  PF6=
9016  PF7=
9017  PF9=
9018  PF10=
9019  PF11=
9020  PF12=

* 9021 = characters displayed beside certain node names:
* 1st - the local PVM system.
* 2nd - directly linked PVM nodes.
* 3rd - directly linked non-PVM systems.
* 4th - directly linked ROCF nodes.
* 5th - name of a line group.

9021  L N S R G

9022  ->
9073  Local Node
9074  Press $ENTER for a Session With the Following #
      Options
9075  Press $CLEAR to View the Selection Screen
9076  Address

```

Figure 29. Initial and Selection Menu Definitions

```

|...+...1...+...2...+...3...+...4...+...5...+...6...
9100    PROGRAM ERROR RECOVERY
9101    The program you are using through PVM has #
        transmitted
9102    control information that is not compatible with your
9103    display unit. At this time you can attempt #
        to recover
9104    from the error by pressing keys that are appropriate
9105    to the program you are using, or you can end your
9106    PVM session. The ENTER key will end your PVM #
        session,
9107    unless you specified a PF or PA key for that purpose.
9108
9109
9110
9111
9112
9113
9114

```

*Figure 30. I/O Error Recovery Menu Definitions*





## Chapter 13. Facilities for Automating PVM and PVMG Operations

### GUIP

Application programming facilities available to the system programmer or system operator are:

- MPVM macros and commands (see the *VM/Pass-Through Facility: Programming Reference* for this information).
- The command response interface (CRI) discussed in this chapter.
- A programmable operator (PVMPROP) message interface discussed in this chapter.
- Application-to-application communication using IUCV protocols (see Chapter 15, “Application-to-Application Overview” on page 159 for this information).

Using these facilities to write application programs can increase the efficiency of system and network operations.

The programmable operator facility allows developing applications for the remote automated operation of PVM systems. Using the programmable operator message interface, applications can perform tasks that do not require an on-site operator.

A sample exec, PVMSNAP EXEC, is provided to show how an application uses PVMPROP/CRI. See “Sample Application” on page 151 for more information about this example application.

---

### PVMPROP - A Programmable Operator Interface for PVM and PVMG

The PVMPROP configuration file record identifies a user ID that receives copies of error messages issued by PVM or PVMG. A PVMPROP user ID can be used to run an application program that acts like a PVM or PVMG operator. An application program can process PVM or PVMG error messages and monitor PVM for problem indicators and then attempt to correct a problem or notify the real operator of a problem.

PVMPROP can also use the Command Response Interface (CRI), described in the next section. Command prefixes are discussed later in this chapter.

Error messages sent to a PVMPROP user ID by PVM or PVMG are sent using the CP SMSG command using the PVMPROP/CRI message protocol.

For example, PVM issues message 137 when there are no free buffers available to a PVM task. This situation can sometimes cause severe performance degradation of the PVM system. A PVMPROP user ID receives a copy of message 137. An application program running on that user ID could be programmed to notify the system programmer of the situation.

In another scenario, a PVM link goes down, and PVM issues a message. One copy goes to the AUDIT file, and the other is sent to the PVMPROP user ID. An installation written application intercepts the message and sends a command using

the *T.* prefix of the CRI to PVM telling PVM to START the link that is down. If successful, the application ends. If the START command is unsuccessful, as determined from the command response, the application could send a message to the system operator, along with specific instructions on what corrective action to take.

The PVMDEST FILE on the PVM system minidisk can be used to find out which messages PVM or PVMG sends to a PVMPROP user ID. Each line in PVMDEST FILE lists a PVM message number and destinations for that message. The character string *PROP* on a line in the file indicates that the message is sent to a PVMPROP user ID. See “Changing Message Destinations” on page 130 for more information on this topic.

---

## The Command Response Interface

Applications can use the PVM *T.* command prefix to tell PVM or PVMG to send command responses in a language- independent format that is easily understood by applications. (For information about using command prefixes, see “Using Command Prefixes” on page 149.) PVM uses the CP SMSG command to send a command response when the *T.* prefix is used.

PVM and PVMG encode the command responses to *T.* prefixes. This encoded message is language-independent and can be easily processed by an application program. See “PVMPROP/CRI Message Protocol” for a description of the information in an encoded message.

PVM sends message number 0001 to indicate the end of the command response. This makes it easier for an application program to determine when the responses for a command are finished.

The *M.* and *S.* prefixes can be used:

- To find out which messages PVM issues in response to a command
- When an application needs to route the natural language message to a human operator.

For example, an application could query PVM or PVMG periodically for network status. If the application uses the *T.* prefix, PVM or PVMG responds in a *tokenized* form that is easily interpreted by the application. If the application uses the *M.* or *S.* prefixes, PVM or PVMG responds in natural language. These responses can be routed to a human operator.

---

## PVMPROP/CRI Message Protocol

PVM and PVMG use the PVMPROP/CRI message protocol format when:

- The command response interface is used
- PVM or PVMG error messages are to be copied to a PVMPROP user ID.

The message format consists of a 30 byte header, which is followed by an encoded message, as follows:

PVM↓↓nnnn↓001↓nodename↓yyyyyy↓message

(Note that each ↓ symbol represents a blank space).

## Message Format

The PVMPROP/CRI message format consists of a 30-byte header (containing five parts) followed by an encoded message. Each part is separated by a blank, making it easier for application programs to process the response. The table that follows describes the header, and the message text that follows.

Table 15. Parts of the PVMPROP/CRI Message Format		
Part	Length	Notes
<b>PVM/PVMG</b>	4	A 4 character constant; there is a blank character at the end of the character string. This identifies whether the message is from PVM or PVMG.
<i>nnnn</i>	4	This is the PVM or PVMG internal message number
<b>00//</b>	4	The message line number. When the message is encoded, as in PVMPROP messages and CRI responses to a <i>T.</i> prefix, the value will always be 0001.
<i>nodeid</i>	8	Contains the node ID of the node where the message originated. If the node ID is less than 8 characters long, the string is filled with blanks on the right.
<i>identifier</i>	6	<p><b>For command responses:</b> the identifier from the original command prefix. If the identifier is less than six characters long, the string is filled with blanks on the right.</p> <p><b>For error messages copied to a PVMPROP user ID, or for responses to an <i>S.</i> prefix, or for <i>T.</i> and <i>M.</i> prefixes without using the identifier:</b> a time stamp in the form <i>hhmmss</i>, where <i>hh</i> is hours, <i>mm</i> is minutes, and <i>ss</i> is seconds.</p>
<i>message text</i>	varies	<p><b>For messages sent to a PVMPROP user ID or responses to a <i>T.</i> prefix:</b> contains an encoded message.</p> <p><b>For responses to an <i>M.</i> or <i>S.</i> prefix:</b> contains the actual message text.</p>

## Deciphering Encoded Messages

PVM and PVMG send messages to PVMPROP user IDs and *T.* prefix responses in an encoded form using the CP SMSG command. An encoded message contains all of the information needed to know everything the message contains, without the natural language text that a human operator would require. Encoded messages do not contain message information that could change depending on the language that PVM or PVMG are using. This allows application programs that receive encoded messages to be language-independent. Table 16 describes the format of PVM and PVMG encoded messages.

The message will not be encoded if the *S.* or *M.* prefixes are used. *T.* prefixes will cause messages to be encoded unless:

- A CP command response is being echoed by PVM
- The encoded message contains a message from a PVM node not running PVM Version 2 or Version 1.4. The portion of the message passed from the other PVM node will not be encoded.

PVM and PVMG modify an encoded message before sending it so that the encoded message does not contain the *break* character (X'15'). To reverse the modification, an application program should:

- Convert each X'FF' in the encoded message to X'15'.
- Divide the message number by 2, including the embedded message number, if present.

Table 16. Encoded Message Format	
Bytes	Meaning
0-1	The PVM or PVMG message number. For PVM or PVMG messages that are issued without a numbered identifier, the message number has a value in the range 7000 through 7999
2-3	The message version number. Some PVM and PVMG messages are broken down into versions in the repository files. This is to allow greater flexibility for different translations. Information sent in an encoded message can vary depending on the message version.
4-6	A three-character module suffix of the PVM or PVMG module issuing the message. These bytes are not used if the message is an unnumbered PVM or PVMG message.
7	The number of substitution variables in the remaining bytes.
8-end	<p>Substitution variables for this message. The format of these variables is:</p> <p><b>Byte 1</b> The length of the substitution variable.</p> <p><b>Byte 2</b> The first byte of the substitution variable.</p> <p><b>Note:</b> There are two special cases for the last variable of an encoded message:</p> <ul style="list-style-type: none"> <li>• If the length byte is 127 (X'7F'), it indicates that the variable is a 16-byte time stamp in the form "MM/DD/YHH:MM:SS".</li> <li>• If the length byte is 128 (X'80') or higher, it indicates a PVM or PVMG unnumbered message in encoded form, the length of which is the contents of the length byte minus 128.</li> </ul>

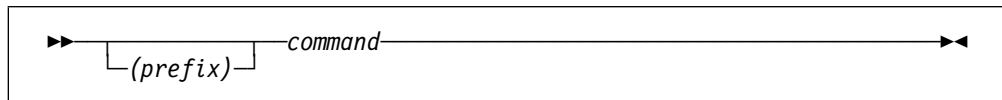
To find out the meanings of the substitution variables used in PVM messages:

- Print and read *Section Two* of the READTHIS FIRST file located on 5684100E's 2C2 minidisk for PVM 2.1.1
- Look at the on-line message description, if the message has an identifier. Look at the *VM/Pass-Through Facility: Messages*, SC24-5648 if the message has no identifier.
- Look at a copy of the source file for the PVM message repository. The file name of the message repository is DVM*langid* REPOS, where *langid* is a language ID. (For example, DVMAMENG REPOS is the source file for the American English message repository.) The message repository can be found on the PVM source staging minidisk (5684100E's 2B1 for PVM 2.1.1).

For a description of the parts of a PVM message repository, including message numbers, message versions, and substitution variables, see "Creating and Using a Message Repository" on page 132.

## Using Command Prefixes

PVM commands can be preceded by a prefix string as described earlier in this chapter. This prefix tells PVM how to respond to a command. The format for command prefixes is:



### Where:

#### *prefix*

Is one of the following:

Prefix	Meaning
<b>L.langid</b>	<p>Tells PVM or PVMG to send any command responses using the language identified by <i>langid</i>. This prefix is intended for use by a general user of PVM or PVMG who needs to see PVM or PVMG command responses in a language other than the default. Use the QUERY command with the LANGS option to display the language IDs of the languages that are available on PVM. For example, issuing:</p> <pre>sm pvm (l.uceng) q lang</pre> <p>will produce a response similar to:</p> <pre>PVM AT RMT_PVM1 IS ENABLED FOR LANGUAGE AMENG PVM AT RMT_PVM1 IS ENABLED FOR LANGUAGE UCENG PVM AT RMT_PVM1 IS ENABLED FOR LANGUAGE KANJI OUR LOCAL DEFAULT LANGUAGE IS AMENG OUR BACKUP NETWORK LANGUAGE IS UCENG</pre>
<b>T.identifier</b>	<p>Tells PVM or PVMG to send the command response in language-independent form using the CP SMSG command. This prefix is intended for use by PVM or PVMG application programs. The <i>identifier</i> is a character string that is returned in the command response. The <i>identifier</i> can be up to 6 characters long. The <b>T.</b> is optional and can be omitted. For example, (<i>xxx</i>) is the equivalent of (T.<i>xxx</i>), since T. is the default.</p> <p>Alternately the <i>identifier</i> can be omitted. In this case, a time stamp is returned in the command response in place of an identifier.</p> <p>For example, (T.) will return a response in an encoded format.</p> <p>See “The Command Response Interface” on page 146 for more information.</p>

Prefix	Meaning
<b>M.identifier</b>	<p>Tells PVM or PVMG to send the command response in the default natural language, prefixed with the 30-byte CRI header (described in “Message Format” on page 147), using the MSGNOH (or MSG) command. The <i>identifier</i> is a character string that is returned in the command response. The <i>identifier</i> can be up to 6 characters long. This tag is used primarily for testing PVM or PVMG responses for system applications using the command response interface. It can be used by an application when routing a natural language message to a human operator. See “The Command Response Interface” on page 146 for more information.</p> <p>For example, issuing:</p> <pre>sm pvm (m.tag) q lang</pre> <p>will produce a response similar to:</p> <pre>PVM 0145 0001 RMTPVM1 TAG PVM at RMTPVM1 is enabled for language AMENG PVM 0145 0001 RMTPVM1 TAG PVM at RMTPVM1 is enabled for language UCENG PVM 0145 0001 RMTPVM1 TAG PVM at RMTPVM1 is enabled for language KANJI PVM 0146 0001 RMTPVM1 TAG Our local default language is AMENG PVM 0146 0001 RMTPVM1 TAG Our backup network language is UCENG PVM 0001 0001 RMTPVM1 TAG Your Q command is complete</pre>
<b>S.langid</b>	<p>Tells PVM or PVMG to send the command response, prefixed with the 30-byte CRI header (described in “Message Format” on page 147), in the language identified by <i>langid</i>, using the MSGNOH (or MSG) command. This tag is used primarily for testing PVM responses for system applications using the command response interface. It can be used by an application when routing a natural language message to a human operator. See “The Command Response Interface” on page 146 for more information.</p> <p>For example, issuing:</p> <pre>sm pvm (s.uceng) q lang</pre> <p>will produce an upper case response similar to:</p> <pre>PVM 0145 0001 RMTPVM1 124414 PVM AT RMTPVM1 IS ENABLED FOR LANGUAGE AMENG PVM 0145 0001 RMTPVM1 124414 PVM AT RMTPVM1 IS ENABLED FOR LANGUAGE UCENG PVM 0145 0001 RMTPVM1 124414 PVM AT RMTPVM1 IS ENABLED FOR LANGUAGE KANJI PVM 0146 0001 RMTPVM1 124414 OUR LOCAL DEFAULT LANGUAGE IS AMENG PVM 0146 0001 RMTPVM1 124414 OUR BACKUP NETWORK LANGUAGE IS UCENG PVM 0001 0001 RMTPVM1 124414 YOUR Q COMMAND IS COMPLETE</pre>

*command*  
is a valid PVM or PVMG command.

For example, entering:

```
smsg pvm (l.kanji) query langs
```

causes PVM to send the response in Japanese (provided that the local PVM or PVMG is enabled for Japanese).

### Notes:

1. The parentheses shown in the format box are a required part of the command prefix.
2. A command prefix cannot be used:
  - With the PASSTHRU command
  - With the MPVM command

- With PVM system programmer commands (PVMBLD, PVMCMP, and PVMGEN)
  - When logged on the PVM or PVMG user ID.
3. To find out if the language is available on a PVM or PVMG system, use the QUERY LANGS command. For example, the PVM response to the command:  

```
smmsg pvm query langs
```

reports the languages that are available to the local PVM system named pvm.
  4. Some languages require special display stations. PVM and PVMG use the requested language without regard to display station requirements.
  5. The command prefix is ignored when it is used by the PVM or PVMG operator while logged on the PVM or PVMG user ID.
  6. Messages sent to other destinations as the result of a command are not affected by the command prefix.

---

### Sample Application

The PVMSNAP EXEC file on the CMS system extension minidisk is a sample application of the command response interface and PVMPROP. PVMSNAP EXEC traps and processes SMSG messages sent by PVM or PVMG. Although the application is written using the REXX language, it is not an endorsement of one tool over another. Write applications using any appropriate tools.

**GUII end**





## Chapter 14. TCP/IP Line Driver Overview

VM/Pass-Through Facility (PVM) Transmission Control Protocol/Internet Protocol (TCP/IP) provides a PVM peer-to-peer line driver using a TCP/IP network as the transport medium. This line driver takes advantage of the high speed capabilities of an internet network, providing VM users with fast, reliable access to applications available from a remote PVM node.

A PVM TCP/IP line driver communicates with another PVM TCP/IP line driver at a different VM node in a peer-to-peer environment. Neither side acts as a client nor a server. Once a connection is established between two nodes, this is not a problem. However, to establish the connection, there needs to be a client and server relationship between the two nodes when PVM is initialized. The PVM TCP/IP redirector task solves this problem by acting as a server waiting for a connect request from the other node. The PVM TCP/IP line driver task from the other node acts as the client by sending the connect request. The redirector task at the receiving node intercepts the connect request, verifies that the request is valid, and passes the request to the local TCP/IP line driver task, establishing the local node's receive path. When both nodes issue and complete connect requests with the opposite node, communications are established between the nodes.

Figure 31 graphically illustrates this client and server relationship between two nodes during PVM initialization.

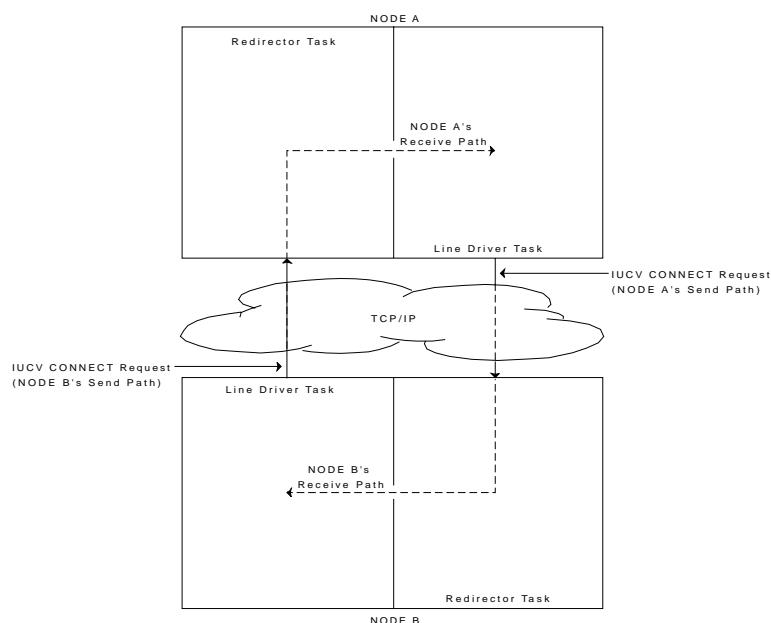


Figure 31. PVM TCP/IP Client-Server Relationship

---

## The PVM TCP/IP Redirector Task

### NOT-PI

The redirector task's function is to listen for a connect request to the local node over a TCP/IP network. The redirector task takes the information from the requesting node and checks to see if there is a link defined for the requesting node. If not, the redirector task sends a message back to the requester indicating that there is no link defined and waits for another request to come in. If there is a link defined, the redirector task informs the TCP/IP line driver task that a valid node is requesting a connection with this node. The redirector task makes the interface available to the TCP/IP line driver task and then waits for another request to come in.

### Redirector Task Error Processing

If the redirector task encounters an error, based on the type of error, one of the following error procedures is initiated:

- If an IUCV SEVER occurs, then the redirector task waits 5 minutes and attempts to re-establish communications with VM TCP/IP.
- If VM TCP/IP gives a return code of -1, and if it is during a CONNECT socket request, then the new socket is closed and the redirector task issues another LISTEN socket request to VM TCP/IP. Otherwise, the redirector task ends.
- If the redirector encounters an IUCV error, the redirector task ends.
- If an End-of-File indication is received on a READ, the new socket is closed and another LISTEN socket request is issued to VM TCP/IP.

In all cases, appropriate error messages are issued.

---

## The PVM TCP/IP Line Driver Task

The TCP/IP line driver task issues a TCP/IP SOCKET CONNECT to the remote PVM node using the defined IP address and port number of the remote node. This SOCKET will become the local node's send path. The TCP/IP line driver then waits for a GIVESOCKET from the local redirector task, created from the remote node's CONNECT request. This SOCKET will become the local node's receive path. Once the SOCKETS are established, the communication link is connected and normal PVM packets can be exchanged.

When a TLSLABEL has been defined, a secure connection with the TLS/SSL server is established during socket connect processing.

### PVM TCP/IP Line Driver Error Processing

If the VM TCP/IP line driver task encounters an error, based upon the type of error, one of the following error procedures is initiated:

- If an IUCV SEVER occurs during initial connect processing, the TCP/IP line driver task waits 5 minutes, then attempts to re-establish communications with VM TCP/IP. Otherwise, the TCP/IP line driver task ends and the network manager restarts the line driver.

- If VM TCP/IP gives a return code of -1, the line driver task ends. It will be restarted again by the network manager only if communications with the remote node had been established.
- If the TCP/IP line driver receives an End-of-File indication on a READ, and the TCP/IP line driver has established communications with the remote node, the line driver task ends and is restarted by the network manager. Otherwise, the CONNECT socket request is retried in 5 minutes
- If an IUCV error is encountered, the line driver task ends. It will be restarted by the network manager only if communications with the remote node had been established.
- If a CONNECT socket request completes but a GIVESOCKET does not arrive from the redirector task from the remote node's CONNECT request within 5 minutes, the socket is closed and a CONNECT socket request is issued again in 5 minutes.
- If a secure connectivity mismatch is encountered for which one side is configured with a TLSLABEL and the other side is not, an error message is issued.

In all cases, appropriate error messages are issued.

**NOT-PI end**

---

## A Word About Sockets

Both the PVM TCP/IP redirector task and the PVM TCP/IP line driver use TCP/IP sockets to interface with TCP/IP. Once these sockets are established, a user can communicate across networks with remote applications. From the line driver and the redirector's perspective, a socket is a resource allocated by the TCP/IP virtual machine. The TCP/IP socket interface provides applications with a network interface that hides the details of the physical network.

For more information about TCP/IP sockets, refer to the *z/VM: TCP/IP Programmer's Reference*.

---

## Tracing Considerations

PVM TCP/IP line driver tracing support is provided similar to current line driver tracing support. I/O tracing for the redirector task is also supported.

---

## Configuration File Example

To show how to specify the new configuration file records to produce a TCP/IP link between two PVM nodes, assume that NODEA will use TCP/IP port number 821 and has an IP address of 9.130.25.200. NODEB will use TCP/IP port number 823 and has an IP address of 9.82.1.101. The virtual machine ID running TCP/IP is TCPIP.

Figure 32 on page 156 shows an example of how the configuration file for node NODEA would be specified:

```
*****
* Configuration file for node NODEA - NODEA CONFIG      *
*****
LANG AMENG
:
LINK * NODEB TCPIP IP 9.82 1.101 PORT 823
:
TCP 821 TCPIP START
:
```

Figure 32. Sample PVM System Configuration File for NODEA -- NODEA CONFIG

Figure 33 shows an example of how the configuration file for node NODEB would be specified:

```
*****
* Configuration file for node NODEB - NODEB CONFIG      *
*****
LANG AMENG
:
LINK * NODEA TCPIP IP 9.130 25.200 PORT 821
:
TCP 823 TCPIP START
:
```

Figure 33. Sample PVM System Configuration File for NODEB -- NODEB CONFIG

Figure 34 shows an example of how the configuration file for node NODEA would be specified when a secure link is desired:

```
*****
* Configuration file for node NODEA when secure - NODEA CONFIG      *
*****
LANG AMENG
:
LINK * NODEB TCPIP IP 9.82 1.101 PORT 823 TLSLABEL MYCERT
:
TCP 821 TCPIP START
:
```

Figure 34. Sample PVM System Configuration File for NODEA as a secure link -- NODEA CONFIG

Figure 35 shows an example of how the configuration file for node NODEB would be specified when a secure link is desired:

```
*****
* Configuration file for node NODEB when secure - NODEB CONFIG      *
*****
LANG AMENG
:
LINK * NODEA TCPIP IP 9.130 25.200 PORT 821 TLSLABEL MYCERT
:
TCP 823 TCPIP START
:
```

Figure 35. Sample PVM System Configuration File for NODEB as a secure link -- NODEB CONFIG

If the two links will not connect, some possible reasons are:

- The redirector task is not started on one or both nodes

- The IP addresses are not correct for the opposite node
- The TCP/IP port number for the redirector task is already in use
- The IP address value was specified using the wrong format for PVM. The correct format for PVM is `www.xxx yyy.zzz` (a blank instead of a period is required by PVM between the second and third address fields).
- When a secure link is desired and the `TLSLABEL` parameter has not been defined on the link statement for both nodes.
- An error number of 1012 on an `IOCTL` socket request may indicate the SSL server is unavailable or the SSL server is unable to access the certificate database.
- The label name used on the `TLSLABEL` parameter does not point to a certificate which has been defined to the SSL server.

**GUPI end**



## Chapter 15. Application-to-Application Overview

### GUI

The PVM PCCF task allows a CMS virtual machine to request one or more PVM sessions with itself, another virtual machine on the local system, or another CMS virtual machine on any PVM node in the network that contains this facility. Intermediate nodes need not have PCCF support.

To use the PVM PCCF facilities, the application programmer is expected to be familiar with VM IUCV protocols. IUCV information is available in:

- *z/VM: CMS Application Development Guide for Assembler* (SC24-6257)
- *z/VM: CP Programming Services* (SC23-0370)
- *z/VM: CMS Application Program Development Reference* (SC23-0402).

After the session is established, the virtual machines can communicate asynchronously using the IUCV SEND function with the one-way option, or synchronously using the IUCV SEND function with reply option.

The data may be of any format and may be character or binary. The data length may be up to 32768 bytes for the SEND message or 32768 bytes for the REPLY message. Any combination of message classes can be used.

### PCCF Data Flow

A PVM PCCF session is established by a virtual machine issuing an IUCV CONNECT request to the PVM virtual machine. The PVM target *nodename* is specified in the first 8 bytes of the user data field (IPUSER) of the IUCV CONNECT parameter list, and the *vmid* of the target virtual machine is specified in the second 8 bytes of that field. User virtual machines must have directory authorizations for IUCV communication with the PVM virtual machine, unless the PVM virtual machine has the IUCV ALLOW directory statement. PVM will forward the CONNECT request to the target PVM machine, which will then issue an IUCV CONNECT to the target *vmid*. The target *vmid* is passed the origin PVM *nodeid* and the origin *vmid* in the user data field of the target PVM's IUCV CONNECT parameter list. If the CONNECT is accepted, the session is completed by PVM, and the original CONNECT is then accepted by the origin PVM. At this point, either user virtual machine can issue IUCV SENDs to pass data back and forth. The CONNECT and ACCEPT options (PRTY, PRMDATA, QUIESCE) are passed from one side to the other.

If the path cannot be completed, a SEVER is done by the origin PVM system.

If both user virtual machines issue a CONNECT for the opposite end, PVM will attempt to start two sessions, and each virtual machine will receive a pending CONNECT request. The virtual machines must decide which path to SEVER and which to ACCEPT.

### Data Transmission

After the session is established, data can be sent between VM applications using an IUCV SEND. If the SEND is one-way, the sending machine will receive the message complete interrupt as soon as the local PVM receives the data, not when the target virtual machine receives the data. The data is then routed to the target node. The sending virtual machine can immediately issue another SEND, thereby causing a stream of data to flow. Because of this, several pieces of data can be transmitted at the same time for the same session. PVM uses end-to-end pacing to limit flooding in this situation. If the number of outstanding transactions reaches a specified limit, the next SEND will not be received by the local PVM virtual machine until the number of outstanding transactions falls below a specified level.

If the SEND requires a REPLY, PVM will send the data to the target node and issue a SEND with REPLY at that node. The reply data is routed back to the sender, and PVM does a REPLY. In this case, the sender will not get the message complete interrupt until the reply is returned.

Any target message class may be used in the IUCV SEND parameter list, and PVM will reflect the same at the target site.

The message size limitations are 32768 bytes for the SEND message and 32768 bytes for the REPLY message, provided both the origin and target PVM nodes are running in VM environments.

The IUCV QUIESCE and RESUME functions will be reflected through to the opposite end. If a IUCV QUIESCE is issued from side A to B, but side B has issued a SEND before the IUCV QUIESCE is reflected, that IUCV SEND will be held at PVM on side A until side A issues an IUCV RESUME.

Although CP queues IUCV PRIORITY messages ahead of non-PRIORITY messages, message transmission within PVM is based upon a First-In-First-Out (FIFO) procedure.

### Session Termination

PVM handles both normal and abnormal terminations. These processes are described in this section.

#### Normal Termination

Either virtual machine may terminate the PVM session by issuing an IUCV SEVER to the path. PVM will then terminate the session and reflect an IUCV SEVER to the other side.

To ensure that any one-way IUCV SENDs that may still be in transit are processed at the opposite end on receipt of a IUCV SEVER from the local application, PVM will transmit a null data packet to the opposite end with a pacing acknowledgement request. It will then wait for all outstanding acknowledgement requests to be satisfied before terminating. At the opposite end, on receipt of a null data packet with a pacing request, the packet will not be processed until all outstanding SENDs have completed. If this side of the path is IUCV quiesced, any queued IUCV SENDs will be purged.



### Abnormal Termination

If the session terminates for any reason other than the normal termination, PVM will reflect a SEVER to both sides. When the virtual machine receives an IUCV SEVER interrupt, it must also issue an IUCV SEVER for that path in order to reuse that *pathid*.

### PCCF Application-to-Application Restrictions

PCCF application-to-application restrictions are as follows:

- Maximum data length on an IUCV SEND/REPLY is 32768 bytes.
- CP IUCV PRIORITY cannot be used unless authorized by appropriate CP Directory statements.
- One-way SENDs are not synchronized, and the *message complete* interrupt does **not** mean that the opposite end has received the data.

---

### Application-to-Application Communication Sample Program

The following illustrates the sequence of functions invoked when two virtual machines communicate, using IUCV protocols, through PVM. In this illustration, a virtual machine with vmid USERX is communicating with a virtual machine with vmid USERY. USERX is located at a different PVM node than USERY.

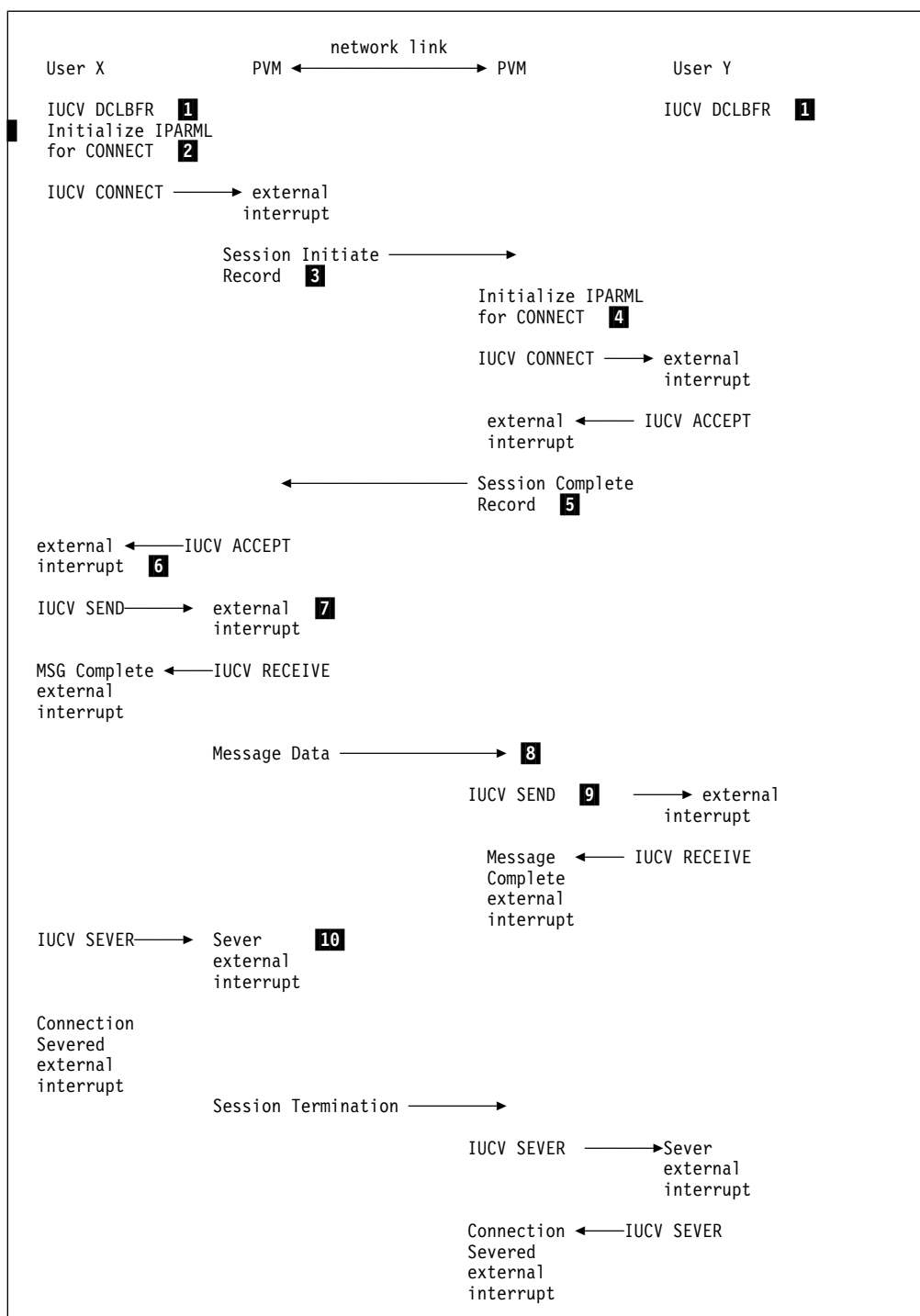


Figure 36. Application-to-Application Communication Sample Program

The IUCV responses that the User X (origin) and User Y (target) virtual machines receive as a result of invoking IUCV commands, reflect the result of communicating with PVM and not with each other. For instance, if the the User X virtual machine connects to the local PVM virtual machine, the return code in the IPARML IPRCODE field is zero regardless of whether the User Y virtual machine is logged on. If the User Y virtual machine is not logged on, the User X virtual machine is passed an IUCV SEVER.

The flow of the Figure 36 is described in greater detail below.

- **1** Both user virtual machines must independently invoke the DECLARE BUFFER function.
- **2** The target nodeid and target vmid are moved into the 16-byte IPUSER field of the IPARML parameter list. The origin PVM vmid is moved into the IPV MID field.
- **3** The origin PVM node has recognized that the USERX virtual machine wants to communicate with the USERY virtual machine at a remote PVM node. It prepares a Session Initiate record and forwards that record to the remote PVM node. The remote PVM node may be on an adjoining link, or it may be several links away. Intermediate PVM nodes do not have to have an active PCCF task.
- **4** The target PVM node has received the Session Initiate record and it prepares an IUCV CONNECT request by initializing the user data field of the IPARML parameter list with the origin PVM nodeid and the origin virtual machine vmid, USERX.

The IPV MID is set to the target virtual machine vmid, USERY. The IUCV CONNECT request is then invoked.

- **5** The target user virtual machine signals acceptance of the IUCV CONNECT request by invoking an IUCV ACCEPT to the PVM virtual machine. PVM then prepares a Session Complete record and forwards it to the origin PVM node.
- **6** The origin PVM node has received the Session Complete record and informs the origin virtual machine, USERX, by invoking an IUCV ACCEPT.
- **7** The origin and target virtual machines may now send data to each other.

The target virtual machine issues a one-way IUCV SEND to PVM. PVM performs an IUCV RECEIVE which results in an IUCV Message Complete external interrupt to the origin virtual machine. Because this is a one-way communication the origin virtual machine is now free to send another message to PVM. PVM's pacing support will control the number of messages that the origin can have outstanding before the target virtual machine is able to receive them. The origin virtual machine can also use two-way sends, but it will not get the Message Complete external interrupt.

- **8** The origin PVM node moves the data into PVM message packets and forwards the data to the target PVM node.
- **9** The target PVM node sends the data to the target virtual machine. The Message Complete external interruption indicates that the virtual machine has performed an IUCV RECEIVE. This confirmation is not returned to the origin virtual machine on one-way communications. If the origin virtual machine requires confirmation it should use the IUCV two-way protocol with a reply expected.

- **10** The origin virtual machine has sent all of the data it intends to send and terminates its IUCV session by issuing an IUCV SEVER. PVM will send to the target any remaining messages that it has not completed sending and then terminate the local session and forward a session termination to the remote PVM node to terminate its session. The remote PVM node notifies the target virtual machine of the terminate through an IUCV SEVER.

**GUPI end**

## Chapter 16. Customizing CVIEW

After the CVIEW distribution tape is installed, you can modify the installed options to meet the needs of your establishment. CVIEW provides two product files, the CONSULT IDNAME file and the DXGKEYS LIST file, that you can edit. CVIEW also provides a customization exec, DXGINIT, which can be executed to incorporate your changes. This chapter provides a detailed description of each installed option and instructions on modifying them.

**Note:** DXGINIT must be run before testing and using CVIEW.

### Listed Consultants — CONSULT IDNAME

You can use the CONSULT IDNAME file to create a list of consultants for the GIVE and RECEIVE panels that are displayed to the consultant and recipient during setup. You can edit this file to define up to 11 listed consultants. For each listed consultant, you can specify name, subject or area of responsibility, telephone number, and password.

The CONSULT IDNAME file has a fixed format. Listed consultant information is added to the last 11 data lines and each data line begins with a number from 2 to 12. This number becomes the consultant's unique identifier, known as the **consultant's ID**. The field format of each data line is as follows:

Field Name	Columns	Length
Consultant's name	10 - 33	24
Subject/responsibility	35 - 63	29
Phone	65 - 79	15
Password	81 - 88	8

### Defining Listed Consultants

To list consultants for your establishment, do the following:

1. Edit the CONSULT IDNAME file. **Do not** delete unused data lines and **do not** change the numbers at the beginning of each data line.
2. Enter the names of your consultants, their subjects or areas of responsibility, their telephone numbers, and their passwords. The use of passwords is optional. Assigning a consultant a password prevents another user from using the consultant's unique identifier for a consult session. Passwords can be added, deleted, or changed later using the PASSWORD privileged command.
3. After making all the entries, file the updated file. The entries, excluding passwords, are displayed to the consultant and recipient during setup mode to identify the consultant.

You can edit the CONSULT IDNAME file whenever you need to change your list of consultants. To incorporate CONSULT IDNAME file changes, execute the customization exec, DXGINIT, to build new GIVE and RECEIVE panels. **Do not** edit the CONSULT IDNAME file or execute DXGINIT while CVIEW is in operation.

---

### Program Function Key Assignments — DXGKEYS LIST

Use the DXGKEYS LIST file to assign PF keys for the HELP, QUIT, HOME, and SESSION functions on the CVIEW product panels. You can change only the PF key number, not the file format or function. After installation, the DXGKEYS LIST file contains the following records:

HELP = PF1

QUIT = PF3

HOME = PF4

SESSION = PF12

### PF Key Functions

The **HELP function** displays an explanation panel to the user. The explanation panel contains additional information about the panel being displayed when the user presses the PF key for HELP. The user returns to the previous panel by pressing the ENTER key.

Using the **QUIT function** from a setup panel drops a user out of setup mode (and, thus, out of CVIEW). In addition, if a consultant presses the QUIT PF key from his CONSULT SESSION CONTROL panel, the consult session ends. If a moderator presses the QUIT PF key from his CONFERENCE CONTROL panel or the CONFERENCE PARTICIPANTS panel, the conference session ends. Any user pressing the QUIT PF key when the NOTIFY panel is being displayed terminates the entire session **for all participants**. When the QUIT PF key terminates a session, all session participants except the one who pressed the QUIT PF key return to the HOME panel. The one who pressed QUIT exits CVIEW.

The **HOME function** returns a user to the HOME panel during setup mode. In addition, if a consultant presses the HOME PF key from his CONSULT SESSION CONTROL panel, the consult session ends and the HOME panel is displayed to the recipient also. If a moderator presses the HOME PF key from his CONFERENCE CONTROL panel or the CONFERENCE PARTICIPANTS panel, the conference session ends, and the HOME panel is displayed to all conference participants. Any user pressing the HOME PF key when the NOTIFY panel is being displayed terminates the entire session **for all participants**, and returns all session participants to the HOME panel.

A moderator uses the **SESSION function** to start a conference session after at least one conference participant completes setup. The moderator can also return to a conference session in progress from the CONFERENCE PARTICIPANTS panel by pressing the SESSION PF key.

### Changing PF Key Assignments

To change the PF key assignments for your installation, do the following:

1. Edit the DXGKEYS LIST file. **Do not** change the words 'HELP, QUIT, HOME, or SESSION' and **do not** delete any records.
2. Enter 'PF1' through 'PF12' to change the current setting for a function. A PF key must be assigned to each function. The same PF key cannot be assigned to more than one function. Your definitions for PF1 through PF12 will be duplicated for PF13 through PF24.

3. After changing the PF key assignments, file the updated file. When your changes are incorporated, your PF key assignments will appear on the CVIEW panels.

You can edit the DXGKEYS LIST file whenever you need to change PF key assignments. To incorporate DXGKEYS LIST file changes, execute the customization exec, DXGINIT, to build a new PF key area on the CVIEW panels. **Do not** edit the DXGKEYS LIST file or execute DXGINIT while CVIEW is in operation.

---

## The Customizing EXEC — DXGINIT

You use the customization exec, DXGINIT, to change the number of concurrent users, the occurrence of the NOTIFY panel, or the defined privileged users, and to incorporate any changes in the CONSULT IDNAME and DXGKEYS LIST files. DXGINIT issues prompting messages to guide you through the customization process. Each customization prompt includes a default. The first time you execute DXGINIT after installation, the customization defaults are the same as the installed options. You enter new options to customize CVIEW for your use. See “Executing the Customization EXEC” on page 168 for the step-by-step procedure.

## Concurrent Users

CVIEW can support a maximum of 30 users at separate display stations who have dialed the CVIEW virtual machine. 5 megabytes are required for the full 30 users. When CVIEW is first installed, the number of concurrent users is set to 30, so the initial customization default is 30. You can change this value to any number from 2 to 29.

## The NOTIFY Panel

Because a user's display station could be set up to participate in a consult or conference session without his knowledge, CVIEW provides a mechanism to let a user know that another user is sharing his interactive session. The NOTIFY panel advises a user that someone else is watching his interactive session and solicits a response before continuing the session. The consult or conference session is suspended until all NOTIFY panels displayed for the session receive a positive response. When a user presses the ENTER key from the NOTIFY panel, he is indicating that he wants to continue the session. If any user presses the QUIT or HOME PF keys from the NOTIFY panel, it indicates a negative response to the NOTIFY panel and the entire session end **for all participants**.

The display of the NOTIFY panel during CVIEW sessions is optional. The initial customization default is YES.

The NOTIFY panel can be displayed to all participants in a session or to only the active participants in a session. The initial customization default for this option is ACTIVE. (An *active* session participant is one who has entered data or commands since the last NOTIFY panel was displayed.)

NOTIFY panels appear during a CVIEW session at two different time intervals: so on after the session begins and at regular intervals thereafter. The initial customization defaults for the two intervals are 15 seconds and 9000 seconds. You can change either or both of these values to any number of seconds other than zero.

### Privileged Users

Privileged users have the same capabilities from their own virtual machine consoles as an operator logged on to the CVIEW virtual machine console. CVIEW supports a maximum of 15 privileged users. A privileged user can send CVIEW privileged commands to the CVIEW virtual machine for execution from his own virtual machine console. The privileged commands control and administer the CVIEW virtual machine. They are described in “CVIEW Commands” on page 76. You should review these commands before defining privileged users. The initial customization default defines the CVIEW and OPERATOR user IDs as the only privileged users.

Note that if you assign a privilege class of ‘B’ to the CVIEW virtual machine, any privileged user can send any of the privilege-class-B CP commands to the CVIEW virtual machine for execution. Some privilege-class-B CP commands affect the operation of the CVIEW virtual machine and system resources. In addition, if you assign a privilege class of ‘G’ to the CVIEW virtual machine, privileged users can enter commands, such as LOGOFF, that affect the operation of the CVIEW virtual machine.

### Executing the Customization EXEC

After you edit the CONSULT IDNAME and DXGKEYS LIST files, you can execute DXGINIT. To change a customization default, enter the value you want to use; to accept a customization default, press ENTER in response to the prompting message. To execute DXGINIT, do the following:

1. Log on to the CVIEW virtual machine. (Enter: L CVIEW)
2. If the CVIEW virtual machine is in operation, enter: SHUTDOWN
3. Enter: DXGINIT
4. Customize issues the first prompting message:

*Have you entered the consultant IDs in the CONSULT IDNAME file?  
Enter Yes or No. Press ENTER for YES.*

If you have edited the file to list your consultant or you do not want to list consultants, press the Enter key and proceed to step 5.

If you have not edited the file and need to list your consultants, enter *No* and DXGINIT will end. You must then edit the CONSULT IDNAME file and restart this procedure at step 3.

5. Customize issues a prompt requesting the number of concurrent users:

*How many users can DIAL into CVIEW?  
Enter a value between 2 and 30. Press ENTER for 4 users.*

6. Customize issues a prompt to determine if NOTIFY panels are to be displayed:

*Do you want to use the Notify Panel?  
Enter Yes or No. Press ENTER for YES.*

If your response was *Yes*, proceed to the next step. If your response was *No*, proceed to step 10.

7. Customize issues a prompt to determine which users are to receive the NOTIFY panel in session mode:

*Do you want to notify all participants in a session or only the active participants in a session?*



*Enter all or Active. Press ENTER for ACTIVE.*

8. Customize issues a prompt to determine when the first NOTIFY panel should be displayed:

*How many seconds should elapse until the Notify Panel is shown the first time?*

*Enter the number of seconds. Press ENTER for 15 seconds.*

9. Customize issues a prompt to determine when subsequent NOTIFY panels should be displayed:

*How many seconds should elapse between subsequent Notify Panels?*

*Enter the number of seconds. Press ENTER for 9000 seconds.*

10. Customize issues a prompt to define the privileged users for the CVIEW virtual machine:

*List of userids that can issue privileged commands:*

*CVIEW OPERATOR*

*Press ENTER to keep the current list of userids.*

*To replace the list, enter the new list of userids on one line, separating each userid with a blank and then press ENTER.*

11. Customization has issued all the prompts that request information. It saves the values you entered and builds all the product panels to incorporate any changes you made to the CONSULT IDNAME and DXGKEYS LIST files. When customization finishes, a completion message appears.

The values you select during customization are saved in the LASTING GLOBALV file on the CVIEW 191 A-disk. The next time you execute the customization exec, the prompts will show those saved values as the customization defaults.



## Part 4. Reference Information

<b>Appendix A. PVM Macros Supported as Programming Interfaces</b>	175
Product-Sensitive Programming Interface Executable Macros	175
<b>Appendix B. PVM Commands</b>	177
* (Comment)	182
AUDIT	182
AUTHORIZ	183
CDBC	185
CMD	186
CMS	186
CP	187
CPQ	188
DEFINE	189
DISC	194
DROP	195
EXEC	196
LMSG	197
LOGMSG	198
MPVM	199
MSG	202
PASSTHRU	203
PAUSE	207
PMSG	208
PVMBLD	209
PVMCMP	210
QUERY	211
QUIESCE	214
QUIT	217
RESUME	217
ROUTE	218
RUNPVM	219
SELECT	220
SET	220
SHOW	222
SHUTDOWN	223
START	224
STATUS	225
TCP	227
TRACE	228
VARY	230
CP Commands Used with PVM	231
DIAL	231
SMSG	232
<b>Appendix C. PVM Configuration File Records</b>	233
* (Comment)	233
AUDIT	233
AUTHORIZ	235
AUTO	235
BUFFERS	236

## Reference Information

DUMP	238
EXT	238
GROUP	239
I/O	240
LANG	241
LINK	242
LNODE	244
LOCAL	245
MAXCONN	245
MAXIDLE	246
MAXSESS	246
MAXSUSP	246
MAXTDISC	247
MAXUSERS	248
MSGNOH	248
MSGOPTNS	248
OPID	250
PVMPROP	250
RESID	252
ROUTE	252
SELECT	253
SESSTERM	253
STATTIME	254
SUBSET	254
TCP	255
TDISC	255
TIMEOUT	256
XDBC	256
 <b>Appendix D. PVMG Commands</b>	 259
AUTHORIZ	262
CDBC	264
CMD	265
DISC	266
DROP	266
INIT	267
LMSG	268
LOGMSG	269
MSG	269
PMSG	270
QUERY	271
QUIESCE	273
RESUME	274
ROUTE	274
RUNPVMG	275
SELECT	276
SHOW	276
SHUTDOWN	277
SPOOL	279
START	280
TRACE	281
CP Command Used with PVMG	282
SMSG	282

<b>Appendix E. PVMG Configuration File Records</b>	283
*	283
APPLN	284
AUTHORIZ	285
DUMP	285
LANG	286
LINK	286
LNODE	287
LOCAL	287
LOGMODE	288
LOGONID	289
LU	290
MSGNOH	291
MSGOPTNS	292
OPID	293
PVMPROP	294
ROUTE	294
SELECT	295
SESSDISC	295
SESSTERM	296
XDBC	296
<b>Appendix F. Setting Up an APPC Link</b>	299
<b>Appendix G. PVM Control Block Formats</b>	301
<b>Appendix H. PVM Access Security Exits</b>	319
The REMOTE AUTH File	319
Definitions of the REMOTE AUTH File Entries	321
GATELINK	321
REM-CMD	322
REM+USER	323
SNA-USER	324
VM-USER	325
Messages	327
Format of SECAUDIT LOG Records	328
Controlling Local Users Using the REMOTE AUTH Table	329
Controlling Remote Users by Using the REMOTE AUTH Table	332
<b>Appendix I. CVIEW Product Panels</b>	335
Setup-mode Panels	335
Notify-mode Panels	343
Control-mode Panels	345



---

## Appendix A. PVM Macros Supported as Programming Interfaces

The macros identified in this appendix are provided to allow a customer installation to write programs that use the services of PVM. Only those PVM macros (other than those provided by VM) identified in this appendix, should be used to request or receive services of the PVM component of the VM/Pass-Through Facility.

---

### Product-Sensitive Programming Interface Executable Macros

The following section lists the PVM executable macros in DVMMAC MACLIB that are supported as Product-Sensitive programming interfaces.

- DVMENTR (see “DVMENTR” on page 116 for further details.)
- DVMRETN (see “DVMRETN” on page 116 for further details.)
- DVMXTRN (see “DVMXTRN” on page 117 for further details.)





## Appendix B. PVM Commands

This appendix describes the commands and EXECs that are used to control and use PVM. (In this appendix, EXECs can be considered synonymous with commands.) It includes discussions (starting on page 231) of two CP commands: DIAL and SMSG.

**Notes on Using Commands:** Certain PVM commands can be issued by users identified in AUTHORIZ records in the PVM configuration file. These commands are the *A* commands listed in Table 18 on page 180. All PVM commands are listed in this table. Some commands must be issued by a PVM operator logged onto the PVM user ID. These commands are the *O* commands. A virtual machine user may use any of the unrestricted commands. These commands are the *G* commands. Some commands are intended to be used by the PVM system programmer. These commands are the *S* commands.

Some PVM command options have restriction levels that are different from the other options for that command.

Unless the issuer is logged on to the PVM user ID, each PVM command expression must be included in a CP SMSG (Special Message) command that contains the PVM user ID. This sends the command to PVM to be executed. Issuing commands from CMS can be simplified by providing an EXEC for a command that automatically puts the *SMSG pvmid* characters in front of the PVM command expression.

A command prefix can be used when issuing PVM commands. Command prefixes tell PVM to send command responses using a particular format. See “Using Command Prefixes” on page 149 for more information about command prefixes. For information on how to use command prefixes in application programming, see Chapter 13, “Facilities for Automating PVM and PVMG Operations” on page 145.

### How to Read the Syntax Diagrams

Throughout this part of the book, syntax is described using the structure defined below.

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

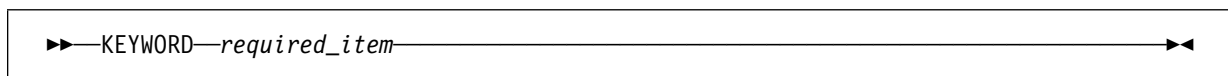
The ►— symbol indicates the beginning of a command.

The —► symbol indicates that the command syntax is continued on the next line.

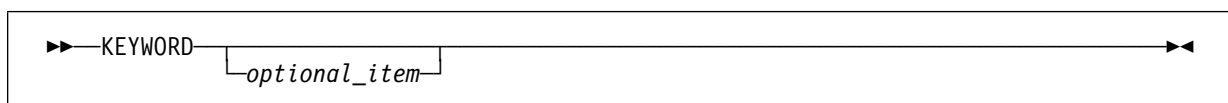
The ►— symbol indicates that a command is continued from the previous line.

The —►◀ symbol indicates the end of a command.

- Required items appear on the horizontal line (the main path).



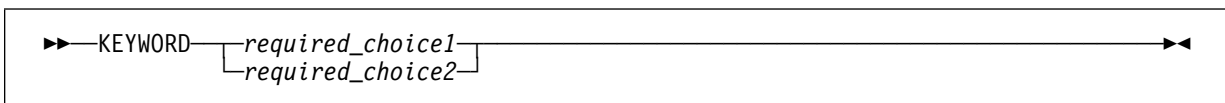
- Optional items appear below the main path.



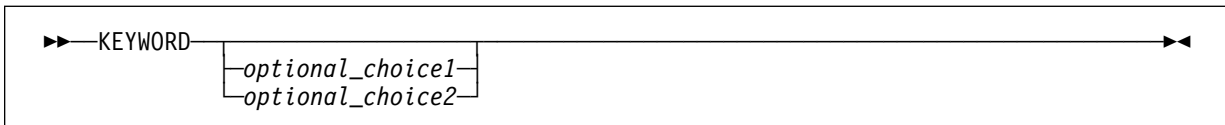
- If there is a choice from two or more items, they appear vertically, in a stack.

If one of the items *must* be chosen, one item of the stack appears on the main path.

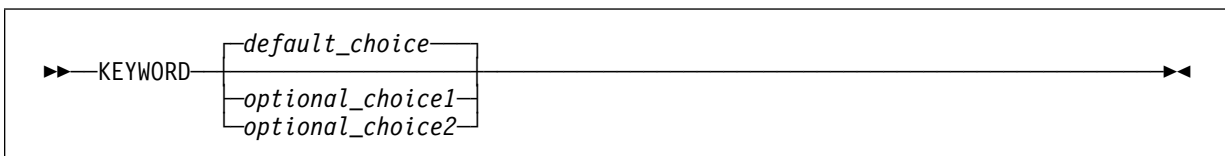
## PVM Commands



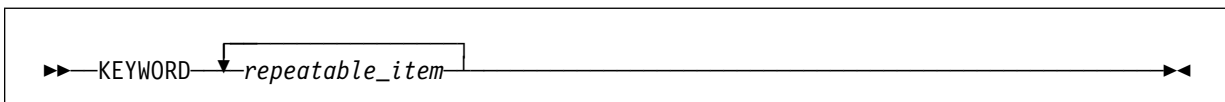
If choosing one of the items is optional, the entire stack appears below the main path.



If one of the choices is a default, it appears above the main path.

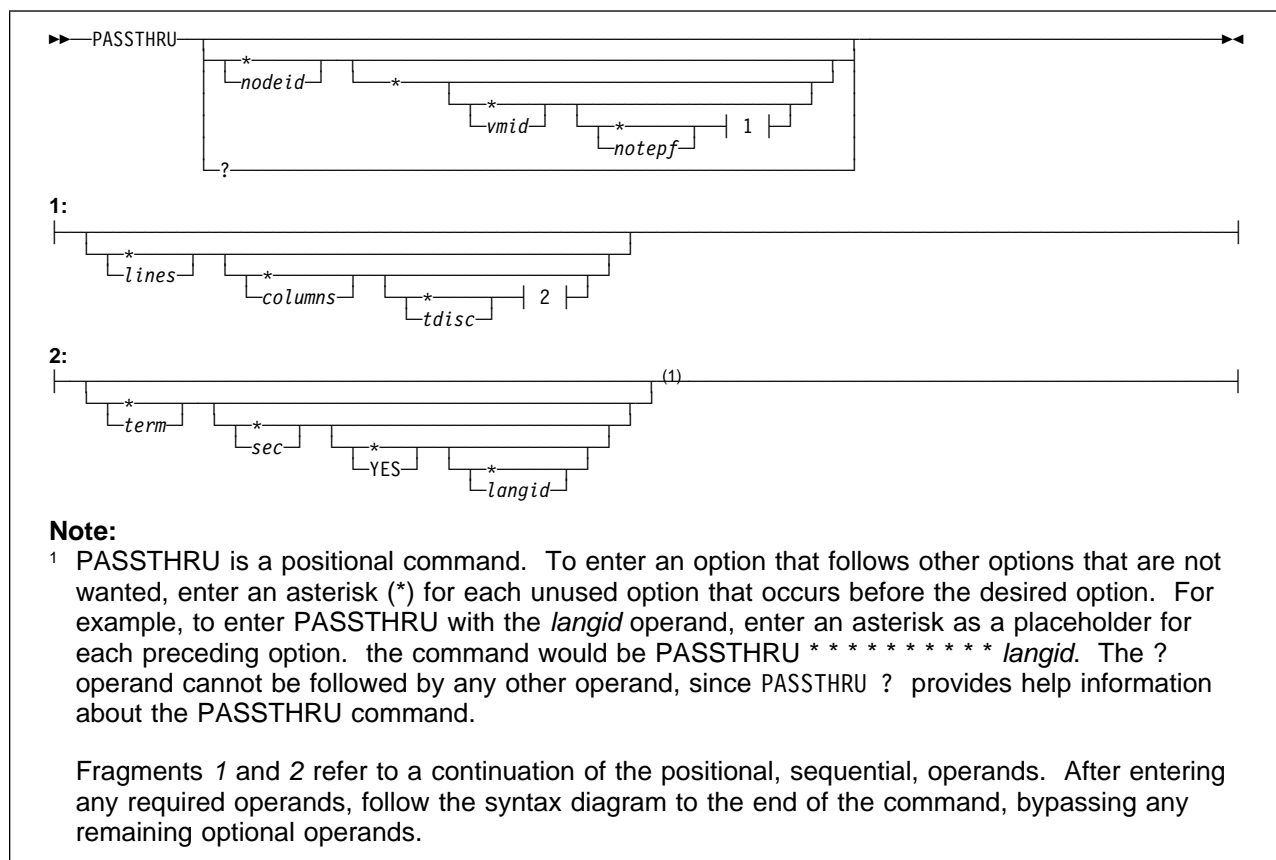


- An arrow returning to the left above the main line indicates an item that can be repeated.



A repeat arrow above a stack indicates that the items in the stack may be repeated.

The following example of the PASSTHRU command illustrates many aspects of a syntax diagram. An explanation of the PASSTHRU command is provided within the syntax diagram.



### Online HELP is available

As a VM user, information about selected PVM commands is available online using the z/VM HELP Facility. To display a menu of selected PVM commands, from CMS enter:

```
HELP PVM MENU
```

To display information for a specific PVM command, PASSTHRU in this example, from CMS enter:

```
HELP PVM PASSTHRU
```

For help with the z/VM HELP Facility, enter `help` to display the main HELP menu or enter `help cms help` for detailed information about the HELP command.

Help for PVM messages with identifiers is only available through the z/VM HELP Facility. Only messages without identifiers and Compiler messages are documented in the PVM hardcopy library.

See the *z/VM: CMS User's Guide*, SC24-6266 for a description of HELP and how to tailor HELP. See the *z/VM: CMS Commands and Utilities Reference*, SC24-6260 for the format and syntax of HELP and related commands.

Table 18 lists the commands, their functions, and the users authorized to use them. Note the following about the first column of this table:

## PVM Commands

Table 17. Key to authorized users.

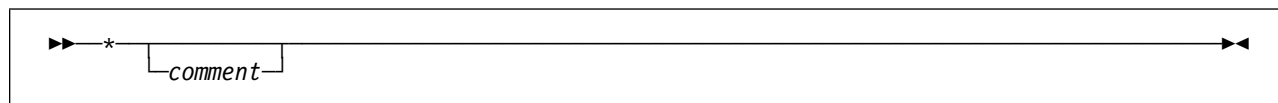
User	Meaning
A	Alternate operator specified on an AUTHORIZ record in the PVM configuration file.
O	PVM operator logged onto the PVM user ID.
G	General VM user.
S	PVM system programmer.

Table 18 (Page 1 of 2). PVM Commands and Execs and Their Functions

User	Command	Function
A O G	*	Comment following asterisk prints out on the PVM operator's display station, but no function is performed. If a PVM audit file is being kept, the comment is also written to that file.
A O	AUDIT	Controls the creation and recording of the PVM audit file.
A O	AUTHORIZ	Adds or deletes entries from the PVM authorized user list. This command must be issued on the local system.
A O	CDBC	Specifies whether display unit can or cannot handle messages in the default language without regard to double-byte character restrictions.
A O G	CMD	Sends PVM commands to another PVM for execution.
A O	CMS	Executes CMS subset commands within PVM.
A O	CP	Executes CP commands within PVM.
A O G	CPQ	Issues one out of a specific set of CP commands on PVM.
A O	DEFINE	Adds or modifies a PVM link, defines a new line group, or adds links to a line group.
O	DISC	Disconnects the PVM user ID.
A O	DROP	Terminates an attached display or printer support task, a line, or a user session.
A O	EXEC	Executes the PVM commands in a specified CMS file.
A O	LMSG	Sends a message to PVM users on a specified link.
A O	LOGMSG	Defines the LOGMSG area on the PVM Selection Menu and on the Initial Menu.
A G	MPVM	Starts a MPVM session from a CMS user ID.
S	MPVMBLD	Regenerates the MPVM user module.
A O G	MSG	Causes PVM to forward a message to a specified VM user.
A G	PASSTHRU	Starts a PVM session from a CMS user ID.
A O	PAUSE	Causes the PVM command processor to enter a wait state for one second.
A O G	PMSG	Causes PVM to forward a message to specified PVM users. The ALL option is restricted to the A and O user groups.
S	PVMBLD	Regenerates the PVM MODULE.
S	PVMCMP	Compiles a PVM message repository.
S	PVMGEN	Generates the PVM module that executes in a user's CMS virtual machine.
A O G	QUERY	Displays information about PVM. The AUTHORIZ option is restricted to the A and O user groups and to the local VM system.
A O	QUIESCE	Quiesces an attached printer or display support task, a line driver task, or the PVM system.
O	QUIT	Terminates the PVM system immediately.
A O	RESUME	Reverses the effect of the QUIESCE command.
A O	ROUTE	Adds, deletes, or changes a node's routing.
O	RUNPVM	Loads and starts PVM.
A O	SELECT	Adds, deletes, or changes nodes assigned to PF keys on the PVM Selection Menu or on the Initial Menu.
A O	SET	Sets the maximum times for idle, suspended, and temporarily disconnected MPVM users, and for the number of users and sessions allowed for MPVM users.
A O G	SHOW	Shows the path of a particular session.
O	SHUTDOWN	Terminates the PVM system.

<i>Table 18 (Page 2 of 2). PVM Commands and Execs and Their Functions</i>		
User	Command	Function
A O	START	Starts a line driver task, an attached display support task, or an attached printer support task.
A O	STATUS	Displays the status of the system pools or the status of tasks within the PVM system.
A O	TCP	Used to start or stop the TCP/IP redirector task and to control tracing of the TCP/IP redirector task.
A O	TRACE	Initiates or stops the line tracing facility, the PVM multitasking supervisor trace facility, or the tracing of a PCCF support task.
A O	VARY	Alters the availability status of links and devices.

## \* (Comment)



## Purpose

Use the \* command to enter comments. This command is non-operational.

## Operands

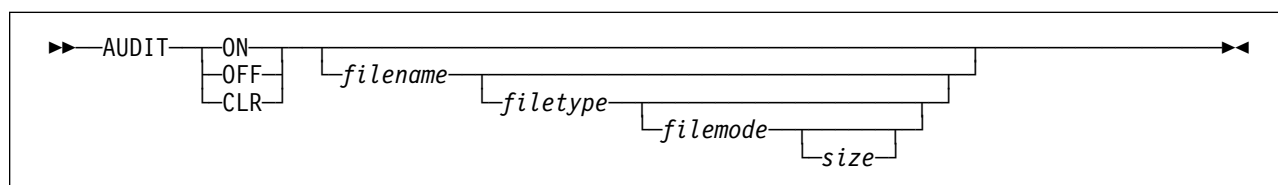
*comment*

is the information reflected on the PVM console and in the optional AUDIT file.

## Usage Notes

1. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

## AUDIT



## Purpose

Use the AUDIT command to dynamically control the creation and recording of the PVM audit file.

## Operands

### OFF

closes the audit file and stops the recording of the audit file.

### ON

opens the audit file. If the audit file does not exist, the AUDIT ON command creates and initializes the audit file.

### CLR

erases the audit file, if it exists. Then, AUDIT CLR creates and initializes a new audit file.

### *filename*

is the name of the audit file. Specify =, to use the current file name. CONSOLE LOG A6 is the default file name, file type, and file mode for the audit file.

*filetype*

is the file type of the audit file. Specify =, to use the current file type.

*filemode*

is the file mode of the audit file. Specify =, to use the current file mode. If only the letter is specified, 6 is the file mode.

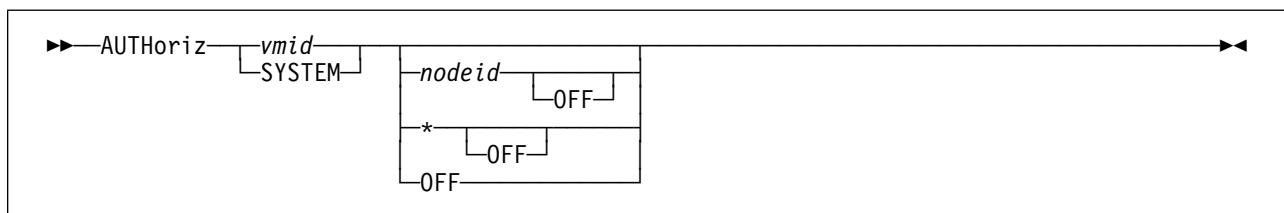
*size*

is the maximum size of the file in 1000s of records. This value can be from 1 to 1000. For example, a size of 3 limits the file to 3000 records; a size of 40 limits the file to 40,000 records.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. Specify ON to clear the specified file if:
  - The *size* value is specified and is greater than the current *size* value
  - The file exists and is in the wrong format, or record one does not contain a valid pointer.
3. When the audit file is full, the data wraps to the beginning of the file overlaying the existing records.
4. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

## AUTHORIZ



## Purpose

Use the AUTHORIZ command to add and delete entries from the PVM authorized user list. When PVM initializes, the authorization list holds up to 30 extra entries. The change is effective only during the execution time of the PVM program. To make this change permanent, add or delete an AUTHORIZ record in the PVM configuration file.

## Operands

*vmid*

is the user identification of the CP virtual machine to add to, or delete from, the authorization list. If neither \* or *nodeid* is specified, this authorization is valid for the local node only.

## AUTHORIZ Command

### SYSTEM

authorizes the PVM operator to issue commands to nodes (*\** or *nodeid*) other than the local node. Use this operand rather than the PVM *vmid*. Without specifying *\** or *nodeid*, this authorization is for the local node only.

#### *nodeid*

is the specific node where *vmid* is authorized. The same *vmid* can be specified with different *nodeids*.

- \** indicates that *vmid* is authorized on any node. Consider this option carefully, because it may be a security risk in a network where user IDs could be duplicated. A *vmid* cannot be added to the authorization list using both a global node ID (*\**) and a specific node ID.

### OFF

removes the specified user from the authorization list.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file and must be issued on the local system.
2. If the operator makes a typographic error in issuing the AUTHORIZ command, PVM may create an entry in its AUTHORIZ table using this incorrect information. If the error is recognized, delete the entry using the **OFF** option of the AUTHORIZ command and submit the correct AUTHORIZ command again.
3. Use the QUERY AUTHORIZ command to display the contents of the AUTHORIZ table.
4. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command.

## Examples

To add authorization for a user ID named JONES to the PVM node NEWYORK, enter the command:

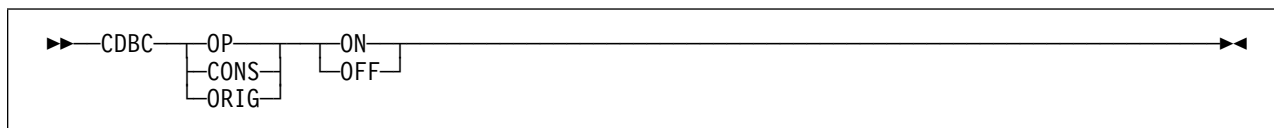
```
auth jones newyork
```

To remove authorization for a user ID named SMITH from all PVM nodes, specify:

```
auth smith * off
```



## CDBC



## Purpose

Use the CDBC (Change *Double-Byte-Characters*) command to change the DBCS-exclusions settings for OP, CONS, and ORIG without having to initialize PVM again. These settings are ignored when a user (other than the PVM console operator) specifically requests a language by using a command prefix.

## Operands

### OP

identifies the CP operator.

### CONS

identifies the PVM console.

### ORIG

identifies all virtual machine users on the local VM system, other than the CP operator and the PVM operator. That is, ORIG identifies the originators of actions or commands to which PVM messages are sent in response.

### ON

indicates that the default language messages should be issued to the user without regard to DBCS content.

### OFF

indicates that the backup language should be used for any messages to the user whenever the default language message that would otherwise be used contains DBCS.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. Issue the CDBC CONS OFF command when logged on to the PVM user ID to prevent errors. Use CDBC CONS ON when the PVM user ID is disconnected.
3. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

---

### CMD

▶▶—CMD—*nodeid*—*command*—◀◀

#### Purpose

The CMD command provides the user the ability to send a command to be executed at another PVM or PVMG node in the PVM network. The user must be authorized at the remote node to have restricted commands issued at that node. The originator receives the output generated from this command. The CMD command is useful for various query commands.

#### Operands

*nodeid*

is the node on the PVM network where the command is to execute. This may be a PVM or PVMG node.

*command*

is any PVM or PVMG command the issuer is authorized to use.

#### Usage Notes

1. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. Place the prefix in front of the CMD command, rather than in front of the *command* portion of the command. For example, use the command:

```
msg pvm (l.kanji) cmd rmtpvm5 query system
```

See “Using Command Prefixes” on page 149 for more information about command prefixes.

---

### CMS

▶▶—CMS—*command*—◀◀

#### Purpose

Use the CMS command to execute CMS commands from within PVM. The output from this command appears only at the PVM user ID display station. Only CMS subset commands are executed using this command.

#### Operands

*command*

is the CMS command executed within PVM.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

**Warning:** PVM stops functioning while the CMS command is executing. If this interval is long, the communications links and PVM users are affected.

---

## CP

»»—CP—*command*—————>>>

## Purpose

Use the CP command to execute CP commands within PVM.

## Operands

*command*

is the CP command executed within PVM. The originator receives the CP output generated from *command*. A maximum of 4096 characters, less the size of the PVM buffer header, of output are returned; the rest are ignored.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. When logged on to the PVM User ID, disconnect while SET RUN is OFF, then reconnect, no log messages or a RECONNECTED message will be displayed. CP READ status will be displayed. The messages will appear after a CP BEGIN command is issued.  
  
However, disconnecting while SET RUN is ON then reconnecting will display the log messages and a RECONNECTED message.
3. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

**Warning:** The system does not check to see if the CP LOGOFF command is entered. The CP LOGOFF command will log off PVM.

## CPQ



## Purpose

Use the CPQ command to issue a specific set of CP commands on PVM.

## Operands

### **CPUid**

displays the local node name, the processor model number, and the level of the VM system.

### **IND**

displays the processor use, processor contention, main storage use, and main storage contention.

### **LOGmsg**

displays the current VM log message.

### **Name**

lists all the users logged on and the display station device address associated with each user. If a user is disconnected, DSC is displayed instead of the address.

### **Time**

returns the equivalent of the CP QUERY TIME command executed on the PVM virtual machine.

### **Users**

displays the number of users logged on and the number of users logically connected to other virtual machines.

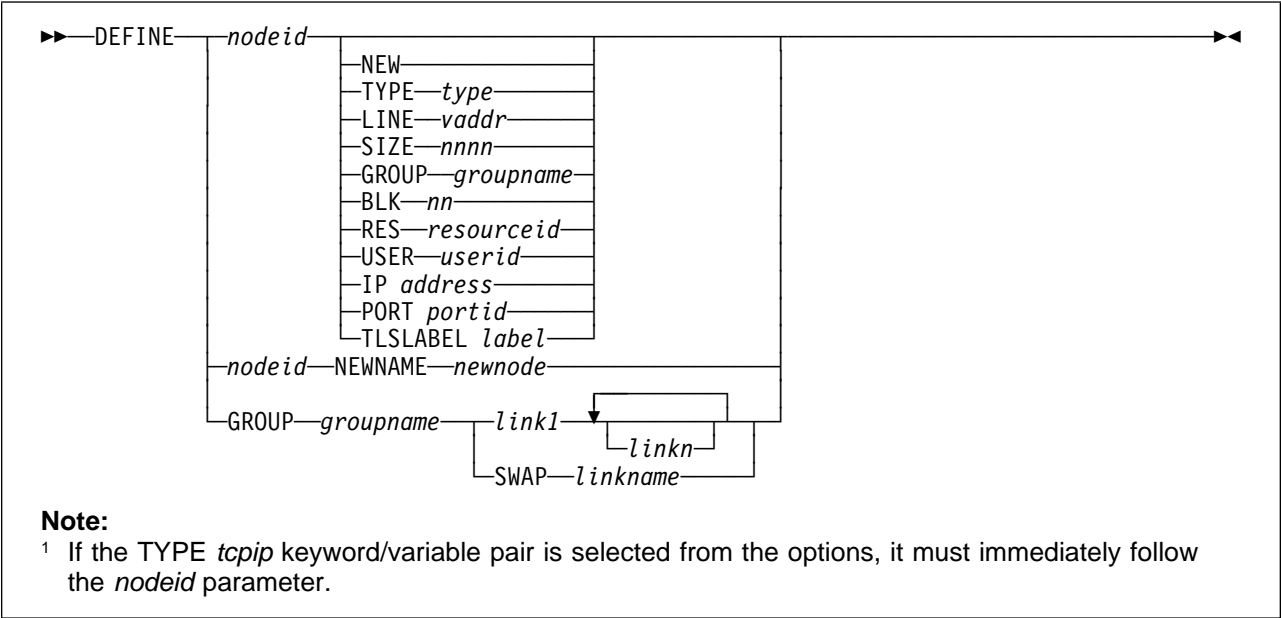
### **Users *userid***

displays the user identification and the display station device address of the specified user if that user is logged on.

## Usage Notes

1. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

DEFINE



Purpose

The DEFINE command allows a system support person to dynamically add or modify a:

- PVM link
- PVM link to a PVMG virtual machine using an IUCV connection
- PVM link to a remote PVM system using an ESCON channel using a 3088 line driver (see Usage Note 1)

Only link types CTCA, APPC, TCPIP, and 3088 can be added to a line group. The line address must be compatible with the link type or an error message will be produced. Only types CTCA, APPC, TCPIP, and 3088 can be changed; and they can only be changed to CTCA, APPC, TCPIP, and 3088. However, other options on the other link types can be changed, provided the link type supports that option.

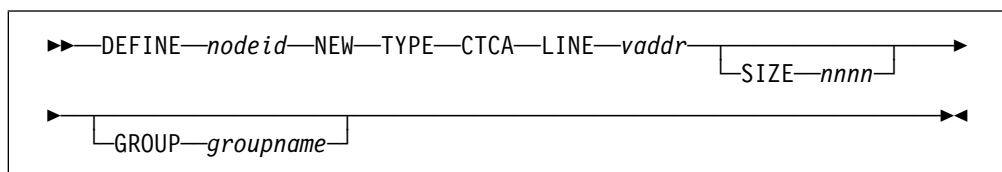
Except for SIZE and GROUP SWAP, use the DEFINE command to modify an existing link only if the link is in the DOWN state. If the link is not in the DOWN state, an error message is produced.

The following are examples to illustrate the format of the line driver types for the DEFINE command. They include formats of the DEFINE command for the specified type of links--CTCA, APPC, IUCV, TCPIP, and 3088.

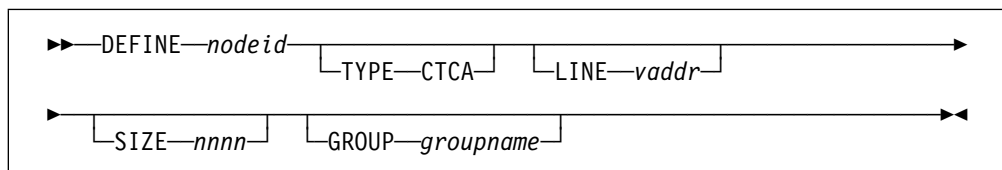
CTCA Format

Use the following format of the DEFINE command to define a new CTCA link.

## DEFINE Command

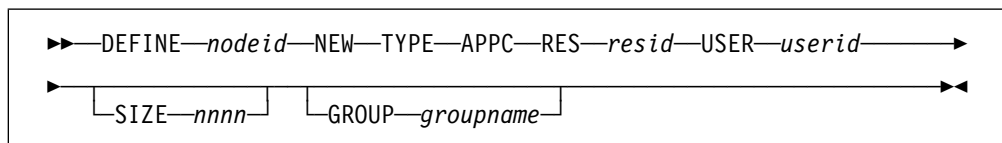


Use the following format of the DEFINE command to change a CTCA link.

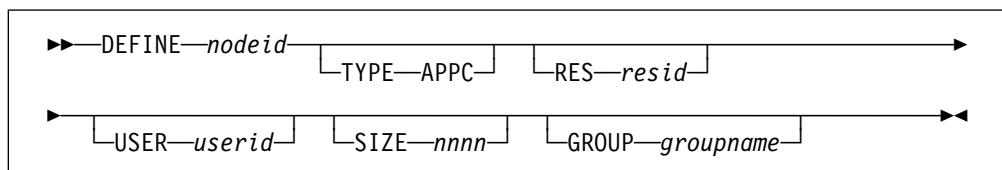


## APPC Format

Use the following format of the DEFINE command to define a new APPC link.

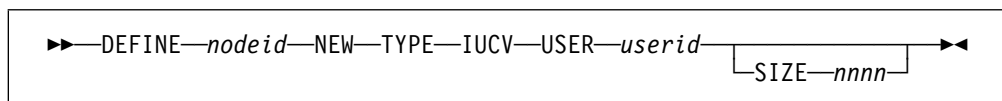


Use the following format of the DEFINE command to change a APPC link.

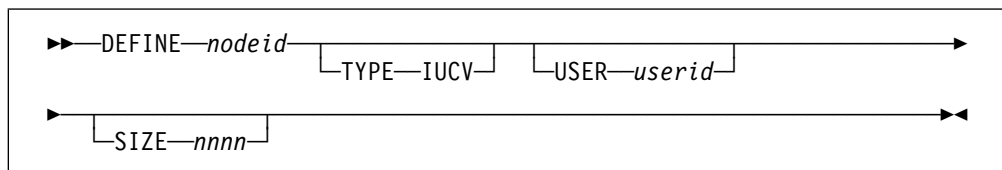


## IUCV Format

Use the following format of the DEFINE command to define a new IUCV link.

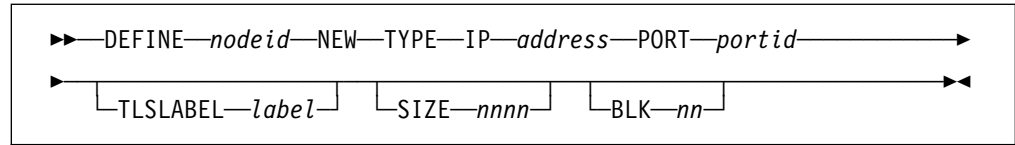


Use the following format of the DEFINE command to change an IUCV link.

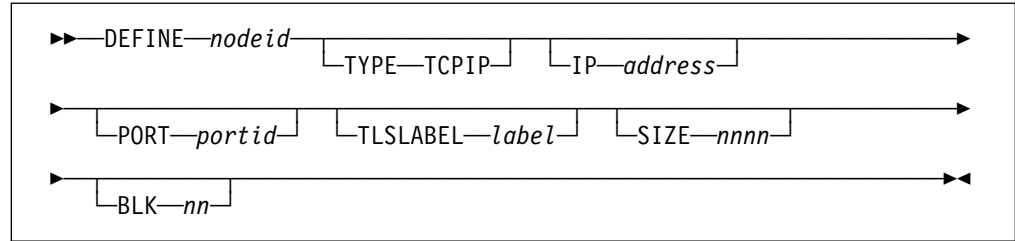


## TCPIP Format

Use the following format of the DEFINE command to define a new TCPIP link.

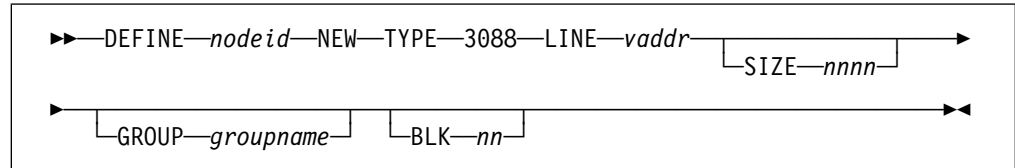


Use the following format of the DEFINE command to change an TCPIP link.

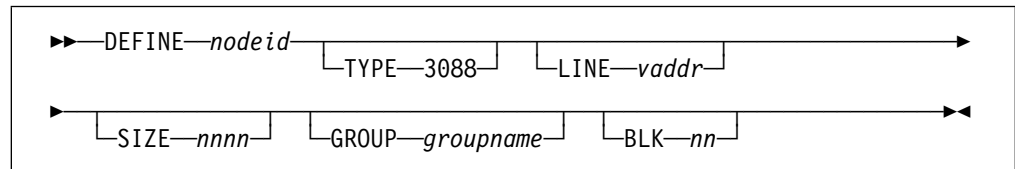


## 3088 Format

Use the following format of the DEFINE command to define a new 3088 link.



Use the following format of the DEFINE command to change a 3088 link.



## Operands

### *nodeid*

is the link node name. The *nodeid* must already exist if new is not specified and *nodeid* must not exist if NEW is specified. The node name cannot already be used as a GROUP name.

### NEW

adds a new link. It must immediately follow the *nodeid*, if present.

### TYPE *type*

assigns a new link type to the link. The type must be CTCA, APPC, IUCV, TCPIP, or 3088. For NEW, TYPE must be included and must follow the keyword NEW. without NEW, TYPE must follow the *nodeid*.

**LINE** *vaddr*

assigns a new line address to the link. For NEW, LINE must be included. The line address is checked to determine if it already exists for another link.

**SIZE** *nnnn*

assigns a new block threshold value to the link. This value governs the minimum block size to be transmitted when there is a queue of data to be sent. The value must be between 200 and 4096. For NEW and not SIZE, the default is 4096.

**GROUP** *groupname*

indicates that *nodeid* is part of the group *groupname*. The specified group must already be defined in PVM. The GROUP operand is only valid for link types that support line groups.

**BLK** *nn*

indicates the number of 4K areas tied together for the output transmission buffer. *nn* must be between 2 and 16. The default value is 3. BLK is only valid if the link type is 3088 or TCPIP.

**RES** *resourceid*

indicates the nickname of the resource to which to connect. This operand only applies to an APPC type link. The :nickname. field in the CMS communications directory (UCOMDIR NAMES) matches this *resourceid*.

**USER** *userid*

For APPC: identifies the local user that is requesting a connection to a remote PVM node over an APPC link.

The *userid* (and password) must match the name defined in the :userid. field of the CMS communications directory.

For IUCV: indicates the user ID of the connecting PVMG virtual machine.

**IP** *address*

indicates the internet address of the remote PVM node. Since an IP address is a 4-level address, each level being a value from 1 to 255, *address* is specified as two fields. The first field contains the first two levels separated by a period, and the second field contains the second two levels separated by a period. An example of *address* would be 9.130 25.200. The IP keyword is only valid for TCPIP type links.

**PORT** *portid*

indicates the port number of the PVM machine at the remote node in the range 10-65534. If not specified, then the port number will default to the one used on the TCP record within the PVM configuration file. The PORT keyword is only valid for TCPIP type links.

**TLSLABEL** *label*

specifies the 1- to 8-character label of an x.509 digital certificate that will be used to encrypt/decrypt all data flowing over the link using TLS protocols. The specified certificate and its corresponding TLSLABEL must exist in the TLS/SSL Server certificate database. For additional information on the TLS/SSL Server and managing its certificate database, see *z/VM: TCP/IP User's Guide* and *z/VM: TCP/IP Planning and Customization*. The TLSLABEL keyword is only valid for TCPIP type links.



## **NEWNAME** *newnode*

modifies the node name of a previously defined link. The restrictions are a:

- Link cannot be currently part of a line group
- Link cannot be started
- New node name cannot be already used as either a link name or a group name.

If any nodes are currently routed to the old link node name, their routing is changed to the new node name of the link.

## **GROUP** *groupname link1 link2 ...*

defines a new line group or adds new links to an existing group. A maximum of eight groups can be dynamically defined.

The *groupname* is the name of the line group. If *groupname* already exists, the link names are added to the current group.

The *link1* is the name of a link assigned to the group. The link must already be defined and cannot be started.

The *link2* is the name of a link assigned to the group. The link must already be defined and cannot be started.

**Note:** Only specify line types that may participate as part of a line group.

**Note:** Only one TCPIP and APPC type link can be in each group.

## **GROUP** *groupname SWAP newlinkname*

takes a link that is not part of a line group, makes the name of this link the name of the group being defined, and renames this link. The *groupname* that was the name of the original link becomes the name of the newly defined group. The *newlinkname* is the new name specified for the original link. The *newlinkname* must not already be used as a link name or a group name.

This function can be executed no matter what state the link is in, but only for links that support line groups. The associated link need not be dropped for the SWAP function.

Note: Only link types that may participate as part of a line group can be specified.

## Usage Notes

1. When defining a FICON channel line driver to a remote PVM system, the same parameters are accepted as would be used to define a 3088 line driver.
2. Line driver grouping is not permitted for a type of IUCV.
3. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
4. The SIZE option can be used to prioritize links in a group. Sessions go through the link in a group that has the highest size (priority), increasing performance. Links that have lesser size values are used for backup purposes.
5. To dynamically define a new line group and links:
  - Define the links without specifying the GROUP parameter
  - Define the GROUP listing the predefined links.
6. To convert a link to a line group:
  - Use the GROUP SWAP operand to convert the link to a line group

- Define any more links adding the GROUP option.
7. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

## Examples

Suppose there is a link from a PVM system whose node ID is PVMA, to another PVM system whose node ID is PVMB. The name of this link is PVMB. To add another link from PVMA to PVMB, do so by using the DEFINE command. First, to define a group. Issue the following DEFINE command:

```
define group pvmb swap pvmb1
```

This command performs two functions. It takes the original link name, PVMB, and makes it the group name. Then, the DEFINE command renames the original link name to PVMB1. Therefore, there is a group with the name PVMB and a line in that group with the name PVMB1. Now add other lines to group PVMB by using the following DEFINE command:

```
define group pvmb link1 ... linkn
```

where link1 ... linkn are the names of the links to add to the group. The links are already defined.

---

## DISC

▶▶—DISC—◀◀

## Purpose

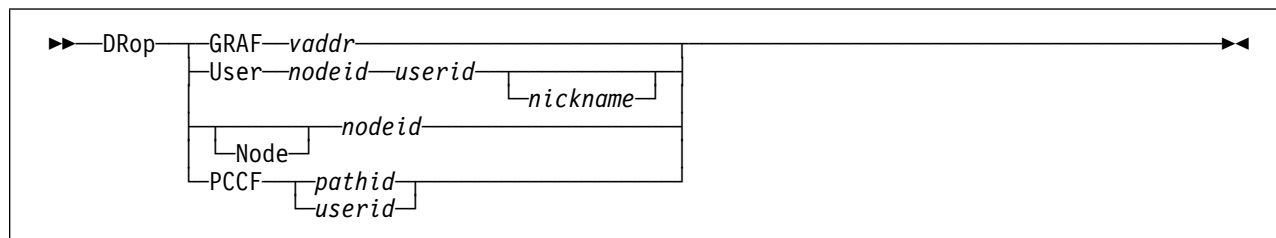
Use the DISC command to disconnect the PVM operator's display station. Use this command rather than the CP DISC command. Using the DISC command allows system spool files to be properly maintained.

## Usage Notes

1. This command is restricted to a PVM operator logged onto the PVM user ID.
2. If the PVM AUDIT command or configuration file record had been previously specified, all "writes" or output messages, including those from PVM, sent to the virtual console will continue to be captured. If PVM tracing to a virtual printer has been started, records will continue to be added as appropriate.
3. If logged onto PVM, disconnecting while SET RUN is OFF, then reconnecting, will not produce any log messages or a RECONNECTED message, but only CP READ status. The messages will appear after a CP BEGIN command is issued.

However, disconnecting while SET RUN is ON, then reconnecting, will display the log messages and a RECONNECTED message.

## DROP



## Purpose

Use the DROP command to terminate:

- An attached display support task
- A user session
- A specific PCCF path
- A PCCF support task started by a START PCCF command

## Operands

### **GRAF** *vaddr*

immediately terminates the specified attached display support task and any active user session.

The *vaddr* is the virtual address of the attached display support task terminated. The address specified must be the same as the address used to start the corresponding attached display support task.

### **User** *nodeid userid nickname*

terminates the specified user session. The user is notified of impending system action. The *nodeid* is the identification of the user's originating node where the user's session is terminated. The *userid* is the identification of the user whose PVM session is terminated.

The session identification is:

- GRAFxxx for PVM dialed users
- the user's user ID for PASSTHRU users (or to drop all of an MPVM user's sessions)
- \*SESnnnn for an MPVM session
- \*nnnnnnn for a PCCF path or user
- an LU name, if the session origin is an SNA application.

For MPVM users, if the *nickname* is supplied, the MPVM session will be dropped if the specified *userid* and *nodeid* are the user's origin VM ID and node ID. A user's session can also be dropped by supplying the PVM user ID for the session (in the form of \*SESnnnn) in the *userid* parameter. In this case, the nickname can be omitted. All of a user's sessions can be dropped by supplying the user's origin VM ID and node ID and omitting the nickname option.

**Node** *nodeid*

immediately terminates the specified PVM link. Any active user sessions associated with this link are also terminated. The *nodeid* is the node name defined in the PVM configuration file.

**PCCF** *pathid*

terminates the PCCF path identified by the PCCF path number *pathid*. If the path is active, it is broken immediately by simulating a SEVER external interrupt to the associated PCCF task.

**PCCF** *userid*

terminates a PCCF support task. The *userid* is the virtual machine for which PVM is handling PCCF connections. If there are any active connections, the task is not terminated until all connections are terminated.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. If there are no sessions left for the MPVM user at the completion of the drop user command, the MPVM user's task will be removed from the PVM virtual machine.
3. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

---

## EXEC

▶▶—EXEC—*filename*————▶▶

## Purpose

Use the EXEC command to execute all PVM commands in the CMS file specified on the command line. The file type of this file must be PVM.

## Operands

*filename*

is the name of the file (file type PVM) to be executed.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. The file may contain any valid PVM commands. The file is processed by reading and stacking all the records within the file and then processing the records as individual commands.

The file can have a fixed or variable record format with a maximum logical record length of 140 bytes. CP commands (for example, CP MSG) are truncated to 240 characters (to conform to the maximum length for a CP

DIAGNOSE code X'08'). If the record format is fixed and the logical record length is from 72 to 80, PVM executes only the first 72 bytes of each record as the command. Columns 73 through 80 are ignored. This allows for serialization of the file.

3. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

---

## LMSG

►►—LMSG—*linkid*—*message*—►◄

### Purpose

Use the LMSG command to send a message to all PVM users associated with the local PVM, or a specific link or group. If the name is the local PVM node name, the message is sent to all users who issued the DIAL, PASSTHRU, or MPVM command on the local system and selected the local node name. If the name is a link name, the message is sent to all users whose session is associated with the specified link. If the name is a group name, the message is sent to all users whose sessions are associated with any link in the group.

When the message arrives at the specified user's system, it replaces the current image on the user's display. The user recalls the stored image by pressing any interrupt key. The sender's node ID and VM ID are displayed with the message at the receiving location(s).

### Operands

*linkid*

is the local PVM node name, a link defined on the local PVM system, or a group defined on the local PVM system.

*message*

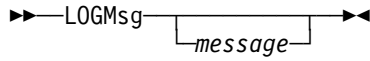
is the text of the message sent. The message can be of any format and characters acceptable to the CP command processor. The total length is limited to the CP command input limit.

### Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. No error message is issued if the specified link has no sessions.
3. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

---

# LOGMSG



## Purpose

The operator uses the LOGMSG command to enter the variable data text that appears in the LOGMSG area on both the PVM Selection Menu and the Initial Menu.

Any entry of the LOGMSG command overlays the previous LOGMSG text. Issuing a LOGMSG command with no text overlays the previous LOGMSG text with blanks.

## Operands

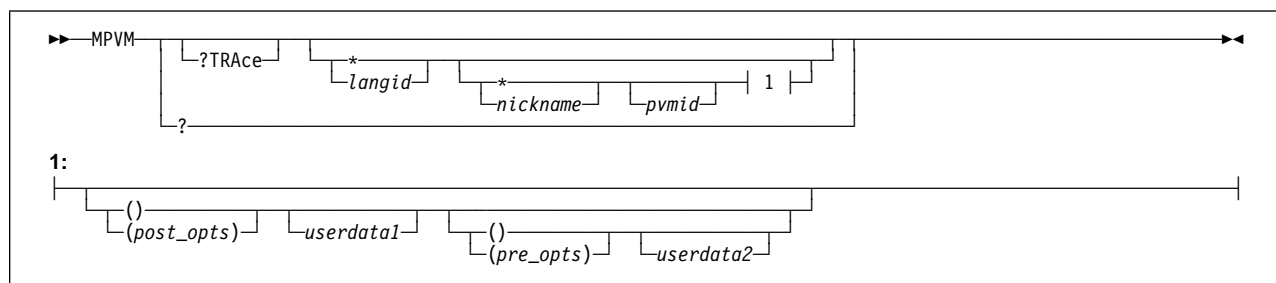
*message*

is the text of the message. Use any characters. This field may be from 0 to 125 characters long, using two lines. With the shortest form of the LOGMSG command, LOGM, the message could be up to 125 characters long. With the longer form, LOGMSG, the message could only be up to 123 characters long.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. When issued from another virtual machine through SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

## MPVM



## Purpose

Use the MPVM command to invoke MPVM. If used without any operands, a panel is displayed showing all session nicknames defined in the MPVMSYS NAMES and MPVM NAMES files.

**Note:** The MPVM command is invoked when MPVM is issued from CMS for the first time. This command calls the MPVM macro using the parameters exactly as given on the original MPVM command line. Subsequent use of MPVM from within the established MPVM environment (such as ?MPVM LOCAL2) calls the MPVM macro directly for establishing additional sessions.

## Operands

? invokes the z/VM HELP Facility for the MPVM command.

### ?TRAcE

Sets the MPVM TRACE on immediately, before session initialization is started.

### langid

use the *langid* parameter to specify the language for MPVM to use for this session. The *langid* is a 1-to-5-character identifier. If the default language is desired and the user wishes to select a session from the command line instead of the selection menu, an asterisk (\*) must be used as a place holder. For example, to invoke the MPVM session PAYROLL, type:

```
MPVM * PAYROLL
```

### nickname

is the 1-to-8-character nickname of the session to establish if the session is not already established. If the session is already active, the name is selected. To specify the next parameter, but not a nickname, use an \* here.

The value specified by the *nickname*:

- Cannot start with an escape (default value ?) character
- Must contain only the characters a—z, A—Z, 0—9, and the following special characters:  
 . , \_ : % & @ # ?
- Must be unique to all currently established sessions for this user
- Cannot be the word *CANCEL*.

### *pvmid*

is the user ID of the PVM server virtual machine to be used. The system could include more than one PVM server virtual machine; however, once an initial connection with a PVM server virtual machine is established, all sessions must be through that same PVM server machine.

### *post\_opts*

specifies MPVM variable name/value pairs that are processed *after* tags in MPVM's NAMES file(s). This allows overriding MPVM variables settings that were defined in the NAMES file(s). These *post\_opts* must be specified in the format:

(*variable\_name value*)

### *userdata1*

specifies additional user information that the MPVM command does not parse. Use this to *pass* data to another MPVM macro; that macro can use the *EXTRACT /ORIGCMD* subcommand to get the entire command string that was supplied on the MPVM command. *userdata1* and *userdata2* will be part of that command string.

To specify *userdata1*, code parentheses (as placeholders) in the *post\_opts* position, even if *post\_opts* is not specified.

### *pre\_opts*

specifies MPVM variable name/value pairs that are processed *before* tags in MPVM's NAMES file(s). This allows defining initial default settings for MPVM variables. These *pre\_opts* must be specified in the format:

(*variable\_name value*)

### *userdata2*

specifies additional user information that the MPVM command does not parse. Use this to *pass* data to another MPVM macro: that macro can use the *EXTRACT/ORIGCMD* subcommand to get the entire command string that was supplied on the MPVM command. *userdata1* and *userdata2* will be part of that command string.

To specify *userdata2*, code parentheses (as placeholders) in the *pre\_opts* position, even if *pre\_opts* is not specified.

## Usage Notes

1. Note the following about the *post\_opts* and *pre\_opts* options:

- MPVM variables that can only be extracted are not valid as *post\_opts* or *pre\_opts*.
- Use these options to give values to *most* MPVM variables that can be set. However, the following variables are exceptions because they have special values:
  - UPDATED
  - EXIT11IF through EXIT20IF.
- MPVM string variables can be set in *post\_opts* and *pre\_opts*. If the string value is only one word, delimiters are not required. If the string value is more than a single word, or if a delimiter character is contained within the word, a delimiter character is required before and after the string value.



The delimiter character can't be used in the string itself, and it must be something other than 0-9, a-z , or A-Z.

Parentheses may be used within string variables in *post\_opts* and *pre\_opts*.

- The *post\_opts* can be used to override variable settings for the current session. To do this, issue the MPVM macro using the nickname of the current session.
- When using the MPVM command to reconnect after a temporary disconnect, *post\_opts* variables that must be set before session initialization are ignored. This is so that the same MPVM command syntax can be used to reconnect as was used with the initial MPVM connection.
- Specifying the MPVM variables EXIT11IF through EXIT20IF as MPVM command line options can be complex, because delimiters must be used within delimiters. Consider the following example:

```
mpvm * mysess (exit11if /"VM READ"3263/ exit11 /MSG There is a VM READ/
```

As a MPVM command option, the *entire* EXIT11IF variable's string value must be surrounded by delimiters; in addition, the part of the EXIT11IF value that represents the string being sought (VM READ) must be delimited from the area specification (3263).

2. To enter mpvm with only a left parenthesis and some postopts, enter

```
MPVM * (TRACE ON
```

instead of entering

```
MPVM (TRACE1 ON
```

This ensures that there is no confusion about the *langid* parameter placeholder.

## Messages

Here are the possible return codes from the MPVM command:

-3	MPVM failed due to an installation error
0	Normal exit after all sessions are terminated by SESSTERM
1	Temporarily disconnected
3	Session was idle too long (MAXIDLE setting exceeded)
4	User quit from an initialization prompt
5	Missing or invalid operand
8	This is an <i>error</i> return code, with a different meaning for each MPVM subcommand.
9	File I/O error
10	Terminal I/O error, or error in 3270 data stream
11	IUCV communications error with PVM server
12	Unable to establish connection to PVM server
15	The last (or only) session was terminated by PVM server without request
16	No current session was initially established or session termination occurred during INITCMD processing
18	The PVM server is in the process of shutting down, or in quiesce mode; no new connections are allowed
19	Maximum number of users are already connected to the PVM server
25	Insufficient free storage available
50	Illegal recursion into the MPVM user module
970	Unable to access MPVM common disk
998	IUCV protocol error
999	Internal logic error

---

# MSG

▶▶—MSG—*nodeid*—*vmid*—*message*————▶▶

## Purpose

MSG causes a line of text to be sent to the specified node and presented to a user or operator using the CP MESSAGE command.

## Operands

*nodeid*

is the identifier of the target node that is to receive the specified message text line. This node can be the local node.

*vmid*

is the identifier of a logged on user at that location.

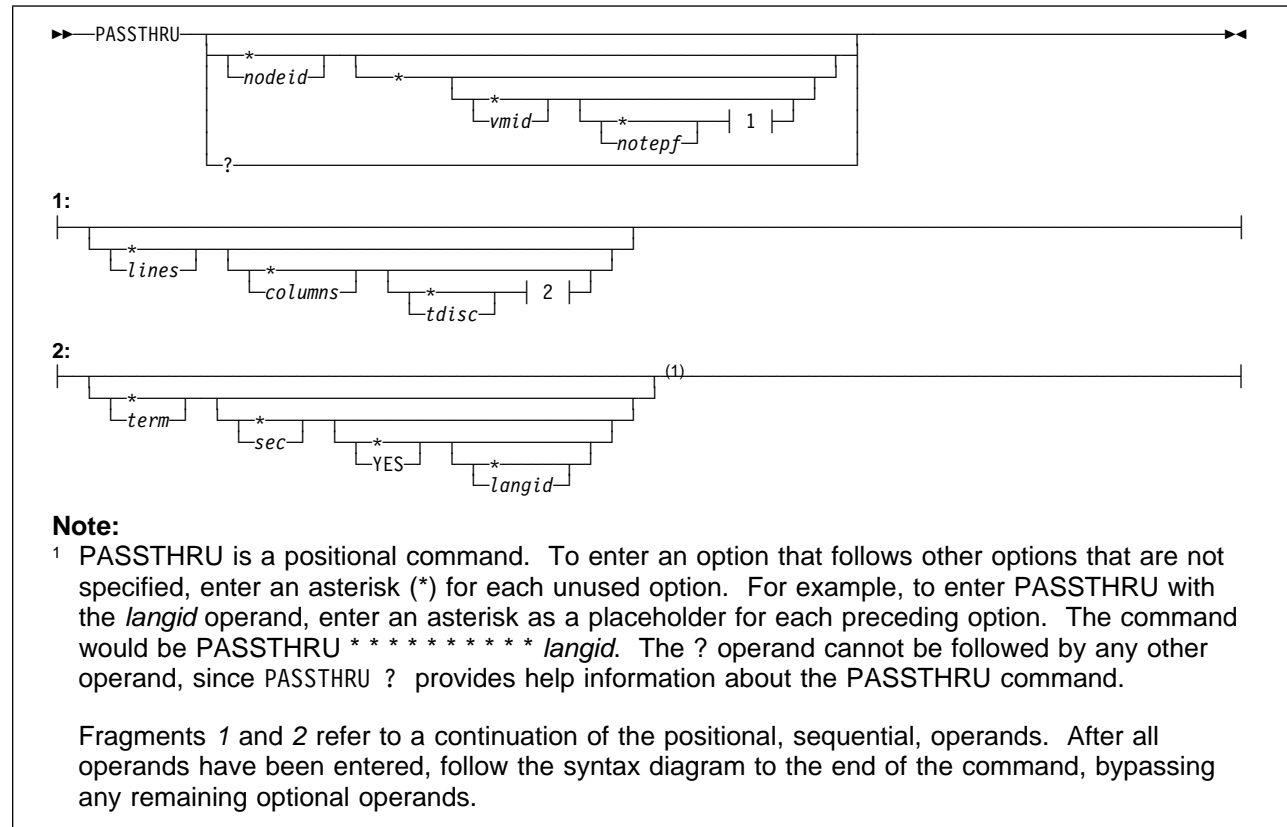
*message*

is the text of the message to be forwarded. The message can be of any format and characters acceptable to the CP command processor. The total length is limited to the CP command input limit.

## Usage Notes

1. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

## PASSTHRU



## Purpose

The PASSTHRU command can only be entered from a 3270-type display station. Users logged on to CMS establish initial contact with PVM by issuing the PASSTHRU command. The target system to be interactively communicated with can be specified by including the target node identifier on the command line. If a node is not specified, one can be selected from the PVM Selection Menu, which will be displayed. The PVM session is active until the user terminates the target application session and exits from PVM.

The PASSTHRU command can execute in CMS subset mode. Defaults can be set for the various parameters and nicknames can be used for nodes.

## Operands

? invokes the z/VM HELP Facility for the PASSTHRU command.

*nodeid*

is the node identification of the target system. If \* is specified, the PVM Selection Menu is displayed.

*vmid*

is the PVM virtual machine ID. If \* is specified, the default (PVM) is used.

*notepf*

is the number of the program function key causing the currently displayed screen to be written to a CMS file named PASSTHRU DATA on the A-disk.

Acceptable values are 1 through 24. Specifying an \* bypasses the Notepad facility. The same PF key cannot be used for the Notepad, temporary disconnect (*tdisc*), or session terminate (*term*) functions. The Notepad facility is available only if a value is specified for *notepf* on the PASSTHRU command line.

### *lines*

is the number of lines from the currently displayed screen written to a CMS file by the Notepad facility, starting from the top of the screen down. Specifying an \* results in a value of the screen size minus 2. Acceptable values are 1 through *nn*, *nn* being the maximum number of lines on the screen of the accessing device. If a value is specified for *lines*, then one must also be specified for *notepf*.

### *columns*

is the number of columns from the currently displayed screen written to a CMS file by the Notepad facility, starting with the leftmost column on the screen. Specifying an \* results in a value of the screen width. Acceptable values are 1 through 132 for 3278 Model 5, or 1 through 80 for other display stations. If a value is specified for *columns*, then one must also be specified for *notepf*. If a PASSTHRU DATA A1 file already exists, its record length must be equal to the value specified in *columns* or the file does not accept Notepad records. Otherwise, if the record length does not equal the value in *columns*, the file must be renamed or erased. A new PASSTHRU DATA A1 file, with the correct record length, is created the next time the Notepad facility is used.

### *tdisc*

is a sequence entered to temporarily disconnect from a PVM session. The sequence can be a 1-to-8-character string (other than a single \*), a PF function key, or a PA function key.

A session may remain disconnected up to the disconnect time limit specified by the installation. The screen image is saved and then restored after the PVM session is resumed. Entering the disconnect sequence while the Selection Menu is displayed terminates the PVM session (temporary disconnect is not activated in this case).

If \* is specified, the temporary disconnect function is not available for that session.

The valid range for PF keys is 1 to 24. Ensure that the device supports the specified PF key before assigning a function to it. PF, PF0, and PFnnn, if specified, are treated as a string. The same character string, PF key, or PA key cannot be used for the temporary disconnect, session termination, or notepad (notepad may only be assigned to a key, not to a string) functions.

### *term*

is a sequence used to terminate a PVM session (following a disconnect from a target application). The sequence can be either a 1-to-8-character string, a PF function key, or a PA function key.

This mode of session termination is not required if the target node is a PVM virtual machine. In that case, the session automatically terminates when the user logs off or disconnects from the target application.

Specifying an \* results in the PVM system default string. Each display station input is tested for the terminate sequence, and if found, the PVM session terminates.

The valid range for PF keys is 1 to 24. Ensure that the device supports the specified PF key before assigning a function to it. PF, PF0, and PFnnn, if specified, are treated as a string. The same character string, PF key, or PA key cannot be used for the session termination, temporary disconnect, or notepad (notepad may only be assigned to a key, not to a string) functions.

*sec*

is a PVM secondary routing node name. If specified, PVM routes the session through the specified node. This is particularly useful if the usual route to a target node is not available.

**YES**

activates I/O verification for this PVM session. Any other value indicates no verification. The default is no verification.

**Note:** Specifying **YES** for verification can cause a significant degradation of performance in the PVM session.

*langid*

identifies the language that PVM will use for the session. If a value for *langid* is not specified, the language used in the PVM session is determined as follows:

1. The language being used for the user's CMS session, except when PVM is not enabled for that language or when the language is a DBCS language and the user's display station cannot handle DBCS
2. The default PVM language except when that language is a DBCS language and the user's display station cannot handle DBCS
3. The PVM backup language.

## Usage Notes

1. CP and CMS facilities can be used to define defaults for PASSTHRU. For example, use:
  - The *userid NAMES* file to define nicknames for target systems
  - A CMS SYNONYM file to set synonyms for the PASSTHRU command
  - The CP SET PFn command to set a PF key to a PASSTHRU command
  - CMS GLOBALV to set up defaults for PASSTHRU parameters
  - An exec written to issue a regularly used or complex PASSTHRU command.

2. To find out what the default session terminate string on the PVM system is, enter:

```
passthru
```

to display the PVM Selection Menu. The default is displayed in the End Session field.

## Examples

Defaults or PF keys can be set so that PASSTHRU can be issued without having to type in values for all the parameters. To make PASSTHRU easier to use:

- Set up nicknames for target systems by using the CMS *userid NAMES* file. Within the CMS NAMES file, use a *:NODE* or *:PVMNODE* tag which PASSTHRU uses but other programs ignore. For example, the name *HARTFORD* could be a nickname for the system *CONN1* and *NYC* could be a nickname for *NEWYORK*. Entries in the CMS NAMES file could look like the following:

```
:nick.HARTFORD      :node.CONN1      ...
:nick.NYC           :pvmnode.NEWYORK  ...
```

- Set a synonym for the PASSTHRU command in a CMS SYNONYM file. For example, to make *PASS* a synonym for PASSTHRU, add this to the SYNONYM file:

```
PASSTHRU  PASS
```

For details on CMS SYNONYM, see the *z/VM: CMS Commands and Utilities Reference*.

- Set a PF key to the command needed to start a PVM session. For example, to set PF01 to start a PVM session and as a temporary disconnect key, enter:

```
cp set pf01 immed passthru newyork * * * * * pf01
```

Pressing PF01 will immediately connect to the *NEWYORK* system. Pressing PF01 from within the *NEWYORK* system temporarily will disconnect from that system, and returns to the host system. By adding this to the CMS PROFILE EXEC, PF01 will be set as the above setting every time CMS is IPL'd. For details on the CP SET command, see the *z/VM: CP Commands and Utilities Reference*.

- Set up defaults for PASSTHRU parameters with the CMS GLOBALV command. For example, to initialize default values for Notepad (the fourth, fifth, and sixth parameters on the PASSTHRU command) and a Temporary Disconnect string (the seventh parameter), enter:

```
globalv select passthru setlp passthru * * * 12 24 80 pf01
```

Thereafter, these defaults apply when the PASSTHRU command is issued. Enter

```
passthru newyork
```

to connect to NEWYORK with custom defaults in effect for Notepad and Temporary Disconnect.

Personal defaults can be overridden. Just use two asterisks (\*\*) in place of those parameters to keep as personal default. For example, to start a PVM session with PF3 set to Temporary Disconnect, enter:

```
passthru newyork ** ** ** ** ** ** ** pf3
```

Use the GLOBALV command to see personal defaults for the PASSTHRU command. To display the default values for the above PASSTHRU example, enter:

```
globalv select passthru list passthru
```

See the *z/VM: CMS Commands and Utilities Reference* for details on the CMS GLOBALV command.

- Set defaults for PASSTHRU parameters within a CMS EXEC. See the *VM/Pass-Through Facility: User's Guide* for details.

---

## PAUSE

▶▶—PAUSE—◀◀

### Purpose

Use the PAUSE command to cause the command processor to enter the wait state for one second. Normally the PAUSE command is used in a PVM exec. This allows any previous commands to complete execution.

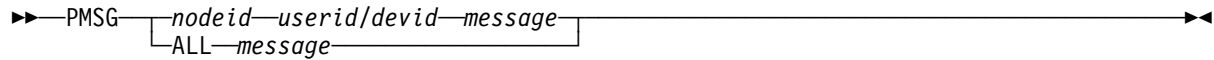
Many of the PVM commands are only partially processed by the command processor. The START command, for instance, is pre-checked by the command processor and then posted to the session manager to complete processing. Because the command processor runs at a higher priority than most other tasks, these type commands do not complete until the command processor has no more work to do.

For example, if a PVM exec contained a DROP command followed by a START command, the START command would not be executed because the DROP command had not completed processing. The PAUSE command allows the sequence to work properly by delaying the command processor to give time for the DROP to complete. The DROP command to some line drivers can take longer than the PAUSE time interval. In this case, more than one PAUSE command can be added.

### Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

### PMSG



### Purpose

Use the PMSG command to send a message to a particular user or all users who have sessions with or through a PVM node. When the message arrives at the specified user's system, it replaces the current image on the user's display. The user recalls the stored image by pressing any interrupt key. The sender's node ID and VM ID are displayed with the message at the receiving locations.

### Operands

#### *nodeid*

is the identification of the target user's node when the message is forwarded to a single user.

#### *userid/devic*

is the identification of the user (*userid*) or the identification of the user's attached display support task (*devic*) when the message is forwarded to a single user.

#### **ALL**

forwards the message to all PVM users whose sessions originate, terminate, or pass through this PVM.

#### *message*

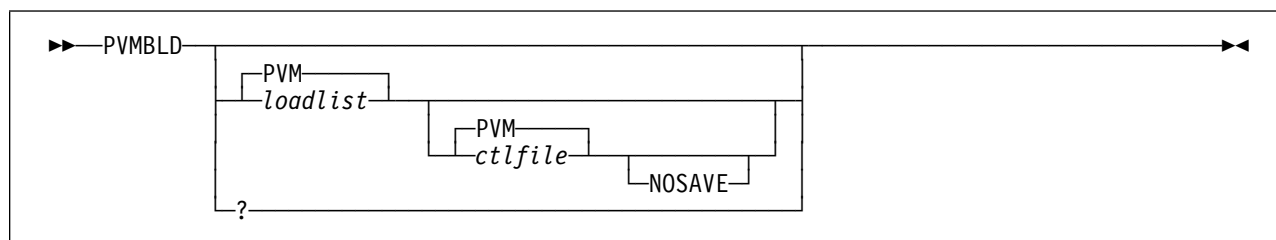
is the text of the message forwarded. The message can be of any format and characters acceptable to the CP command processor. The total length is limited to the CP command input limit.

### Usage Notes

1. The **ALL** option of this command is restricted to the PVM console operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.



## PVMBLD



## Purpose

Use the PVMBLD command when regenerating the PVM module.

## Operands

### *loadlist*

specifies the name of an exec file that contains the names of object modules in the order in which they are to reside in the resulting module. The default value is **PVM**. The module created has the same file name as the loadlist.

The command supplied on the distribution tape may be modified to customize it for the installation.

### *ctlfile*

specifies the file name of the control file (file type CNTRL). The default value is **PVM**.

### **NOSAVE**

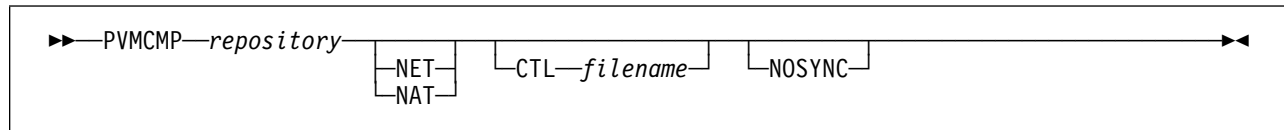
specifies to replace the original files with the new files (filetype MODULE, MAP, and PRELMAP). The default is to save the current files by prefixing the file names of the original files with an *O* and saving them prior to creating new files.

? displays the valid parameter format at the user's display station.

## Usage Notes

1. This command is intended for use by the PVM system programmer.

## PVMCMP



## Purpose

Use the PVMCMP command to compile a PVM message repository.

## Operands

### *repository*

The file name of the repository that is to be compiled. *DVMlangid* is used for PVM and PVMG messages. *EFGlangid* is used for DVMUSI and MPVM messages.

### **NET**

Indicates that the repository is to be compiled for use as the backup language. NET is the default when the repository is for the UCENG language.

### **NAT**

Indicates that the repository is to be compiled for use as a national language. NAT is the default when the repository is for a language other than UCENG.

### **CTL** *filename*

Indicates that the repository is to be updated before it is compiled using the CMS UPDATE command and the control file that is identified by *filename*. (The control file must have a file type of CNTRL.)

### **NOSYNC**

Is used primarily by IBM Translation Centers. NOSYNC forces message text expansion rules for English repositories to allow for translation. \$LISTING and \$COMP*nnn* files are produced, but the PVMDEST FILE is not read and object and \$SYNC*nnn* files are not produced.

## Usage Notes

1. This command is intended for use by the PVM system programmer.
2. Files created by this command include:
  - A repository object code file.  
 If the CTL option is not used, this file is named *DVMlangid* TEXT or *EFGlangid* TEXT. If the CTL option is used, then PVMCMP uses the update level identifier in the CNTRL file to name the file type of the output file. If the update level identifier of the last located update file is anything other than TEXT, the update level identifier is prefixed with the characters TXT to form the file type. If no update files are found the file type is named TEXT.
  - A listing file named *DVMlangid* \$LISTING or *EFGlangid* \$LISTING.  
 This file contains an input definitions plus any syntax error messages.

- A comparison file named *DVMIlangid \$COMPnnn* or *EFGlangid \$COMPnnn*, where *nnn* is the highest compatibility level for which items were defined in the repository with a PVM release.

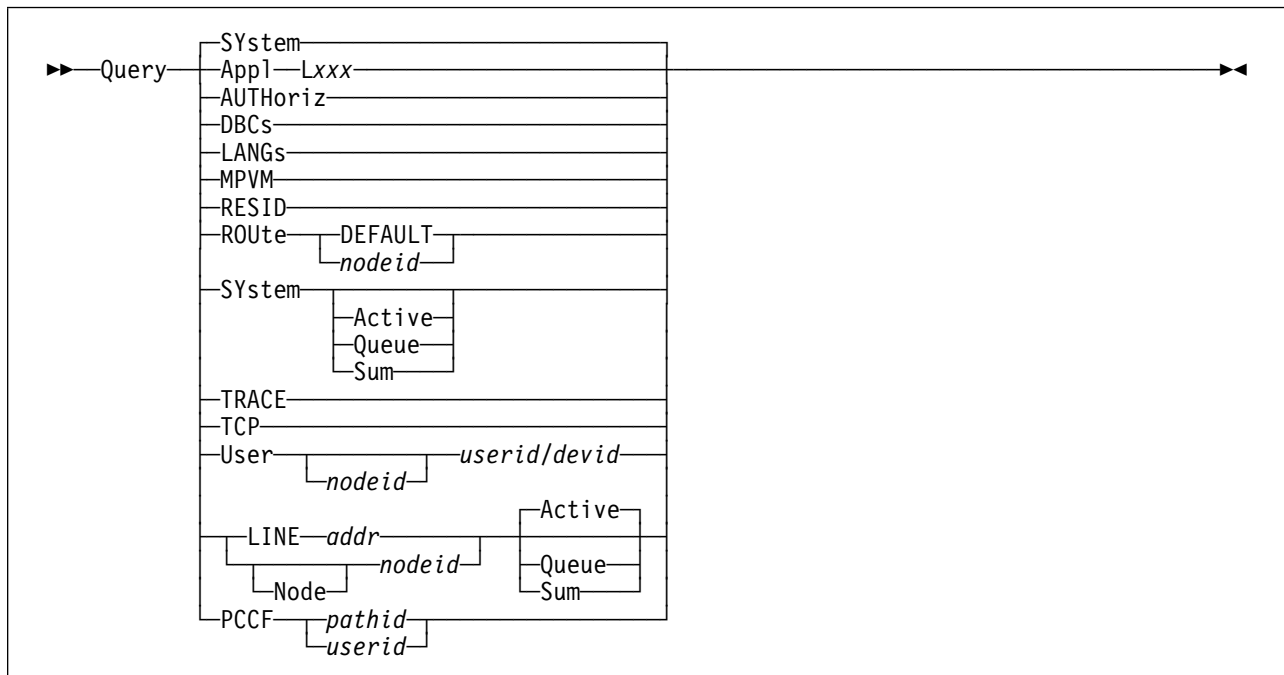
This file contains all of the nontranslatable information in the repository, not counting information added with a SERVICE control line. By comparing it to the corresponding, upper case English comparison file (supplied by IBM), it verifies that none of the nontranslatable information in the repository has been changed.

- When compiling an uppercase English repository file (UCENG), the comparison file for that repository must not be accessed in file mode A.
- Except when the NOSYNC option is specified, PVMCMP produces a file named *DVMIlangid \$SYNCnnn* or *EFGlangid \$SYNCnnn*, where *nnn* is the highest compatibility level encountered in the repository.

This file verifies that the object files are synchronized with respect to their common messages. The repositories are synchronized if *DVMIlangid \$SYNCnnn* and *EFGlangid \$SYNCnnn* are identical.

3. If the CTL option is specified, PVMCMP uses the CMS UPDATE command to process any update files for the message repository and PVMDEST FILE. PVMCMP does not erase any files created by CMS UPDATE.

## QUERY



## Purpose

Use the QUERY command to display one of the following:

- Users authorized to issue restricted PVM commands
- System status
- Current status of each line supported by PVM
- Current status of a logical device

## QUERY Command

- Contents of the routing table
- Current value accumulated in the error counter for a specified line
- User session information
- Current DBCS settings
- National languages enabled on a PVM system
- A specific PCCF path
- All PCCF paths for a specific virtual machine
- Current settings and statistics for MPVM users
- APPC resource identifier defined in the PVM configuration file
- Current TRACE settings.
- TCP/IP related information

QUERY command with no operands defaults to QUERY SYSTEM.

## Operands

### **Appl Lxxx**

displays the status of the session associated with the local logical device Lxxx.  
The value for xxx is a hexadecimal number between X'000' and X'FFF'.

### **AUTHoriz**

displays the current list of authorized users in the system who can enter PVM restricted commands.

### **DBCs**

displays the DBCS settings for CP operator, the PVM console (and the AUDIT file), and other VM users when the default language for the PVM system is a DBCS language.

### **LANGs**

displays the language identifiers of the languages for which PVM is enabled.  
LANGS also displays the languages used as the default national language and as the backup language at the local node.

### **MPVM**

displays the current number of MPVM users, the highest number of MPVM users, the current total number of MPVM sessions, and the highest total number of MPVM sessions. Also displayed are the current settings for maximum idle, maximum suspend and temporary disconnect times, maximum number of sessions per user, and maximum number of MPVM users allowed in the PVM server.

### **RESID**

displays the unique resource identifier defined in the PVM configuration file.

### **ROUte**

displays the contents of the local PVM's routing table for either the DEFAULT route or the particular route name.

#### **DEFAULT**

indicates if there is DEFAULT routing. If DEFAULT routing exists, the node being routed to is displayed.

#### **nodeid**

The node's entry from the routing table is displayed.

### **SYstem**

displays information about the status of the system. Without other operands, the status of each link defined within the system is displayed. Query SYstem is the default value if the QUERY command has no operands.

### **Active**

displays the status of each active link only. If there are no active links, only the name of the local system is displayed.

### **Queue**

displays the status and the associated user sessions for each active link.

### **Sum**

displays the statistics of each line driver. This includes both transaction counts and any line error statistics.

### **TRACE**

displays trace activities of PVM.

### **TCP**

display the defined virtual machine running VM TCP/IP, the defined TCP/IP port number of the local PVM machine, and the status of the redirector task. The status is displayed as one of the following:

#### **DOWN**

The PVM TCP/IP redirector task has not started, or it has ended because it encountered an error.

#### **WAITING**

The redirector task is in the process of connecting with VM TCP/IP.

#### **CONNECT**

The redirector task has connected with VM TCP/IP and is currently sending the initial data string.

#### **ISEND**

The initial data string has been sent and the redirector task is currently issuing a SOCKET socket request to VM TCP/IP.

#### **SOCKET**

The SOCKET request has been accepted by VM TCP/IP and the redirector task is currently issuing a BIND socket request to VM TCP/IP.

#### **BOUND**

VM TCP/IP has accepted the BIND request and the redirector task is currently issuing a LISTEN socket request to VM TCP/IP.

#### **LISTEN**

VM TCP/IP has accepted the LISTEN socket request. The redirector task is either blocked on an ACCEPT socket request to VM TCP/IP, or is processing a remote node's connect request.

### **User** *userid/devicid*

displays information about the associated local user (*userid*) or the attached display (*devicid*). The specified user must be connected to the local PVM server. The response includes the user's real display station address, type, and model number, and the session node and ID. \*SESnnnn can be used to get information on a particular MPVM session. The number of sessions, with information on each session, will be displayed when querying an MPVM user.

## QUIESCE Command

### **User** *nodeid userid/devic*

displays information about the associated session identified by *nodeid* and *userid/devic*. The specified session must be known to the local PVM. The response includes the user's real display station address, type, and model number, along with the session origin and destination IDs.

### **LINE** *addr*

displays the status of the specified link.

### **Node** *nodeid*

displays the status of the specified node.

### **Active**

displays the status of the specified line, if active. Active is the default if Queue or Sum is not specified.

### **Queue**

displays the status of the specified node and the associated user sessions for the specified node.

### **Sum**

displays the statistics of the line driver. This includes both transaction counts and any line error statistics.

### **PCCF** *pathid*

displays the PCCF path identified by the PCCF path number specified by *pathid*.

### **PCCF** *userid*

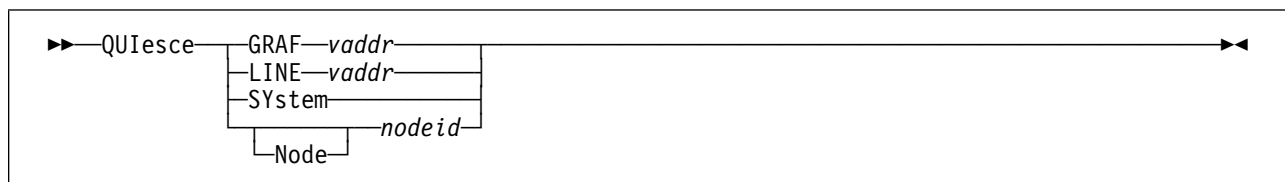
displays all PCCF paths associated with the virtual machine identified by *userid*.

## Usage Notes

1. The AUTHorize operand is restricted and must be issued from a user ID on the local system.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

---

## QUIESCE



## Purpose

The PVM operator uses the QUIESCE command to quiesce:

- A line driver task
- The PVM system.

*Quiesce* is a term used to describe the state of a task (or PVM itself). A task in a quiesce state does not allow any new sessions and terminates when all active sessions have ended.

The effect of the QUIESCE command can be reversed by issuing the RESUME command. Quiesce does not terminate any line drivers or attached display support tasks that have active sessions.

## Operands

### **GRAF** *vaddr*

indicates that the attached display support task *vaddr* is quiesced. A user currently dialed onto PVM is allowed to complete the session. At session termination the user is dropped, and the specified attached display support task terminates. If no user session is active, the specified task terminates immediately.

The *vaddr* specifies the virtual address of the attached display support task. The address specified must be the same as the address used to start the corresponding attached display support task (START GRAF nnn).

### **LINE** *vaddr*

indicates that the line *vaddr* is quiesced. After issuing this command, no new users are allowed to initiate sessions through the associated *vaddr*. When the last user session on the line terminates, the line driver terminates. If no user session is active, the specified line terminates immediately.

The *vaddr* specifies the address of the line. The address specified must be the same as the address used to start the corresponding line.

### **SYstem**

places the entire PVM system in a quiesced state. No new sessions can be initiated within or through this PVM system after issuing SYSTEM. This command does not terminate any line drivers, attached display support tasks, or user sessions.

### **Node** *nodeid*

indicates that link *nodeid* is quiesced. After the command is issued, no new users are allowed to initiate sessions through the associated *nodeid*. When the last user session on the line terminates, the line driver terminates. If no user session is active, the specified *nodeid* terminates immediately.

The *nodeid* must be the same name defined in the configuration file for the corresponding node.

### Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.



## QUIT

►►—QUIT—◄◄

### Purpose

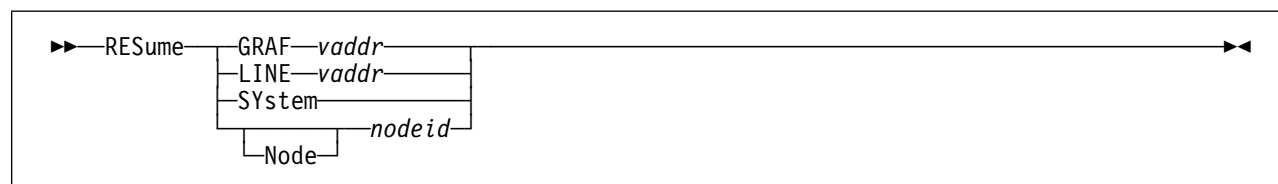
Use the QUIT command to terminate the PVM system immediately. No users are notified that the system is terminating. Not all allocated storage is freed when control returns to CMS. So CMS should be reinitialized, or an HX command should be issued to free allocated storage.

Before issuing the QUIT command, issue a DROP command for each active line or attached display support task so that all sessions are terminated.

### Usage Notes

1. This command is restricted to a PVM operator logged on to the PVM user ID.

## RESUME



### Purpose

Use the RESUME command to reverse the effect of a previously issued QUIESCE command. As a result, the specified line, attached display/printer support task, or the entire system is made available.

**Note:** If the quiesced line driver or attached display support task has already terminated due to the QUIESCE command, use the START command instead of the RESUME command.

### Operands

#### GRAF *vaddr*

indicates that the attached display support task *vaddr* is not terminated when the current user's session ends. The *vaddr* specifies the address of the affected attached display support task. The address specified must be the same as the address used to start the corresponding attached display support task (START GRAF *vaddr*).

#### LINE *vaddr*

indicates that use of line *vaddr* is resumed. Following the issuing of this command, new users are allowed on the affected line. The previously issued QUIESCE command does not terminate the line. The *vaddr* specifies the address of the affected line driver. The address specified must be the same as the address used to start the corresponding line.

## ROUTE Command

### SYstem

indicates that the entire PVM system is to be resumed. New sessions may be initiated after this command is issued.

### Node *nodeid*

indicates that the link whose node name is *nodeid* is resumed. After the command is issued, new users are allowed to initiate sessions on the affected line.

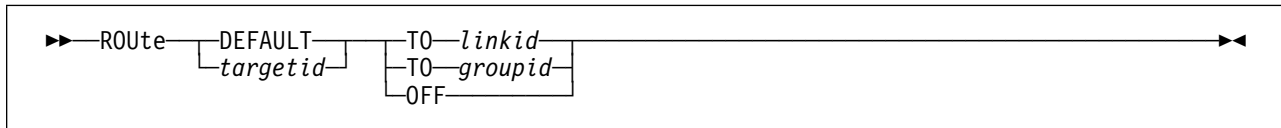
The *nodeid* must be the name defined in the configuration file for the corresponding line.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

---

## ROUTE



## Purpose

The ROUTE command modifies the in-storage routing table used by PVM for session routing. Node names may be added, deleted, or their routing altered by the command.

## Operands

### DEFAULT

changes the DEFAULT routing.

### *targetid*

is the target node added, deleted, or modified.

### TO *linkid*

indicates that the target node is routed through the link *linkid*. *linkid* is the node name of the link. This node must be a PVM node directly connected to the local PVM system.

### TO *groupid*

indicates that the target node is routed through the link group *groupid*. The *groupid* is the name of the link group. This group must be defined for the local PVM system.

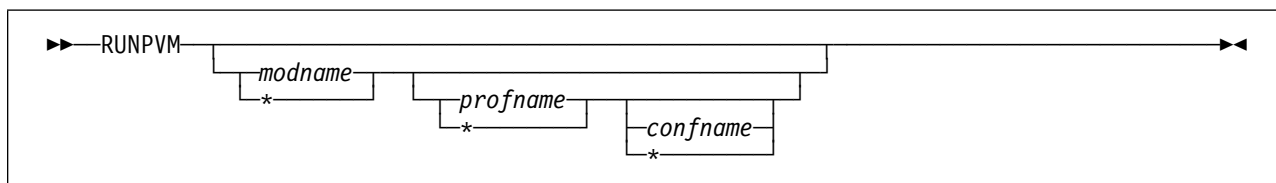
### OFF

removes the specified target node from the routing table. OFF may not be specified as a default route.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

## RUNPVM



## Purpose

Use the RUNPVM command to load and start PVM.

## Operands

### *modname*

is the name of the PVM module loaded. Specifying \* uses the default file name is PVM. The file type is MODULE.

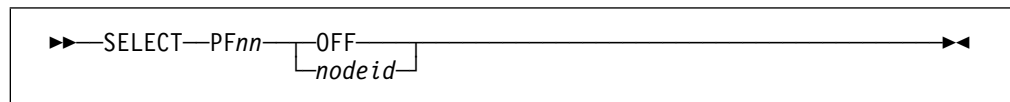
### *profname*

is the name of the PVM exec executed at startup time. Specifying \* uses the default file name, *profname*, is PROFILE. The file type of this exec is PVM.

### *confname*

is the name of the configuration file. Specifying \* uses the default file name, *confname*, is PVM. The file type of this configuration file is CONFIG.

## SELECT



### Purpose

Use the SELECT command to dynamically add, delete, or modify the nodes assigned to PF keys by the SELECT records of the configuration file.

### Operands

#### PFnn

is the PF key modified. The value of *nn* ranges from 1 to 12, excluding 8.

#### OFF

indicates that the PF key definition is to be reset.

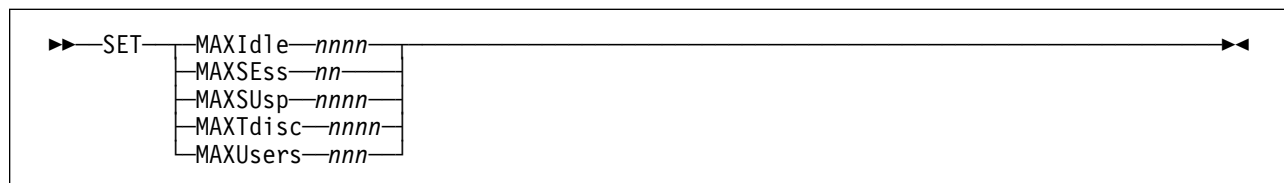
#### nodeid

is the name of the node selected by the specified PF key. The node name should not be a member of a group. It can identify a group, but cannot identify an individual member of a group.

### Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

## SET



### Purpose

Use the SET command to change settings for MPVM users. These settings include the maximum:

- Time a MPVM user can remain idle
- Time a MPVM can have a session suspended
- Time a MPVM user can be temporarily disconnected
- Number of MPVM users allowed on the PVM server
- Number of sessions an MPVM user can have.

## Operands

### **MAXIdle** *nnnn*

The maximum number of seconds a MPVM user may remain idle before all of that user's sessions will be terminated by PVM. (Idle means there is no communication from the MPVM user to PVM for any session.) The *nnnn* ranges from 60 to 9998. A setting of 9999 will mean that there is no maximum idle time limit. PVM sends out a warning message when approximately three-fourths of the time interval has elapsed.

### **MAXSEss** *nn*

The maximum number of sessions a MPVM user may establish. The *nn* ranges from 1 to 99. The maximum number of sessions allowed for all MPVM users combined is 4096.

Decreasing the maximum number of sessions allowed below the current number of sessions owned by a particular MPVM user is not going to affect those existing sessions; the user would not be able to start any more sessions.

### **MAXSUsP** *nnnn*

The maximum number of seconds a MPVM user may have a session suspended before that user's session will be terminated by PVM. The value of *nnnn* ranges from 60 to 9998. A setting of 9999 will mean no maximum suspend time limit. PVM will send out a warning message when approximately one-half of the time limit has elapsed, and then again when three-fourths of the time limit has passed.

### **MAXTdisc** *nnnn*

The maximum number of seconds a MPVM user may remain temporarily disconnected before all of the user's sessions will be terminated by PVM. The *nnnn* ranges from 60 to 9998. A setting of 9999 will mean no maximum temporary disconnect time limit. A setting of 0 will indicate that MPVM users may not temporarily disconnect. PVM will send out a warning message when approximately three-fourths of the time limit has elapsed.

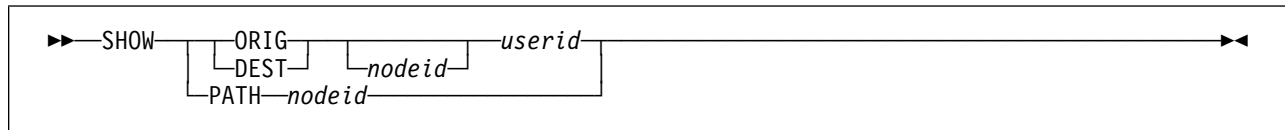
### **MAXUsers** *nnn*

The maximum number of MPVM users who may currently connect to PVM. The *nnn* ranges from 1 to 999. The maximum number of users can also be limited indirectly through the number of IUCV and APPC/VM paths supported by PVM. This value can be changed through the MAXCONN record in the configuration file but cannot exceed the limit specified in the CP OPTION directory record for the PVM virtual machine.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

## SHOW



## Purpose

Use the `SHOW` command to show the path of a particular session, or the path to a particular node. The system on which the command is first executed locates the session and displays information indicating that it is part of the path. This message displays the link upon which the session comes in, and the link upon which the session goes out.

## Operands

**ORIG**

traces the path to the origin. If ORIG is specified and the origin is not the local system or the incoming link, the command is forwarded to the node connected to the incoming link for processing.

## DEST

traces the path to the destination. If DEST is specified and the destination is not the local system or the outgoing link, the command is forwarded to the node connected to the outgoing link for processing. In this way, the command is executed on each intermediate node along the path.

## PATH

shows the PVM routing to get to a particular target node.

## nodeid

is the node name of the origin or destination of the session. For ORIG or DEST, if no value is specified for *nodeid*; the local node name is assumed. For PATH, a value for *nodeid* must be specified, and it is the target node for which path information is required.

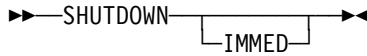
## userid

is the PVM identification for the origin or destination of the session.

## Usage Notes

1. **SHOW PATH** shows the PVM routing to get to a specific target node. The command finds the outbound path to the target node. If the target node is the local node or a non-network node, messages identify the path to the target node, and then the message **COMMAND COMPLETE**. If the target node is not the local node or a non-network node, the response is Path to node xxxxx is through node yyyy. The **SHOW** command is then forwarded to the node yyyy for execution.
2. When issued from another virtual machine using **SMSG**, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

## SHUTDOWN



### Purpose

Use the SHUTDOWN command to conditionally terminate the PVM server. If the PVM server has no active sessions, then PVM will shut down immediately. However, if the PVM server determines that there are active sessions, it will not shut down until all active sessions end. All line drivers and attached display support tasks are placed in a quiesced state. The attached display support tasks terminate as soon as the user session terminates. Line drivers terminate when the last users of the associated lines terminate their sessions. PVM terminates when the last user session terminates and all associated tasks terminate with it unless the IMMEDIATE option is specified. The operator can then wait for the sessions to terminate, or issue the SHUTDOWN IMMEDIATE command to shut down immediately.

During SHUTDOWN processing, an MPVM user task will be removed from the PVM virtual machine if there are no sessions for that user. If a MPVM user has any active sessions, these sessions are left active, but no new sessions are allowed. When the MPVM users end all sessions, their user tasks are removed from the PVM machine to allow PVM to shutdown.

Not all allocated storage is freed when control returns to CMS. Therefore, CMS should be reinitialized or an HX command should be issued to free allocated storage.

### Operands

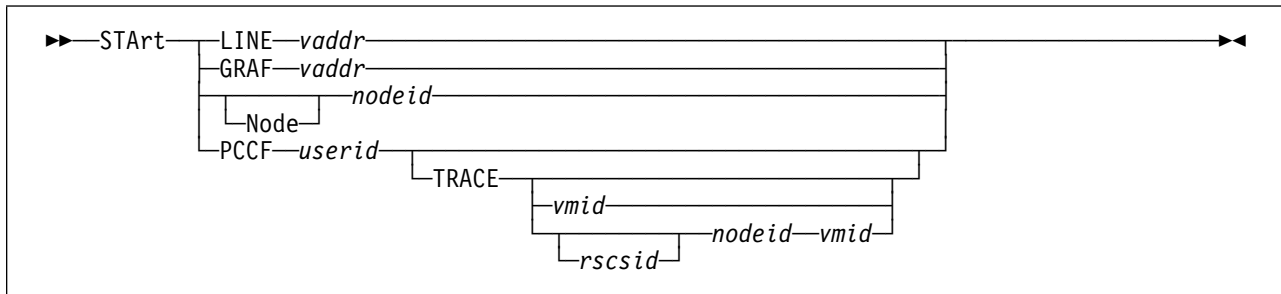
#### IMMED

indicates that the shutdown is immediate and all sessions are terminated.

### Usage Notes

1. This command is restricted to a PVM operator, logged on with a PVM user ID.
2. For SHUTDOWN without the immediate option if:
  - An established session has had no I/O activity for a two minute interval, the session is terminated.
3. For SHUTDOWN IMMEDIATE, sessions are terminated immediately.

## START



## Purpose

The PVM operator uses the START command (a restricted command) to start:

- A line driver task to a PVM virtual machine
- An IUCV line driver task to a PVMG virtual machine
- An attached display support task
- A PCCF support task

## Operands

### LINE *vaddr*

starts a line driver at line *vaddr*. The address must be defined within the configuration file LINK record. To start a line to a remote PVM connection using a FICON channel, the address must be defined with a link type of 3088 in the configuration LINK record. The *vaddr* specifies the virtual address of the affected line driver.

### GRAF *vaddr*

starts an attached display support task at virtual address *vaddr*. This address must have a value from X'20' to X'FFF'. The *vaddr* specifies the virtual address of the affected attached display support task.

### Node *nodeid*

starts a line driver to the line whose node name is *nodeid*. The *nodeid* must be defined in the configuration file.

### PCCF *userid*

starts a PCCF support task. The *userid* is the virtual machine for which PVM is to handle PCCF connections.

### TRACE

Indicates that tracing for the PCCF support task is to begin immediately.

### *vmid*

issues a CP SPOOL command to transfer output to the reader (*vmid*).

### *rscsid nodeid vmid*

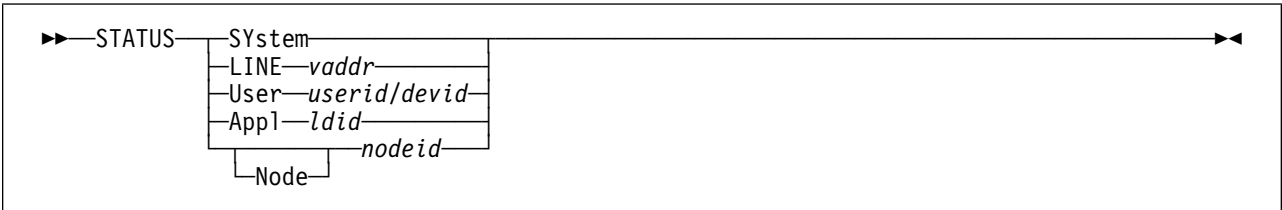
issues a CP SPOOL command and a CP TAG command to send the output over the RSCS network (*rscsid*) to the reader (*vmid*) at the specified node (*nodeid*).



Usage Notes

- 1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
- 2. The START GRAF command must be issued for each device being made available for users to DIAL.
- 3. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

STATUS



Purpose

Use the STATUS command to determine the status of the system pools or the status of tasks within the system. The output from the STATUS SYSTEM can be used to determine if the sizes of the system pools are correct. The other options can be used for problem determination.

Operands

- SYstem**  
displays information about the system Post Queue Element pool and buffer pool. This information gives the allocated size, peak usage, and current usage of each pool. To adjust these allocations, change the value specified on the BUFFERS record in the configuration file.
- LINE vaddr**  
displays information concerning the associated line driver. This information includes the DVMTCB control block address, current task status, task flags, last (or current) CAW address, next trace entry address in the wrap trace table, and the CCW pointed to by the CAW. *vaddr* specifies the address of the line driver.
- User userid/devic**  
displays information concerning the task associated with the specified ID. This information includes the DVMTCB control block address, current task status, and task flags.  
  
The *userid/devic* identifies the user (*userid*) and the attached display (*devic*). The subject ID must be on the local system.
- Appl ldid**  
displays information concerning the logical device support task associated with the specified ID. This information includes the DVMTCB control block address, current task status, and task flags.

## STATUS Command

The *ldid* specifies the address of the logical device support task. The address should be specified in the form *Lnnn*.

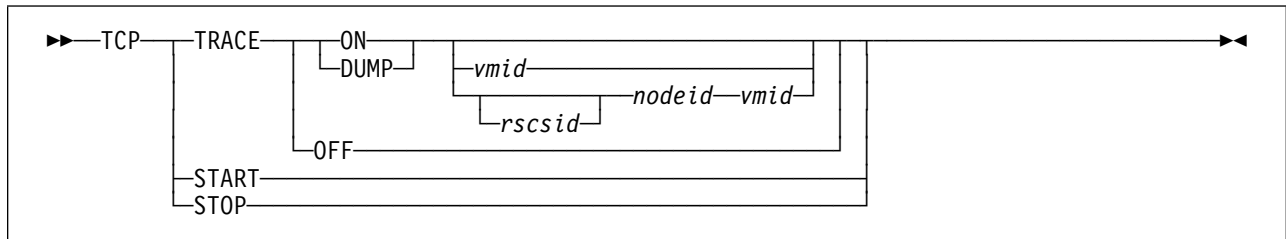
### **Node** *nodeid*

displays information specifies that information concerning the associated line driver whose node is *nodeid*. This information includes the DVMTCB control block address, current task status, task flags, last (or current) CAW address, next trace entry address in the wrap trace table, and the CCW pointed to by the CAW.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

## TCP



## Purpose

Use the TCP command to start or stop the TCP/IP redirector task and to control the tracing facility of the TCP/IP redirector task's connection with VM TCP/IP.

## Operands

### TRACE

starts or stops the redirector task tracing facility

### ON

starts the trace

### OFF

stops the trace

### DUMP

causes the redirector task to dump it's control blocks and trace tables

*rscsid nodeid vmid*

issues a CP SPOOL command and a CP TAG command to send the output over the RSCS network (*rscsid*) to the reader (*vmid*) at the specified node (*nodeid*). If *rscsid* is omitted, it defaults to RSCS.

### START

starts the TCP/IP redirector task.

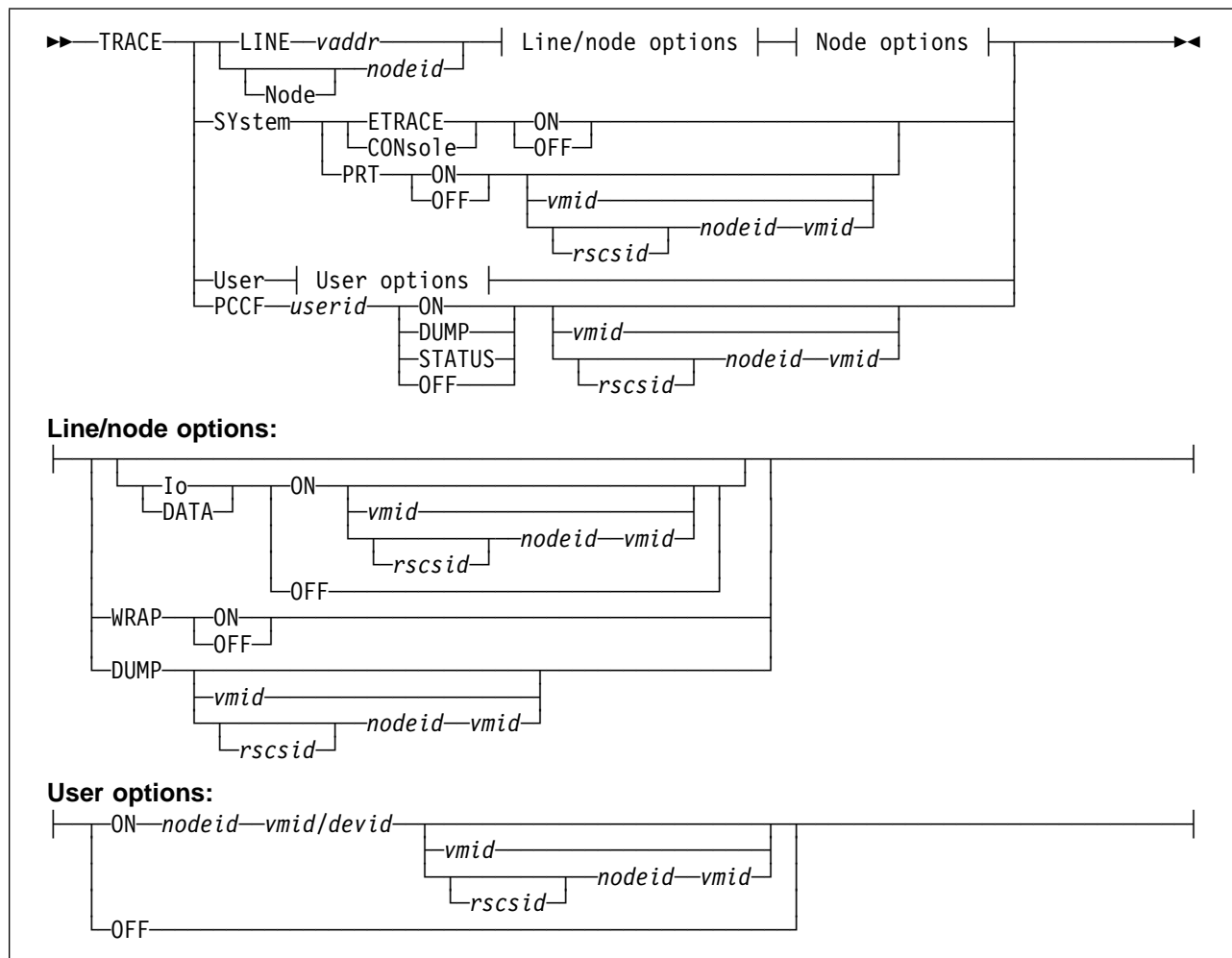
### STOP

stops the TCP/IP redirector task.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. The TRACE operand is initially set to OFF.
3. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

# TRACE



## Purpose

Use the TRACE command to start or stop the line tracing facility, the PVM multitasking supervisor trace facility, or the tracing of a PCCF support task.

## Operands

**LINE** *vaddr*

specifies the virtual line address (*vaddr*) associated with the PVM line for which tracing is requested. The line address must be a valid address as specified by a LINK record or a DEFINE command.

Node *nodeid*

specifies the node name (*nodeid*) associated with the PVM line for which tracing is requested. The node name specified must be a valid node name as specified by a LINK record or a DEFINE command.

**lo** initiates a trace of line driver I/O activity for the specified line driver.

## DATA

initiates a buffer trace for the specified line driver. Output goes to a dynamically defined virtual printer.

**WRAP**

causes wraparound I/O tracing for an attached line driver task. Output is stored in a buffer area managed by the affected task.

**DUMP**

causes the associated task to dump the local task control blocks and the internal I/O trace table.

**SYstem**

causes a trace for the PVM multitasking supervisor. Specifying OFF terminates only the specified system tracing option. Other system trace options are not terminated.

**ETRACE**

record the output in a wrap buffer in memory. The trace data may be accessed by taking a VMDUMP of the PVM guest and examining it using VM Dump Tool.

**Note:****CONsole**

writes the output to the PVM virtual console at virtual address 009.

**PRT**

writes the output to the virtual printer address 00F.

**User**

begins or ends tracing for a specific PVM session. When tracing is active, each task within PVM that is processing session data checks to see if the current piece of data is associated with the session to be traced. If it is, the CP DUMP command dumps the data block. The trace record contains the module name of the task producing the record, the task name, and the time stamp.

**ON** *nodeid vmid/devid*

specifies the identification of the node (*nodeid*) and PVM user (*vmid*) or attached display (*devid*) of the session to be traced. The session need not be currently active. Because a given session has both an origin node and user ID, and a destination node and user ID, either pair can be specified in the TRACE command.

If the *vmid* is the user ID of a MPVM user, the data for all of the user's sessions will be traced if the *nodeid* is the origin node ID of the MPVM user. If the *vmid* is a PVM user ID for a single session in the form of *\*sesnnnn*, only data for that particular session is traced.

**ON**

initiates the trace.

**OFF**

terminates the trace.

**PCCF** *userid*

specifies the PCCF support task. The *userid* is the virtual machine for which PVM is handling PCCF connections.

**STATUS**

provides PCCF path information.

*vmid*

issues a CP SPOOL command to transfer output to the reader (*vmid*).

## VARY Command

*rscsid nodeid vmid*

issues a CP SPOOL command and a CP TAG command to send the output over the RSCS network (*rscsid*) to the reader (*vmid*) at the specified node (*nodeid*).

For TRACE LINE, TRACE NODE, and TRACE USER, if *rscsid* is omitted, it defaults to RSCS.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. All TRACE operands are initially set off.
3. The following table lists the file naming conventions for trace output files.

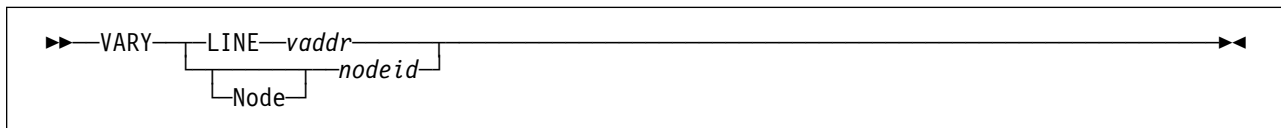
TRACE Command Options	Filename	Filetype
IO	<i>link_name</i>	IOTRACE
DATA	<i>link_name</i>	DATATRAC
DUMP	<i>link_name</i>	TASKDUMP
SYstem PRT	<i>local_node_ID</i>	SUPTRACE
User	<i>local_node_ID</i>	SESSTRAC

The file naming convention for trace output with the WRAP option, SYSTEM ETRACE option and SYSTEM CONSOLE option is not applicable.

4. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

---

## VARY



## Purpose

Use the VARY command to alter the availability status of links and devices.

## Operands

**LINE** *vaddr*

identifies that line *vaddr* is varied on-line or off-line.

**Node** *nodeid*

identifies that node *nodeid* is varied on-line or off-line.

## Usage Notes

1. This command is restricted to the PVM operator or an alternate operator specified on an AUTHORIZ record in the PVM configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVM how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

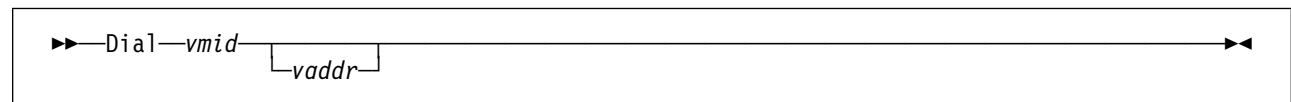
---

## CP Commands Used with PVM

This section describes CP commands that interact with PVM.

---

### DIAL



### Purpose

Use the CP DIAL command to access PVM directly from a VM logon screen. The PROFILE PVM or an authorized user must issue a START GRAF command for each device being made available for users to DIAL.

### Operands

*vmid*

is the virtual machine identification of the PVM user ID.

*vaddr*

is the virtual display device address.

---

## SMSG

►—SMsg—*vmid*—└─(*prefix*)─┘—*command*—►

### Purpose

Use the CP SMSG command to issue PVM commands to PVM from another virtual machine.

### Operands

*vmid*

is the virtual machine identification of the PVM user ID.

(*prefix*)

is any valid PVM command prefix. See “Using Command Prefixes” on page 149 for more information about command prefixes.

*command*

is any valid PVM command.

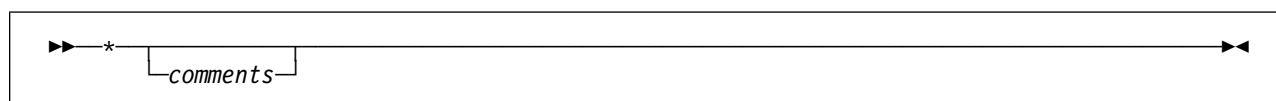


## Appendix C. PVM Configuration File Records

PVM configuration files contain fixed-length, 80-character records. Input must be in uppercase and is restricted to columns 1 through 71 with no continuation. Entries in a configuration file record must be separated by one or more blanks.

For more information on using PVM configuration file records, see “Creating the Configuration File” on page 28, and “Storage Considerations” on page 15.

### \* (Comment)



### Purpose

Use the \* record to imbed comments in the configuration file. Place this record anywhere in the file. The record is ignored during configuration file processing.

### Operands

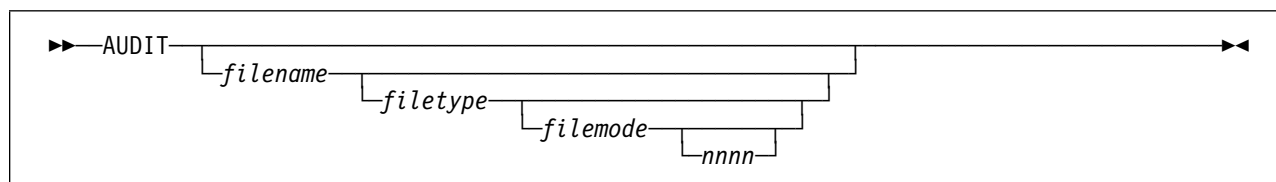
*comments*

is any text message. This information, plus at least one blank, must follow the asterisk (\*).

### Usage Notes

1. The Comment record must begin in column one of the PVMG configuration file.
2. There can be no blank lines in the PVM configuration file. To insert lines between records, use this record.

## AUDIT



### Purpose

Use the AUDIT record to indicate that console data is to be stored in a specified CMS file. This file is created the first time data is written to it. If the Message Service Facility is enabled, messages resulting from CMS commands or commands submitted directly to CP are stored. Data is stored chronologically, along with the current date and time. When the audit file is full, it wraps to the beginning of the file, overlaying the existing records.

## AUDIT Record

The audit file is a fixed length file, with the first record containing the record number of the next record to be written. The length of the AUDIT file depends on if the audit records contain message numbers and/or time stamps. The length of the AUDIT file is:

- 129 bytes if the audit record contains neither time stamps nor message numbers
- 140 bytes if the audit record does not contain time stamps but does contain message numbers
- 129 bytes + *maximum timestamp length* + 1, if the audit record contains a time stamp but does not contain message numbers
- 140 bytes + *maximum timestamp length* + 1, if the audit record contains a time stamp and message numbers.

The + 1 allows for a single blank following the timestamp.

*Maximum timestamp length* refers to the longer of the AM or the PM timestamp for the default language. However, if the default language is a DBCS language, the *maximum timestamp length* is the longest of the:

- AM or PM timestamp for the default language
- AM or PM timestamp for the backup language.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If this record is omitted, no audit file is created. If there is more than one AUDIT record in the configuration file, only the last AUDIT record in the list is accepted. PVM ignores the other AUDIT records.

## Operands

### *filename*

is the file name of the CMS file containing the console data. If *filename* is not specified, the default value is CONSOLE.

### *filetype*

is the file type of the CMS file containing the console data. If *filetype* is not specified, the default value is LOG.

### *filemode*

is the file mode of the CMS file containing the console data. If *filemode* is not specified, the default value is A6.

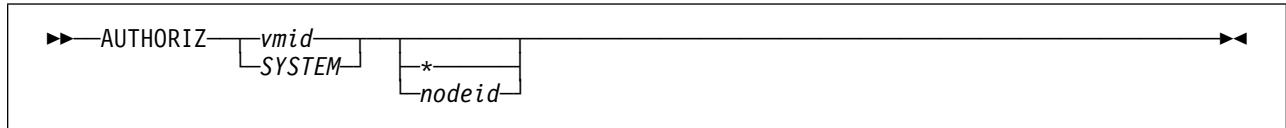
### *nnnn*

is the maximum size of the audit file in 1K records. The value of *nnnn* ranges from 1 to 1000, allowing a maximum of 1 million records. The default is 999,999 records.

## Usage Notes

1. Specify a maximum size for the audit file that prevents the PVM minidisk from getting full.

## AUTHORIZ



### Purpose

Use the AUTHORIZ record to identify users who are authorized to issue the restricted subset of PVM commands and the nodes from which these commands may be issued.

All LOCAL, LINK, GROUP, or ROUTE records must precede this record.

### Operands

*vmid*

is the identification of a VM user (or virtual machine) authorized to issue restricted PVM commands. Without specifying \* or *nodeid*, this authorization is for the local node only.

#### SYSTEM

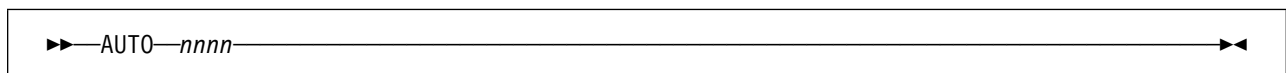
authorizes the PVM operator to issue commands to nodes (\* or *nodeid*) other than the local node.

\* authorizes the user to issue restricted PVM commands *from any node*.

*nodeid*

is a specific node where *vmid* is authorized to issue restricted PVM commands.

## AUTO



### Purpose

#### GUIP

Use the AUTO record to indicate that PVM is to automatically execute the AUTO PVM exec if it exists. AUTO PVM is a customer written exec that can issue CP, CMS, or PVM commands to monitor activities such as status of links or nodes in the PVM network. PVM does not pass the AUTO PVM exec any inputs, nor does it take action on any passed return codes or data.

The exec is executed based on the time interval specified in the AUTO record. If the AUTO record is not specified, no automatic execution is performed.

#### GUIP end

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record.

### Operands

*nnnn*

is the number of minutes between invocations of the exec. The value of *nnnn* ranges from 1 to 1440.

---

## BUFFERS

►►—BUFFERS—*nnnn*—◄◄

### Purpose

Use the BUFFERS record to indicate the number of 4096-byte pages allocated for internal data buffers. PVM also uses the BUFFERS record to compute the size of the Post Queue Element (PQE) pool.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If this record is omitted, the default value is 30 pages. If there is more than one BUFFERS record in the configuration file, only the last BUFFERS record in the list is accepted. PVM ignores the other BUFFERS records.

### Operands

*nnnn*

is the number of pages to be allocated in the range 25 to 2048. The number of pages specified should be greater than half the maximum number of users. Specifying insufficient buffers can degrade system performance, or cause PVM to completely quiesce.

### Usage Notes

1. To calculate the *nnnn* value, use the following formulas:

---

When  $(\text{SMAX}/2 + 15)$  is greater than UMAX:

$$\text{BUFFERS} = 25 + 3(\text{NMAX}) + \text{CMAX} + \text{SMAX}/2$$

Or, when  $(\text{SMAX}/2 + 15)$  is less than or equal to UMAX:

$$\text{BUFFERS} = 10 + 3(\text{NMAX}) + \text{CMAX} + \text{UMAX}$$

---

#### NMAX

is the maximum number of active PVM to PVM network CTCA and APPC type line drivers, and PVM to PVMG IUCV line drivers.

#### CMAX

is the maximum number of active 3088 and TCPIP type line drivers times the average number of buffer pages per driver. The number of pages is specified by the BLK operand of the LINK configuration file record. If BLK is not specified, the default number of pages is 6 (3 for input and 3 for output).

**SMAX**

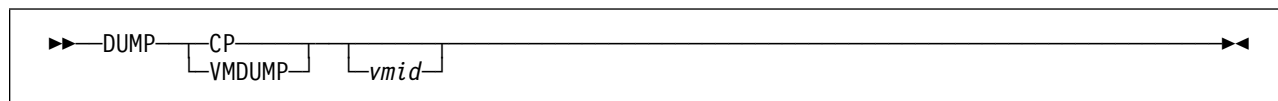
is the maximum number of active PVM user and PCCF sessions, plus the number of MPVM user sessions, active at any one time.

**UMAX**

is the maximum number of sessions simultaneously accessing the local CP system as a target node.

2. The calculation for BUFFERS is only an approximation. The PVM operator should monitor the usage of the buffer pool using the STATUS SYSTEM command. For larger systems, the TOTAL BUFFER PAGES displayed should exceed the MAX USED by 50 pages.

## DUMP



### Purpose

Use the DUMP record to indicate whether the dump taken during abnormal termination of PVM is a CP dump or a VMDUMP.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If the DUMP record is omitted, the default is CP-type dump. If more than one DUMP record in the configuration file, only the last DUMP record in the list is accepted. PVM ignores the other DUMP records.

### Operands

#### CP

indicates that on abnormal termination of PVM the dump is CP DUMP.

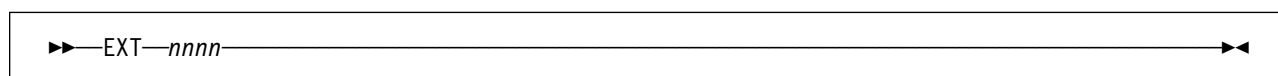
#### VMDUMP

indicates that on abnormal termination of PVM the dump is VMDUMP.

#### *vmid*

indicates where the dump is to be spooled. The default is written to the printer for a CP dump and spooled to the *vmid* of OPERATNS for a VMDUMP.

## EXT



### Purpose

Use the EXT record to specify the maximum number of entries allotted in the internal table that handles VMCF and logical device external interrupts and also interrupts on both IUCV and APPC/VM paths. An entry is made for a user accessing this PVM system from the CMS environment using the PASSTHRU command, and two entries are made for a user using the MPVM command. One entry is made for logical devices created for this PVM node. Add two entries for use by the system. Specify enough entries to handle the peak number of simultaneous CMS PASSTHRU and MPVM users, logical devices, and IUCV and APPC type links.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If this record is omitted, the default value is 50 entries. If more than one EXT record in the configuration file, only the last EXT record in the list is accepted. PVM ignores the other EXT records.

## Operands

nnnn

is the maximum number of entries created. The value of *nnnn* ranges from 10 to 4096.

## Usage Notes

1. To calculate the value for EXT use the following formula:

$$\text{EXT} = 3 + \text{VMAX} + 2(\text{AMAX}) + \text{UMAX} + \text{CMAX} + 2(\text{MPVM}) + \text{IMAX} + 2(\text{TCPIP}) + 1$$

**VMAX**

is the maximum number of users simultaneously accessing PVM by the PASSTHRU command.

**AMAX**

is the maximum number of active APPC/VM line drivers.

**UMAX**

is the maximum number of users simultaneously accessing the local CP system as a target node.

## CMAX

is the maximum number of simultaneous PCCF sessions.

## MPVM

is the maximum number of simultaneous MPVM users,

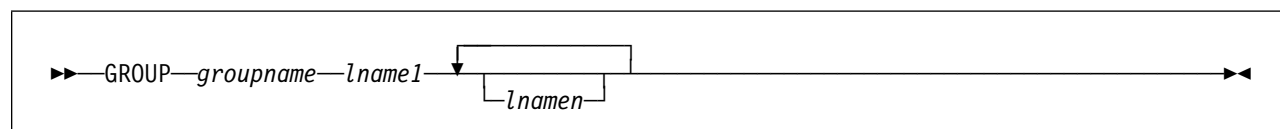
## IMAX

is the maximum number of IUCV line drivers going to PVMG virtual machine(s).

## TCPIP

is the maximum number of active TCPIP line drivers.

## GROUP



## Purpose

Use the GROUP record to define a set of links that form a line group. The group name is the node name of the adjacent PVM system the links are connected to. The same group name may appear on more than one group record. In this case, the line group consists of the sum of all the links specified on the multiple group records.

The GROUP records must be placed after all the LOCAL, and LINK records, but the GROUP records must precede any other records.

## I/O Record

The only types of links that can belong to a line group are CTCA, APPC, TCPIP, and 3088 links. An APPC or TCPIP link cannot appear in a GROUP with another APPC or TCPIP link. However, it may be combined with any of the other types of supported links.

The GROUP record must contain at least one link node name. In a group of two or more links, the link types do not have to be identical. When a link is defined as part of a group, the link node name **should not** appear on a SELECT record.

**Note:** It is **very** important that *groupname* specified in the GROUP record be a name that is unique throughout the **entire** PVM network. Because *groupname* is used as a node name on the PVM Selection Menu, the location of the site should be apparent by the name.

## Operands

*groupname*

is the node name of the line group.

*lname1*

is the link node name in the line group.

*lnamen*

is the link node name in the line group.

---

## I/O

►►—I/O—nnnn—►►

## Purpose

Use the I/O record to specify the maximum number of entries allotted in the PVM internal table that handles I/O devices. Use one entry for each virtual address that PVM performs I/O operations on. Therefore, there will be one entry for each active line driver (START LINE), one for each attached display support task (START GRAF), and one for the PVM user ID display station. Specify enough entries to handle the peak number of active line drivers and attached display support tasks.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If this record is omitted, the default value is 50. If more than one I/O record in the configuration file, only the last I/O record in the list is accepted. PVM ignores the other I/O records.

## Operands

*nnnn*

is the maximum number of I/O devices expected. The value of *nnnn* ranges from 10 to 2048.



## Usage Notes

1. To Calculate the value for I/O use the following formula:

---


$$I/O = 1 + LMAX + GMAX$$


---

where:

**LMAX**

is the maximum number of active line drivers. Add one extra for each active 3088 type line driver.

**GMAX**

is the maximum number of attached display support tasks.

---

## LANG

►►—LANG—*langid*—————►►

## Purpose

Use the LANG record to identify a particular language as the default national language for a node.

During PVM initialization, the backup language is considered the national language for a node before the LANG record is processed. It continues as the default national language for the node if the LANG configuration file record is not specified.

Place the LANG record anywhere within the configuration file; it takes effect immediately. The LANG record can be entered before the LOCAL record to establish the language used for any error messages issued in conjunction with subsequent configuration file processing.

## Operands

*langid*

is the identification of the selected language. It is a 1-to-5-character language identifier.

The IBM-provided languages are:

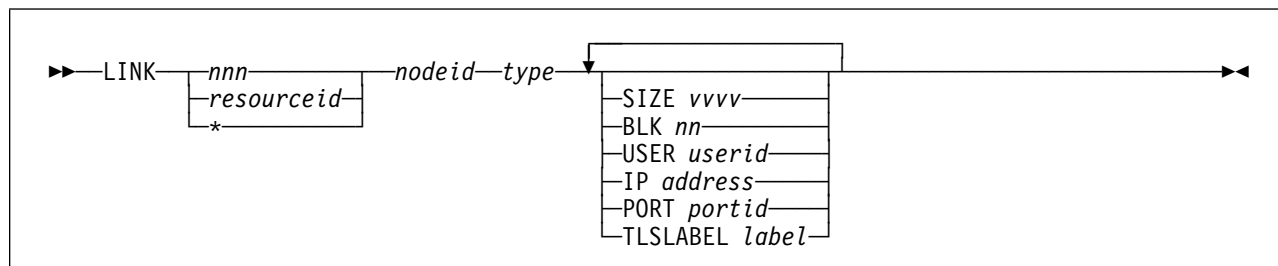
**AMENG** American English

**KANJI** Japanese

**UCENG** Uppercase English

**GER** German

## LINK



## Purpose

Use the LINK record to define the line address and line type (where applicable) for connecting PVM to a remote system, including PVMG.

The LOCAL record must precede this record. The order of LINK records in the configuration file should correspond to their frequency of use. The same node identifier cannot appear in more than one LINK record.

**Note:** It is **very** important that the *nodeid* specified in the LINK record be a name that is unique throughout the **entire** PVM network.

## Operands

*nnn*

is the line address. This address ranges from X'100' to X'FFF' for CTCA and 3088 links.

*resourceid*

is the name of the resource nickname to which to connect. The *resourceid* is a 1-to-8-character string. This value applies only to APPC type links. The *resourceid* refers to the nickname entry found in the CMS communications directory (UCOMDIR NAMES). PVM uses the *resourceid* to invoke a CMSIUCV RESOLVE function.

\* indicates that a line address is not provided. For a type of IUCV or TCPIP, this parameter is required.

*nodeid*

is the node name of the system on the remote end of this link. Node names are available to the user, through the PVM Selection Menu, for node selection at session initiation.

*type*

specifies the type of line driver used to drive the line. The valid types are:

**APPC** uses the APPC/VM protocol to communicate over the logical path.

**IUCV** uses the IUCV protocol to communicate with a PVMG virtual machine.

**3088** specifies that this link to another PVM is made by a FICON channel. The 3088 driver supports a consecutive EVEN/ODD pair of 3088 addresses. Because the 3088 driver uses virtual addresses instead of real addresses, the real addresses do not necessarily have to be consecutive. However, to eliminate confusion the real addresses should be consecutive too.

The address specified in the LINK record indicates the address of the OUTPUT link. It must be the EVEN address on one side of the link and the ODD address on the opposite side of the link. If the specified address is EVEN, PVM assumes the input address is one higher. If the address is ODD, PVM assumes the input address is one lower.

**CTCA** specifies that this link to another PVM is made by a channel-to-channel adapter.

**TCPIP** indicates that the link type to another PVM node is made over a TCP/IP network.

**SIZE** *vvvv*

indicates the *threshold block size* for this link. This value governs the minimum block size to be transmitted by network and IUCV line drivers when there is a queue of data to send. The threshold block size should be set at a value that is low enough to reduce interference between interactive and bulk, but large enough to efficiently transmit the data. Generally, the value should not be greater than the number of characters that can be transmitted across the link in one second.

The default value is 4096 for IUCV, CTCA, APPC, TCPIP, and 3088. The value of *vvvv* ranges from 200 to 4096.

**BLK** *nn*

indicates the number of 4K areas that are tied together for the output transmission buffer. The value of *nn* ranges from 1 to 8 for TCPIP links and 2 to 16 for 3088 type links. The default value is 3. BLK is only valid if the link type is 3088 or TCPIP. The BUFFERS value may have to be increased depending on the number of 3088 or TCPIP links and the BLK values. PVM allocates from the buffer pool the number of pages specified in the BLK record on each side for every 3088 or TCPIP link. Therefore, the default is 6 pages for each 3088 or TCPIP link that is started. The 3088 or TCPIP line driver sends handshaking information to the other side that specifies an output size based on the BLK value and a maximum input size of 64K. When it reads the handshake from the other side, it adjusts the input size to fit the opposite sides output size. If the opposite side indicates a maximum receive size that is smaller than the BLK value, the output size is adjusted downward.

**NOTE:** This is a reserved capability. PVM will not send an input size which is less than the maximum size possible.

**USER** *userid*

identifies the local user that is requesting a connection to a remote PVM node over an APPC link, or the user ID of the connecting PVMG virtual machine. This *userid* (and password) must match the name defined in the :userid. field of the CMS communications directory. This *userid* is a 1-to-8-character string identifying the:

- Local user that is requesting a connection to a remote PVM node over an APPC link,
- PVMG virtual machine to which an IUCV connection is to be made.

This operand applies to an APPC type link and to an IUCV connection to a PVMG virtual machine, and it must be specified.

### IP *address*

indicates the internet address of the remote PVM node. Because an IP address is a 4-level address, each level being a value from 1 to 255, *address* is specified as two fields. The first field contains the first two levels separated by a period, and the second field contains the second two levels separated by a period. An example of *address* would be 9.130 25.200. The IP keyword is only valid, and required, for TCPIP type links.

### PORT *portid*

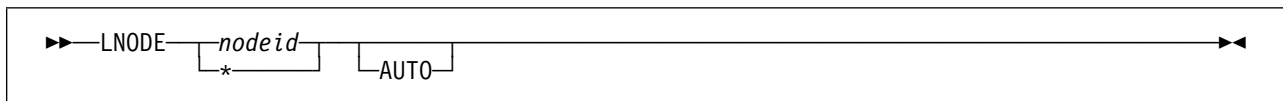
indicates the port number of the PVM virtual machine at the remote node in the range 10-65534. If not specified, the port number will default to the one used in the TCP configuration file record. The PORT keyword is only valid for TCPIP type links.

### TLSLABEL *label*

specifies the 1- to 8-character label of an x.509 digital certificate that will be used to encrypt/decrypt all data flowing over the link using TLS protocols. The specified certificate and its corresponding TLSLABEL must exist in the TLS/SSL Server certificate database. For additional information on the TLS/SSL Server and managing its certificate database, see *z/VM: TCP/IP User's Guide* and *z/VM: TCP/IP Planning and Customization*. The TLSLABEL keyword is only valid for TCPIP type links.

---

## LNODE



## Purpose

Use the LNODE record to define the node name that is selected by the ENTER key from the Initial Menu.

The AUTO option on this record allows the Initial Menu and the PVM Selection Screen to be bypassed. To go to a different system than the one specified by the AUTO option on the LNODE record, log on to a CMS virtual machine and specify the target system on the CMS PASSTHRU command.

**Note:** This record does not apply to MPVM.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If this record is omitted, the default value is the node name from the LOCAL record.

## Operands

### *nodeid*

is the name of the default selection node.

\* indicates that the local node is the default selection node.

### AUTO

specifies that the Initial Menu is bypassed and a session is initiated with the default selection node.

## LOCAL

```

▶▶—LOCAL—nodeid—zone—▶▶

```

### Purpose

Use the LOCAL record to define the name given to the local PVM system. The node identifier is also treated as an APPC/VM resource ID if PVM is to be recognized for APPC/VM communication and a RESID record does not appear in the PVM configuration file.

Usually, this record must be the first record in the configuration file. However, it may be preceded by the comment, LANG, and XDBC records.

### Operands

#### *nodeid*

is the name of the local node. The *nodeid* can be 1 to 8 characters. *DEFAULT* is a reserved word and should not be used as *nodeid*.

#### *zone*

is the time zone from GMT. The *zone* operand is entered as an unsigned number (*nn*) if the zone is west of GMT, and as *-nn* if the zone is east of GMT.

**Note:** If DIAGNOSE code X'0' returns the time zone offset, then that value overrides *zone*.

## MAXCONN

```

▶▶—MAXCONN—nnnn—▶▶

```

### Purpose

Use the MAXCONN record to indicate the maximum number of IUCV and APPC/VM paths supported by PVM. Use this value to generate a table for storing path information. The value ranges from 100 to 4096. If this record is missing, the default value is 1096.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record.

### Operands

#### *nnnn*

indicates the maximum number of IUCV paths supported by PVM. This number should not exceed the MAXCONN value in the CP OPTION Directory record for PVM. The value of *nnnn* ranges from 100 to 4096.

---

## MAXIDLE

▶▶—MAXIDLE—nnnn—◀◀

### Purpose

Use the MAXIDLE record to specify the maximum time a MPVM user may remain idle before PVM will terminate all of the user's sessions. The user is considered idle when there is no communication from the MPVM user to PVM for any session.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If omitted, the default value is 600 seconds (10 minutes). If more than one MAXIDLE record is entered in a configuration file, only the last MAXIDLE record in the list is accepted, the others are ignored. PVM will send out a warning message when approximately three-fourths of the time interval has elapsed.

### Operands

**nnnn**

Is the number of seconds allowed for an idle user. The value of nnnn ranges from 60 to 9998. A setting of 9999 will mean no maximum idle time limit.

---

## MAXSESS

▶▶—MAXSESS—nn—◀◀

### Purpose

Use the MAXSESS record to specify the maximum number of sessions a MPVM user may establish.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If omitted, the default value is 5 sessions. If more than one MAXSESS record is entered in a configuration file, only the last MAXSESS record in the list is accepted, the others are ignored.

### Operands

**nn** Is the number of sessions allowed for each MPVM user. The value of nn ranges from 1 to 99.

---

## MAXSUSP

▶▶—MAXSUSP—nnnn—◀◀

## Purpose

Use the MAXSUSP record to specify the maximum time a MPVM user may have a session suspended before that user's session will be terminated by PVM. A session is a suspended if:

- It is not the current session
- The MPVM SUSPEND command was issued for the current session
- A CP or CMS command was issued to the host session during a current session.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If omitted, the default value is 1200 seconds (20 minutes). If more than one MAXSUSP record is entered in a configuration file, only the last MAXSUSP record in the list is accepted, the others are ignored. PVM will send out a warning message when approximately one-half of the time interval has elapsed, and then again when approximately three-fourths of the time interval has elapsed.

## Operands

**nnnn**

Is the number of seconds allowed for a suspended MPVM session. The value of nnnn ranges from 60 to 9998. A setting of 9999 will mean that there is no maximum suspend time limit.

---

## MAXTDISC

►►—MAXTDISC—nnnn—◄◄

## Purpose

Use the MAXTDISC record to specify the maximum time a MPVM user may remain temporarily disconnected before all of the user's session will be terminated by PVM.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If omitted, the default value is 1200 seconds (20 minutes). If more than one MAXTDISC record is entered in a configuration file, only the last MAXTDISC record in the list is accepted, the others are ignored. PVM will send out a warning message when approximately three-fourths of the time interval has elapsed.

## Operands

**nnnn**

Is the number of seconds allowed for a temporarily disconnected MPVM user. The value of nnnn ranges from 60 to 9998. A setting of 9999 will mean no maximum temporary disconnect time limit. A setting of 0 will indicate that temporary disconnect for MPVM users is not allowed.

---

### MAXUSERS

▶▶—MAXUSERS—*nnn*—◀◀

#### Purpose

Use the MAXUSERS record to specify the maximum number of MPVM users who may connect to PVM.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If omitted, the default value is 100 MPVM users. If more than one MAXUSERS record is entered in a configuration file, only the last MAXUSERS record in the list is accepted, the others are ignored. The maximum number of users can also be limited indirectly through the number of IUCV and APPC/VM paths supported by PVM. This value can be changed through the MAXCONN record in the configuration file but cannot exceed the limit specified in the CP OPTION directory record for the PVM virtual machine.

#### Operands

**nnn**

Is the maximum number of MPVM users allowed. The value of nnn ranges from 1 to 999.

---

### MSGNOH

▶▶—MSGNOH—◀◀

#### Purpose

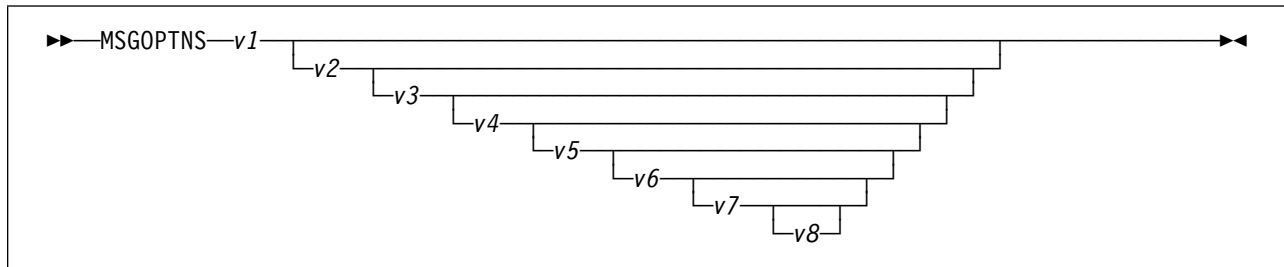
Use the MSGNOH record to indicate that responses to all commands entered by the CP SMSG command must be returned using the CP MSGNOH command rather than the CP MSG command, unless prohibited by the MSGOPTNS on that record when sending messages to the CP OPERATOR. The PVM user ID must have Class B privilege to issue the CP MSGNOH command.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If there is more than one MSGNOH record in the configuration file, only the last MSGNOH record in the list is accepted. PVM ignores the other MSGNOH records.

---

### MSGOPTNS





## Purpose

Use the MSGOPTNS record to indicate what options PVM will use regarding time stamps and message numbers for messages directed to the PVM user ID, to the audit file, to users, and to the operator. The installation can choose for each destination whether time stamps and message numbers should be appended to the messages. All parameters must be either a Y for yes or a N for no.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record.

## Operands

- v1* specifies the time stamps on the messages to the PVM user ID. The default is N.
- v2* specifies the message numbers on the messages to the PVM user ID. The default is N.
- v3* specifies the time stamps on the messages to the PVM audit file. The default is Y.
- v4* specifies the message numbers on the messages to the PVM audit file. The default is N.
- v5* specifies the time stamps on the messages to users. The default is N.
- v6* specifies the message numbers on the messages to users. The default is N.
- v7* specifies all messages written to the operator. These messages are written by a CP MSG command, overriding the MSGNOH record. The default is N, meaning that the MSGNOH option will take effect.
- v8* specifies the message numbers on the messages to the operator. The default is N.

## Usage Notes

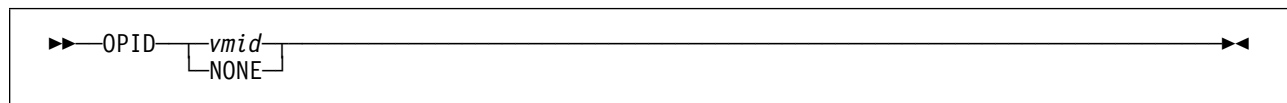
1. The values for the PVM audit file affect the record length of that file. The length of the AUDIT file depends on whether the audit records contain message numbers and/or time stamps. The length of the AUDIT file is:
  - 129 bytes if the audit record contains neither time stamps nor message numbers
  - 140 bytes if the audit record does not contain time stamps but does contain message numbers
  - 129 bytes + *maximum timestamp length* + 1, if the audit record contains a time stamp but does not contain message numbers

## PVMPROP Record

- 140 bytes + *maximum timestamp length* + 1, if the audit record contains a time stamp and message numbers.
2. The + 1 allows for a single blank following the timestamp.
  3. *Maximum timestamp length* refers to the longer of the AM or the PM timestamp for the default language. However, if the default language is a DBCS language, the *maximum timestamp length* is the longest of the:
    - AM and PM timestamp for the default language
    - AM and PM timestamp for the backup language.

---

## OPID



## Purpose

Use the OPID record to specify the virtual machine ID that receives messages that usually go to the system operator. This record can also be used to stop messages from being sent to the system operator.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If this record is omitted, the default is OP.

## Operands

*vmid*

is the virtual machine to receive operator messages.

**NONE**

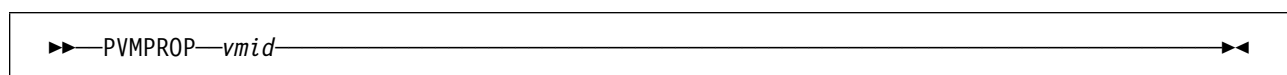
suppresses the messages sent to the operator.

## Usage Notes

1. If this record is omitted, operator messages are sent to the OPERATOR user ID.
2. The format of the messages sent to the specified operator depends on the options specified on the MSGOPTNS record and on whether PVM is authorized to use the CP MSGNOH command.
3. Only one *vmid* can be specified on this record.

---

## PVMPROP



## Purpose

Use the PVMPROP record to identify a virtual machine running a programmed operator that receives a copy of PVM error messages. All messages are transmitted to the virtual machine by the CP Special Message (CP SMSG) Facility. The messages are sent to the PVMPROP virtual machine in the same encoded format as sent to the Command Response Interface. See Chapter 13, “Facilities for Automating PVM and PVMG Operations” on page 145 for more information about PVMPROP.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record.

## Operands

*vmid*

is the virtual machine to receive error messages. All PVM messages are sent to the *vmid* in encoded form using the CP SMSG command.

### RESID

```
►►—RESID—resourceid—◄◄
```

#### Purpose

Use the RESID record to define a unique resource identifier. If the RESID record is not specified, the resource identifier defaults to the local node identifier specified in the LOCAL record.

The LOCAL record must precede the RESID record.

#### Operands

*resourceid*

is the APPC/VM resource identifier for this node. The *resourceid* must be unique within the system(s) accessing PVM using this name.

### ROUTE

```
►►—ROUTE—targetid—nextid  

DEFAULT—nodeid—◄◄
```

#### Purpose

Use the ROUTE record to define the next nodal path to a specified target node from the local PVM node. For PVM to PVM communications, routing must be consistently defined on both ends of network paths.

Use the **DEFAULT** *nodeid* form of the ROUTE record to define default routing for nodes not defined by LINK or ROUTE records. Only one default route record can be entered.

All LOCAL, LINK, and GROUP records must precede this record.

#### Operands

*targetid*

is the identification of the target system node.

*nextid*

is the identification of the next PVM node defined in a configuration file LINK or GROUP record, in the routing pattern from the local node to the target system node.

**DEFAULT** *nodeid*

indicates that the *nodeid* is the default node identifier. The *nodeid* is the PVM node through which all unknown nodes are to be routed. This node must be defined in a LINK or GROUP configuration file record. To avoid difficulties, do not define any PVM node identifiers with the name *DEFAULT*.

**Note:** Links that are members of groups should not be specified on ROUTE records.

## SELECT



### Purpose

Use the SELECT record to assign a frequently used node name to a PF key. Up to 11 SELECT records can be specified to assign 11 PF keys. The first SELECT record *nodeid* is assigned to PF1, the second SELECT record *nodeid* is assigned to PF2, and so forth to PF7. The eighth SELECT record is assigned to PF9 (PF8 is reserved for the SCROLL FORWARD function). The ninth SELECT record is assigned to PF10, and so forth to PF12.

A SELECT record without specifying the node ID omits the PF key associated with this SELECT record. This allows a PF key to be skipped.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record.

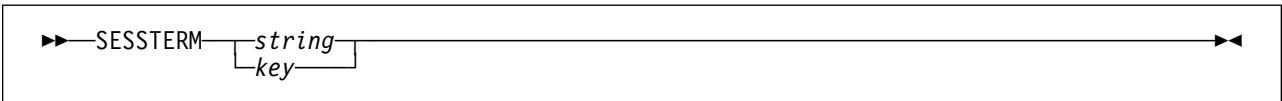
### Operands

*nodeid*  
is the identification of the node assigned to the PF key represented by this SELECT record. The *nodeid* may be 1 to 8 characters long. The name must not be a link name that is part of a GROUP definition.

### Usage Notes

1. Two SELECT records in the configuration file cannot contain the same node name.

## SESSTERM



### Purpose

Use the SESSTERM record to define the default session termination string, PF*n*, or PA key used at the local node.

All LOCAL, LINK, GROUP, ROUTE, OR AUTHORIZ records must precede this record. If this record is not entered in the configuration file, PVM uses the default string '####'.

### Operands

*string*

is a 1-to-8-character session termination string. Mixed case strings are allowed.

*key*

is the key to assign as the session termination key. Use the form *PFnn* or *PAn* to specify the key. For PF keys, *nn* can be a number from 01 (or 1) to 24. For PA keys, *n* can be 1 or 2.

---

## STATTIME

►►—STATTIME—*nnnn*—◄◄

### Purpose

Use the STATTIME record to specify the time interval between calls to the Link Statistics Exit (DVMNMEL) for connected links. This is also the interval that timing records are sent down the network links for determining link delay time.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If this record is omitted, the default value is 600 seconds (10 minutes). If there is more than one STATTIME record in the configuration file, only the last STATTIME record in the list is accepted. PVM ignores the other STATTIME records.

### Operands

*nnnn*

is the number of seconds between calls to the Link Statistics Exit. The value of *nnnn* ranges from 10 to 1800.

---

## SUBSET

►►—SUBSET—*string*—◄◄

### Purpose

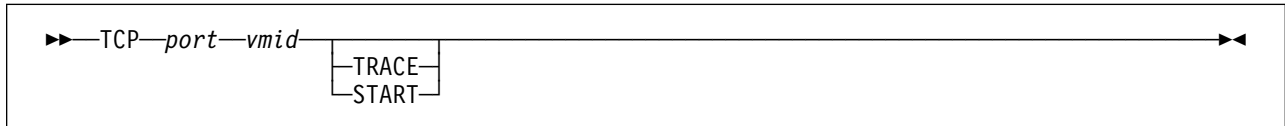
Use the SUBSET record to identify the command names that will be executed as CMS Subset Mode commands. Commands entered using this option do not produce the return code response and other responses, and they are not invoked with the *CMS* prefix.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record, and all SUBSET records must appear together.

## Operands

*string*  
is a 1-to-8-character command name.

## TCP



## Purpose

Use the TCP record to define information about an installation's site TCP/IP configuration to the PVM machine.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If there is more than one TCP record in the configuration file, only the last TCP record in the list is accepted. PVM ignores the other TCP records.

## Operands

*port*  
defines the TCP/IP port number for the local PVM machine in the range 10-65534. The default port number value is 999.

*vmid*  
defines the virtual machine ID running VM TCP/IP. The default user ID is TCPIP.

### TRACE

start an I/O trace of the TCP/IP redirector task's connection with VM TCP/IP.

### START

automatically start the TCP/IP redirector task during PVM initialization.

## TDISC



## Purpose

Use the TDISC record to specify the maximum length of time a PASSTHRU user may remain in a temporary disconnected state from a PVM session before PVM terminates the session.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If this record is omitted, the default value is 1200 seconds (20 minutes). If there is more than one TDISC record in the configuration file, only the last TDISC record in the list is accepted. PVM ignores the other TDISC records.

## Operands

*nnnn*

is the number of seconds to be allowed for temporary disconnect. The value of *nnnn* ranges from 30 to 9999.

---

## TIMEOUT

►►—TIMEOUT—*nnnn*—◄◄

## Purpose

Use the TIMEOUT record to specify the time interval between line status checks by network line drivers when no data is being transferred. For the network line drivers, if there has been no activity on the line during the specified time interval, a transaction is sent down the line to ensure that the line is still up.

This record does not apply to IUCV line drivers.

All LOCAL, LINK, GROUP, ROUTE, or AUTHORIZ records must precede this record. If this record is omitted, the default value is 300 seconds (five minutes). If there is more than one TIMEOUT record in the configuration file, only the last TIMEOUT record in the list is accepted. PVM ignores the other TIMEOUT records.

## Operands

*nnnn*

is the number of seconds between status checks. The value of *nnnn* ranges from 30 to 9999.

---

## XDBC

►►—XDBC—

OP
CONS
ORIG

—◄◄

## Purpose

Use the XDBC record to exclude certain users on the local system from receiving, in the national language, any PVM messages that, in the national language, contain double-byte characters. The identified users receive those messages in the backup language instead. The record may appear anywhere in the file.

The principal reason for the XDBC record is that PVM cannot tell whether the virtual machine consoles of other virtual machines on a VM system are capable of handling double-byte characters.



## Operands

**OP**

identifies the CP Operator.

**CONS**

identifies the PVM user ID console (display station) and the AUDIT file.

**ORIG**

identifies all virtual machine users on the local VM system who are not PVM users, other than the CP operator and the PVM operator. That is, the originators of actions or commands to which PVM messages are sent in response.

The position of the XDBC record in the configuration file does not matter, and it takes effect immediately. If the LANG record identifies a DBCS language and the PVM user ID or the CP operator should not receive DBCS messages, the XDBC record should be specified immediately before or after the LANG record.

If PVM finds an error in the XDBC record, PVM immediately excludes OP, CONS, and ORIG from receiving DBCS messages. The CDBC command must be entered to enable messages to these destinations.



## Appendix D. PVMG Commands

The PVMG administrator can use a variety of commands on the PVMG virtual machine to display system status, change system parameters, send commands to other PVMG nodes, or send messages to interactive session users.

These commands, when issued from PVMG's console, must be issued in the format:

```
pvmg command operands...
```

### PVMG

is to signify a PVMG command follows.

**Note:** Throughout this chapter, it is assumed that the PVMG program is running in a virtual machine whose user ID is *PVMG*.

The *pvmg* prefix is assumed for all commands in this chapter.

### Notes on Using Commands

Certain PVMG commands can be issued by users identified in *AUTHORIZ* records in the PVMG configuration file. These commands are the *A* commands listed in Table 20 on page 261. All PVMG commands are listed in this table. Some commands must be used by a PVMG operator logged onto the PVMG user ID. These commands are the *O* commands. A virtual machine user may use any of the unrestricted commands. These commands are the *G* commands. Some commands are intended to be used by the PVMG system programmer. These commands are the *S* commands.

Some PVMG command options have restriction levels that are different from the remaining options for that command.

Unless entered at the PVMG user ID, each PVMG command expression must be included in a CP *SMMSG* (Special Message) command that contains the PVMG user ID. This sends the command to PVMG to be executed. The exception to this rule is when a local installation has provided an *exec* for each command that automatically puts the *SMMSG pvmid* characters in front of the PVMG command expression.

A command prefix can be used when issuing PVMG commands. Command prefixes tell PVMG to send command responses using a particular format. See "Using Command Prefixes" on page 149 for more information about command prefixes. For information on how to use command prefixes in application programming, see Chapter 13, "Facilities for Automating PVM and PVMG Operations" on page 145.

**Note:** In applying prefixes the user should understand that the command *PVMG* is not a PVMG component command. It is a GCS command required to direct the command entered on the same line to the PVMG component. Therefore, prefixes should be applied to the particular command being entered, for example, *Query*.

## How to Read the Syntax Diagrams

Throughout this book, syntax is described using the structure defined below.

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

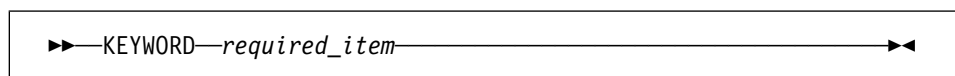
The **▶▶**— symbol indicates the beginning of a command.

The —**▶** symbol indicates that the command syntax is continued on the next line.

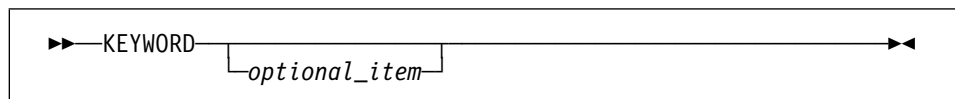
The **▶**— symbol indicates that a command is continued from the previous line.

The —**◀◀** symbol indicates the end of a command.

- Required items appear on the horizontal line (the main path).

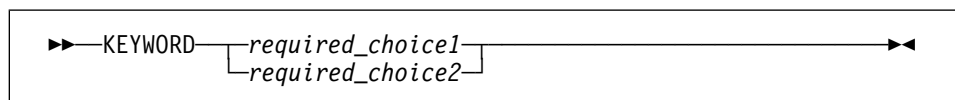


- Optional items appear below the main path.

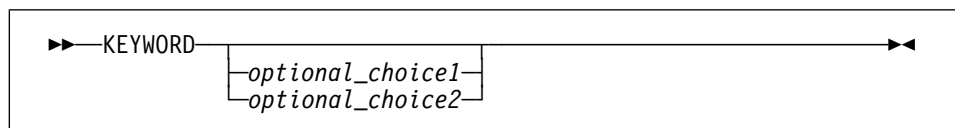


- If a choice can be made from two or more items, they appear vertically, in a stack.

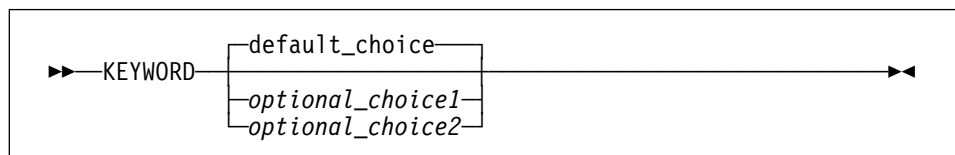
If one item *must* be chosen, one item of the stack appears on the main path.



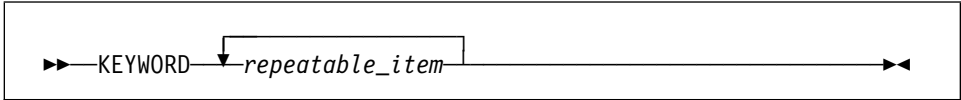
If choosing one of the items is optional, the entire stack appears below the main path.



If one of the choices is a default, it appears above the main path.



- An arrow returning to the left above the main line indicates an item that can be repeated.



An arrow above a stack indicates that the items in the stack can be repeated.

The following example of the PASSTHRU command illustrates many aspects of a syntax diagram. Explanation of the PASSTHRU command is provided within the syntax diagram.

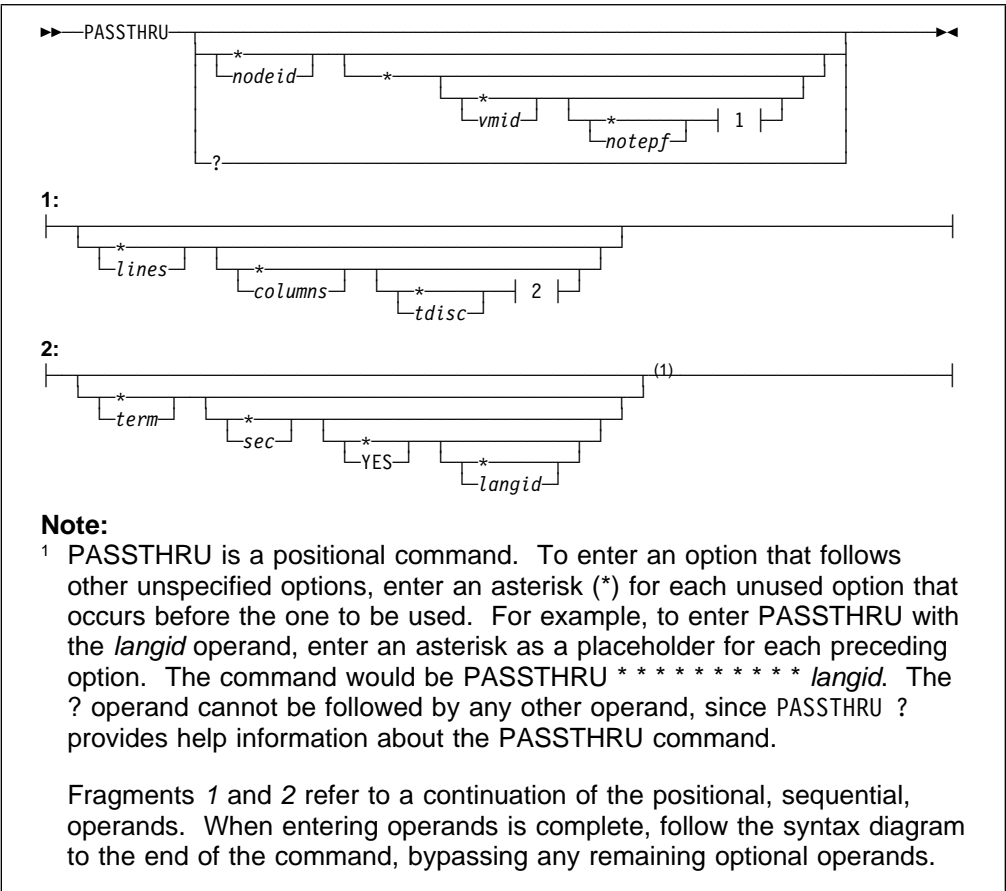


Table 20 lists the commands, their functions, and the users authorized to use them. Note the following about the first column of this table:

Table 19. Key to authorized users

User	Meaning
A	Alternate operator specified on an AUTHORIZ record in the PVMG configuration file.
O	PVMG operator logged onto the PVMG user ID.
G	General VM user.
S	PVMG system programmer.

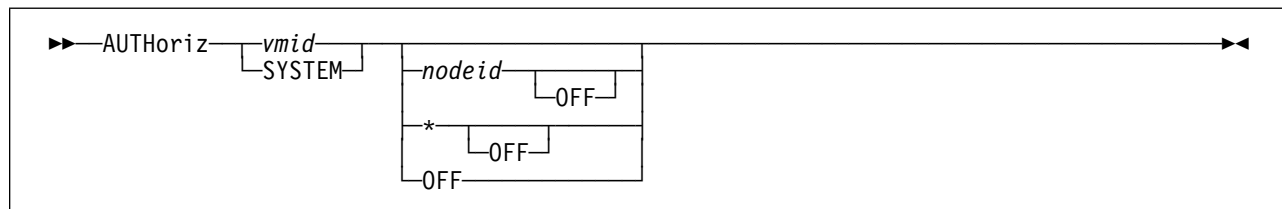
Table 20 (Page 1 of 2). PVMG Commands and Execs and Their Functions		
User	Command	Function
A O	AUTHORIZ	Adds or deletes entries from the PVMG authorized user list. This command must be issued on the local system.

## PVMG AUTHORIZ Command

Table 20 (Page 2 of 2). PVMG Commands and Execs and Their Functions

User	Command	Function
A O	CDBC	Specifies whether display unit can or cannot handle messages in the default language without regard to double-byte character restrictions.
A O G	CMD	Sends PVMG commands to another PVM or PVMG node for execution.
O	DISC	Disconnects the PVMG user ID.
A O	DROP	Terminates an IUCV link to a PVM virtual machine, or a user session.
O	INIT	Starts PVMG initialization.
A O	LMSG	Sends a message to PVMG users on a specified link.
A O	LOGMSG	Defines the LOGMSG area on the PVMG Selection Menu.
A O G	MSG	Causes PVMG to forward a message to a specific VM user.
A O G	PMSG	Causes PVMG to forward a message to specified PVMG users. The ALL option is restricted to the A and O user groups.
S	PVMGBLD	Regenerates the PVMG LOADLIB.
A O G	QUERY	Displays information about PVMG. The AUTHORIZ option is restricted to the A and O user groups and to the local VM system.
A O	QUIESCE	Quiesces the PVMG server, or a line driver task.
A O	RESUME	Reverses the effect of the QUIESCE command.
A O	ROUTE	Adds, deletes, or changes a node's routing.
O	RUNPVMG	Loads and starts PVMG.
A O	SELECT	Adds, deletes, or changes nodes assigned to PF keys on the PVMG Selection Menu.
A O G	SHOW	Shows the path of a particular session.
O	SHUTDOWN	Terminates the PVMG server.
A O	SPOOL	Logs console messages and information to a virtual printer.
A O	START	Starts an IUCV link.
A O	TRACE	Starts or stops line tracing on a link, or system tracing on the PVMG server.

## AUTHORIZ



## Purpose

Use the AUTHORIZ command to add or delete entries from the PVMG authorized user list. To make this change permanent, the installation must add or delete an AUTHORIZ record in the PVMG configuration file.

## Operands

### *vmid*

is the user identification of the CP virtual machine to add to or delete from the authorization list. Without specifying \* or *nodeid*, this authorization is for the local node only.

### **SYSTEM**

authorizes the PVMG operator to issue commands to nodes (\* or *nodeid*) other than the local node. Use this operand rather than the PVMG *vmid*. Without specifying \* or *nodeid*, this authorization is for the local node only.

### *nodeid*

is the specific node where the *vmid* is authorized. the same *vmid* can be specified from different *nodeids*.

- \* indicates that *vmid* is authorized on any node. Consider this option carefully, since it may be a security risk in a network where user IDs could be duplicated. The *vmid* cannot be added to the authorization list using both a global node ID (\*) and a specific node ID.

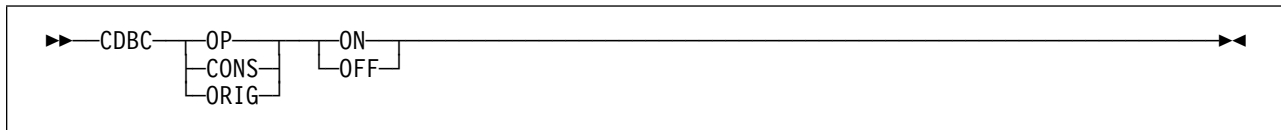
### **OFF**

removes the specified user from the authorization list. Specifying OFF as the second operand will result in using the local *nodeid* as the default specified node. This parameter is not validated against the list of link or route names provided to PVMG by the LINK and ROUTE records.

## Usage Notes

1. This command is restricted to the PVMG operator or an alternate operator specified on an AUTHORIZ record in the PVMG configuration file and must be issued on the local system.
2. If the operator makes a typographic error in issuing the AUTHORIZ command, PVMG may create an entry in its AUTHORIZ table using this incorrect information. If the operator recognizes this error, delete the entry using the **OFF** option and submit the correct AUTHORIZ command again.
3. Use the QUERY AUTHORIZ command to display the contents of the AUTHORIZ table.
4. Without specifying \* or *nodeid*, this authorization is for the local node only.
5. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

## CDBC



## Purpose

Use the CDBC (Change Double-Byte-Characters) command to change the DBCS-exclusion settings for OP, CONS, and ORIG without having to add an XDBC record to the PVMG Configuration File and re-initialize PVMG. These settings are ignored when a user (other than the PVMG console operator), specifically requests a language by using a command prefix.

## Operands

### OP

identifies the CP operator.

### CONS

identifies the PVMG console.

### ORIG

identifies all virtual machine users on the local VM system, other than the CP operator and the PVMG operator. That is, ORIG identifies the originators of actions or commands to which PVMG messages are sent in response.

### ON

indicates that the default language messages should be issued to the user without regard to DBCS content.

### OFF

indicates that the backup language should be used for any messages to the user whenever the default language message that would otherwise be used contains DBCS.

## Usage Notes

1. This command is restricted to the PVMG operator or an alternate operator specified on an AUTHORIZ record in the PVMG configuration file.
2. Issue the CDBC CONS OFF command when logged on to the PVMG user ID to prevent errors. Use CDBC CONS ON when disconnected.
3. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.



## CMD

```
►►—CMD—nodeid—command—◄◄
```

## Purpose

The CMD command provides the ability to send a command to be executed at another PVM or PVMG node in the PVM network. The issuer must be authorized at the remote node to have restricted commands issued at that node and receives the output generated from this command. The CMD command is useful for various query commands.

## Operands

*nodeid*

is the node on the PVM network where the command is to execute. This may be a PVM node or another PVMG node.

*command*

is any PVMG or PVM command authorized to the issuer.

## Usage Notes

1. The CMD command sends commands from a PVMG server to a remote PVM or PVMG node, and then returns the results. For example, suppose the command is entered:

```
pvmg cmd other q sys
```

This sends a *QUERY SYSTEM* command from the PVMG virtual machine through the PVM network to the node called *OTHER*. The PVM or PVMG server at the node called *OTHER* issues the command and then sends the results back.

2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. Place the prefix in front of the CMD command, rather than in front of the *command* portion of the command. For example, use the command:

```
smsg pvmg (l.kanji) cmd rmtpvmg5 query system
```

instead of:

```
smsg pvmg cmd rmtpvmg5 (l.kanji) query system
```

See “Using Command Prefixes” on page 149 for more information about command prefixes.

---

## DISC

```
▶▶—DISC—◀◀
```

### Purpose

Use the DISC command to disconnect the PVMG operator display station. Use this command rather than the CP DISC command. Using the DISC command allows system spool files to be properly maintained.

### Usage Notes

1. This command is restricted to the PVMG operator logged on to the PVMG user ID.
2. If the PVMG SPOOL CONSOLE START command has been issued, all "writes" or output messages, including those from PVMG, sent to the virtual console will continue to be captured.
3. If PVMG is logged on, disconnecting while SET RUN is OFF, then reconnecting, will not display any log messages or a RECONNECTED message but only the CP read status. The messages will appear after a CP BEGIN command is issued.

However, disconnecting while SET RUN is ON, then reconnecting, will display log messages and a RECONNECTED message.

---

## DROP

```
▶▶—DRop—┐
          │┐
          │├──Node───nodeid──┐
          │├──User──nodeid──userid──┐
          └──────────────────────────▶▶
```

### Purpose

Use the DROP command to terminate a:

- Specific IUCV link to a PVM virtual machine
- User session.

### Operands

#### **Node** *nodeid*

immediately terminates the specified PVMG link. Any active user sessions associated with this link are also terminated. The *nodeid* is the node name defined in the PVMG configuration file.

#### **User** *nodeid userid*

terminates the specified user session. The user is notified of impending system action. The *nodeid* is the identification of the user's origin or destination node where the user's session is terminated. The *userid* is the identification of the user whose PVMG session is terminated.

The session identification is:

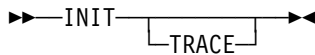
- GRAFxxx for PVM dialed users
- the user's user ID for PASSTHRU users
- \*SESnnnn for an MPVM session
- a logical device address if the session destination is a PVM node,
- a PVMG session ID (GRAFxxxx), if the session destination is an SNA application, or
- an LU name, if the session origin is an SNA application.

## Usage Notes

1. This command is restricted to the PVMG operator or an alternate operator specified on an AUTHORIZ record in the PVMG configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

---

## INIT



## Purpose

Use the INIT command to start PVMG initialization.

During initialization, PVMG processes its configuration file records, attaches its execution tasks, and opens its VTAM access control block (ACB). The INIT command cannot be executed again until PVMG has been shutdown.

## Operands

### TRACE

TRACE indicates that system wrap trace is to be invoked during PVMG initialization. PVMG will generate a 16K wrap table for storing trace records. Tracing is stopped using the TRACE command.

## Usage Notes

1. The INIT command is restricted to the PVMG operator.
2. The INIT command may be processed by the PROFILE GCS exec, which executes after the GCS recovery machine is IPLed, or it can be issued from the PVMG console.

---

## LMSG

►—LMSG—*nodeid*—*text*—◄

### Purpose

Use the LMSG command to send a message to all PVMG users associated with a specific link, or to all interactive users who have sessions that access the PVMG server through VTAM, or that access VTAM through PVMG. If the name is the local PVMG node name, the message is sent to all users, except those with PVM-to-PVM sessions using the PVMG server as an intermediate node.

When the message arrives at the specified user's system, it replaces the current image on the user's display. The user recalls the stored image by pressing any interrupt key. The sender's node ID and VM ID are displayed with the message at the receiving location(s).

### Operands

*nodeid*

specifies a specific VM node ID for users who have interactive sessions using PVMG, or the local PVMG node name.

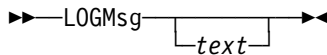
*text*

is the text of the message sent. The message can be of any format and characters acceptable to the CP command processor. The total length is limited to the CP command input limit.

### Usage Notes

1. This command is restricted to the PVMG operator or an alternate operator specified on an AUTHORIZ record in the PVMG configuration file.
2. No error message is issued if the specified link has no sessions.
3. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

# LOGMSG



## Purpose

The operator uses the LOGMSG command to enter the variable data text that appears in the LOGMSG area on the PVMG Selection Menu.

Any entry of the LOGMSG command overlays the previous LOGMSG text. Issuing a LOGMSG command with no text overlays the previous LOGMSG text with blanks.

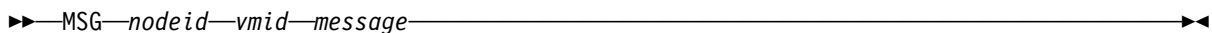
## Operands

*text*

is the text of the message. which may contain any characters. This field may be from 0 to 125 characters long, using two lines. With LOGMS, the message could only be up to 124 characters long. With LOGMSG, the message could only be up to 123 characters long.

## Usage Notes

1. This command is restricted to the PVMG operator or an alternate operator specified on an AUTHORIZ record in the PVMG configuration file.
2. When issued from another virtual machine through SMSG, the text operand length may be up to 136 characters long.
3. When issued from another virtual machine through SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

**MSG**

## Purpose

MSG causes a line of text to be sent to the specified node and presented to a user or operator using the CP MESSAGE command.

## Operands

nodeid

is the identifier of the target node that is to receive the specified message text line. This node can be the local node.

*vmid*

is the identifier of a logged on user at that location.

### *message*

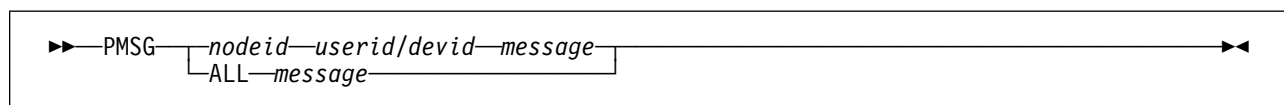
is the text of the message to be forwarded. The message can be of any format and characters acceptable to the CP command processor. The total message length is limited to the CP command input limit.

## Usage Notes

1. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

---

## PMSG



## Purpose

Use the PMSG command to send a message to a particular user or all users who have sessions with or through PVMG. When the message arrives at the specified user's system, it replaces the current image on the user's display. The user recalls the stored image by pressing any interrupt key. The sender's node ID and VM ID are displayed with the message at the receiving locations.

## Operands

### *nodeid*

is the identification of the target user's node when the message is forwarded to a single user.

### *userid/devid*

is the identification of the user (*userid*) or the identification of the user's attached display support task (*devid*) when the message is forwarded to a single user.

### *ALL*

forwards the message to all PVMG users whose sessions originate, terminate, or pass through this PVMG server machine.

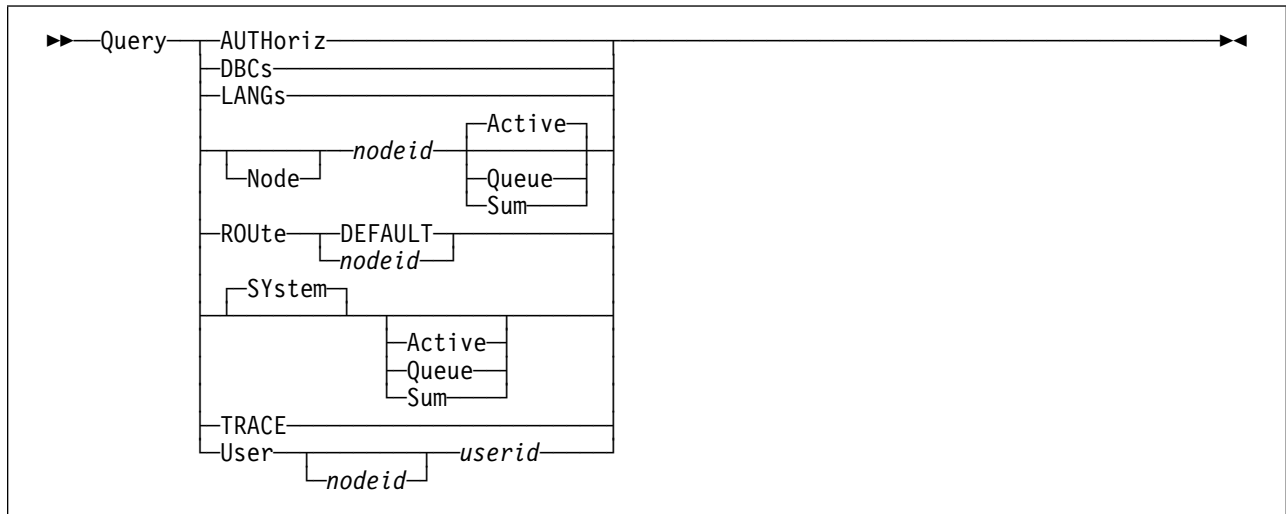
### *message*

is the text of the message forwarded. The message can be of any format and characters acceptable to the CP command processor. The total length is limited to the CP command input limit.

## Usage Notes

1. The ALL option of this command is restricted to the PVMG console operator or an alternate operator specified on an AUTHORIZ record in the PVMG configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

## QUERY



## Purpose

Use the QUERY command to display:

- Users authorized to issue restricted PVMG commands
- Contents of the routing table
- System status
- User session information
- Current DBCS settings
- National languages enabled on a PVMG system
- Current status of a given link
- Current trace settings.

QUERY SYSTEM is the default when the QUERY command is issued with no operands,

## Operands

### AUTHoriz

displays the current list of authorized users in the system who can enter PVMG restricted commands.

### DBC's

displays the DBCS settings for CP operator, the PVMG console, and the VM users when the default language for the PVMG virtual machine is a DBCS language.

### LANGs

displays the language identifiers of the languages for which PVMG is enabled. LANGs also displays the languages used as the default national language and as the backup language at the local node.

### Node *nodeid*

displays the status of the specified link.

**Active**

displays the status of the specified link, if active. Active is the default if Queue, or Sum are not specified.

**Queue**

displays the status of the specified link, and the associated user sessions for the specified link.

**Sum**

displays the statistics of the link, including the number of transaction counts (the number of data blocks)

**ROUte**

displays the contents of the PVMG server's routing table for either the DEFAULT route or a particular route name.

**DEFAULT**

displays the default route name.

**nodeid**

the node's entry from the routing table is displayed.

**SYstem**

displays information about the status of the system. with no other operands, the status of each link defined within the system is displayed. QUERY SYstem is the default value if the QUERY command is entered with no operands.

**Active**

displays the status of each active link. If there are no active links, only the name of the local system is displayed.

**Queue**

displays the status and the associated user sessions for each active link.

**Sum**

displays the status of each link, including transaction counts

**TRACE**

displays trace status

**User** *nodeid userid*

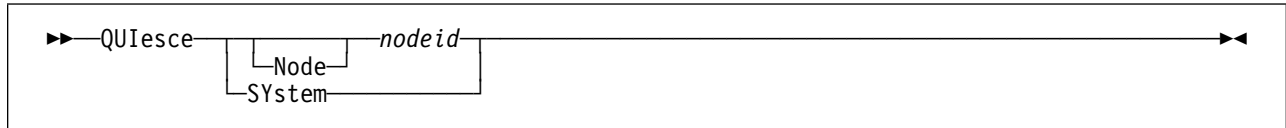
displays information about the associated session identified by *nodeid* and *userid*. The specified session must be known to the PVMG server. The response includes the user's real display station address, type, and model number, along with the session origin and destination IDs.

## Usage Notes

1. The AUTHorize operand is restricted and must be issued from a user ID on the local system.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.



## QUIESCE



### Purpose

The PVMG operator uses the QUIESCE command to quiesce:

- The PVMG server
- A line driver task.

*Quiesce* is a term used to describe the state of a task (or PVMG itself). A task in a quiesce state does not allow any new sessions and terminates when all active sessions have ended.

The effect of the QUIESCE command can be reversed by issuing the RESUME command. QUIESCE does not terminate any line drivers that have active sessions.

### Operands

#### **Node** *nodeid*

indicates that the node *nodeid* is quiesced. After issuing this command, no new users are allowed to initiate sessions through the associated *nodeid*.

When the last user session on the *nodeid* terminates, the node terminates. If no user session is active, the specified *nodeid* terminates immediately.

The *nodeid* must be the same name defined in the configuration file for the corresponding node.

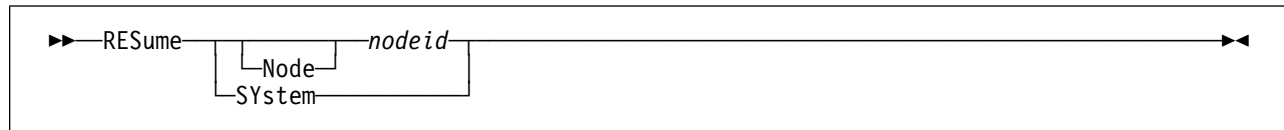
#### **SYstem**

indicates that the PVMG server is quiesced. After issuing this command, no new users are allowed to initiate sessions through the PVMG server.

### Usage Notes

1. This command is restricted to the PVMG operator or an alternate operator specified on an AUTHORIZ record in the PVMG configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

## RESUME



### Purpose

Use the RESUME command to reverse the effect of a previously issued QUIESCE command. As a result, the specified line, or the entire system is made available.

**Note:** If the quiesced line driver has already terminated due to the QUIESCE command, use the START command instead of the RESUME command.

### Operands

#### **Node** *nodeid*

indicates that the use of line defined by *nodeid* is resumed. Following the issuing of this command, new users are allowed to initiate sessions on the affected *nodeid*.

The *nodeid* must be the name defined in the configuration file for the corresponding line.

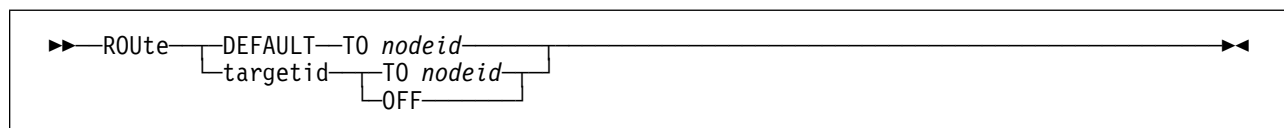
#### **SYstem**

indicates that the use of the PVMG server is resumed. After issuing this command, new users are allowed on the PVMG server.

### Usage Notes

1. This command is restricted to the PVMG operator or an alternate operator specified on an AUTHORIZ record in the PVMG configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

## ROUTE



### Purpose

The ROUTE command modifies the in-storage routing table used by PVMG for session routing. Node names may be added, deleted, or have their routing altered by the command.

Operands

DEFAULT

changes the DEFAULT routing.

*targetid*

is the target node added, deleted, or modified.

TO *nodeid*

indicates that the target node is routed through the link *nodeid*. The *nodeid* is the node name of the link. This node must be a PVM node directly connected to the local PVMG server.

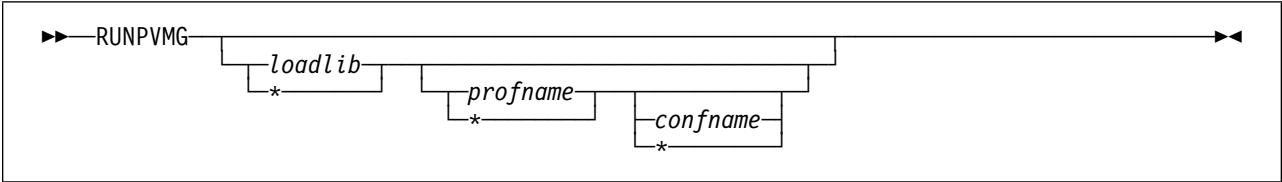
OFF

removes the specified target node from the routing table. OFF may not be specified as a default route.

Usage Notes

- 1. This command is restricted to the PVMG operator or an alternate operator specified on an AUTHORIZ record in the PVMG configuration file.
- 2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

RUNPVMG



Purpose

Use the RUNPVMG command to load and start PVMG.

Operands

*loadlib*

is the name of the PVMG loadlib loaded. Specifying \*, the default file name, *loadlib*, is PVMG. The file type is LOADLIB.

*profname*

is the name of the PVMG exec executed at startup time. Specifying \*, the default file name, *profname*, is PROFILE. The file type of this exec is GCS.

*confname*

is the name of the configuration file. Specifying \*, the default file name, *confname*, is PVMG. The file type of this configuration file is CONFIG.

## SELECT

```

▶▶—SELECT—PFnn—OFF—
                        └─nodeid─┘

```

### Purpose

Use the SELECT command to dynamically add, delete, or modify the nodes assigned to PF keys by the SELECT records of the configuration file.

### Operands

#### PFnn

is the PF key modified. The value of *nn* ranges from 1 to 12, excluding 8.

#### OFF

indicates that the PF key definition is to be reset

#### nodeid

is the name of the node selected by the specified PF key.

### Usage Notes

1. This command is restricted to the PVMG operator or an alternate operator specified on an AUTHORIZ record in the PVMG configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.

## SHOW

```

▶▶—SHOW—┬──ORIG──┬──userid──
          │└─DEST─┘
          └─PATH─nodeid─┘

```

### Purpose

Use the SHOW command to show the path of a particular session, or the path to a particular node. The system on which the command is first executed locates the session and displays information indicating that it is part of the path. This message gives the link upon which the session comes in, and the link upon which the session goes out.

### Operands

#### ORIG

traces the path to the origin. Using ORIG when the origin is not the local system or the incoming link, the command is forwarded to the node connected to the incoming link for processing.

**DEST**

traces the path to the destination. Using DEST when the destination is not the local system or the outgoing link, the command is forwarded to the node connected to the outgoing link for processing. In this way, the command is executed on each intermediate node along the path.

**PATH**

shows the PVM routing to get to a particular target node.

*nodeid*

is the node name of the origin or destination of the session. For ORIG or DEST, if a value is not specified for *nodeid*, the local node name is assumed. For PATH, a value must be specified for *nodeid*, and it is the target node for which path information is required.

*userid*

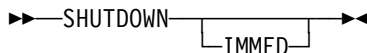
is the PVMG identification for the origin or destination of the session.

## Usage Notes

1. SHOW PATH shows the PVMG routing to get to a particular target node. The command locates the outbound path to the target node. If the target node is the local node or a nonnetwork node, messages indicate the path to the target node, followed by the message COMMAND COMPLETE. If the target node is not the local node or a nonnetwork node, the message Path to node xxxxx is through node yyyyy is displayed and the SHOW command is forwarded to the node yyyyy for execution.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

---

## SHUTDOWN



## Purpose

Use the SHUTDOWN command to conditionally terminate the PVMG server. If the PVMG server has no active sessions, then PVMG will shut down immediately. However, if the PVMG server determines that there are active sessions, it will not shut down until all active sessions end. All line drivers are placed in a quiesced state. Line drivers terminate when the last users of the associated lines terminate their sessions. PVMG terminates when the last user session terminates and all associated tasks terminate with it unless the IMMEDIATE option is specified. The operator can then wait for the sessions to terminate, or issue the SHUTDOWN IMMEDIATE command to shut down immediately.

## PVMG SHUTDOWN Command

### Operands

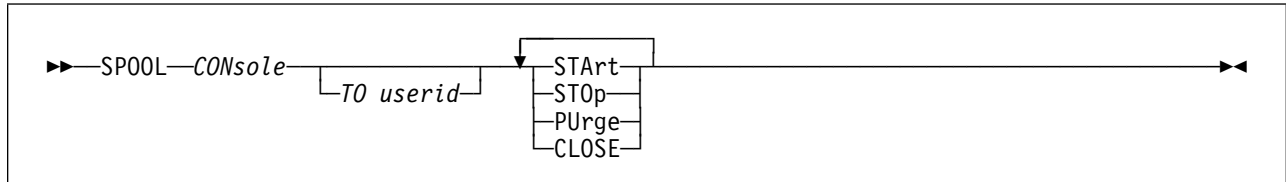
#### **IMMED**

terminates the PVMG server, canceling all active sessions.

### Usage Notes

1. This command is restricted to a PVMG operator logged onto the PVMG user ID.
2. For SHUTDOWN IMMED, sessions are terminated immediately.

# SPOOL



## Purpose

Use the PVMG SPOOL command to log console messages and information to a virtual printer to transfer later to a user ID on the same system. This command can be used for monitoring the PVMG machine.

## Operands

## CONsole

modifies the options for the virtual console spool file and/or initiates or stops the spooling of virtual console input and output, including CP input/output and PVMG console messages.

**TO userid**

sends the output of the virtual device to the virtual reader of the specified userid. The distribution code on the spool file retains the distribution code specified for the originator in the system directory.

**CLOSE**

closes the console spool file. If PURGE is specified, CLOSE is not necessary.

## PURge

closes and purges the console spool file. If PURGE is specified, CLOSE is not necessary. PURGE is equivalent to entering CLOSE with the PURGE option.

**STArt**

begins placing input and output from the PVMG console into a spool file. After the console is closed, the file becomes a printer spool file to be transferred to the userid, if specified.

**STOp**

terminates the spooling of the PVMG console. STOP does not close the console spool file.

## Usage Notes

1. The START and STOP operands can not be specified together.
2. When disconnecting from the PVMG console, use the PVMG DISC command instead of the CP DISC command so PVMG will continue to issue messages to the console.

---

## START

```
>>STAr<Node>nodeid<<
```

### Purpose

Use the START command to start an IUCV link. The START command does not have to be invoked during PVMG initialization. PVMG attempts to connect each line defined with a configuration file LINK record during initialization. Use the START command if, after initialization, a link needs to be placed in the connect state.

### Operands

**Node** *nodeid*

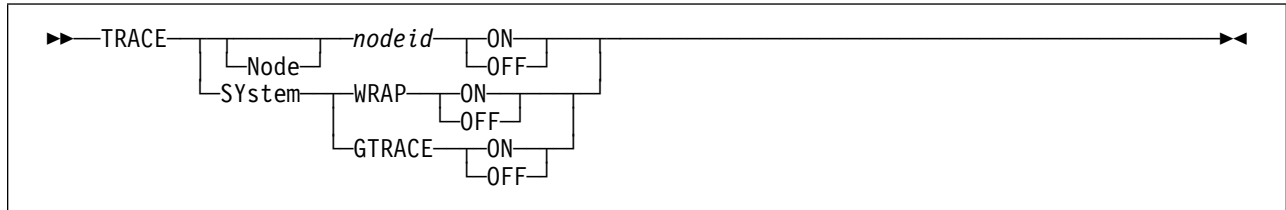
is the node ID to be started. The *nodeid* must be defined by a configuration file LINK record.

### Usage Notes

1. This command is restricted to the PVMG operator or an alternate authorized operator.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See “Using Command Prefixes” on page 149 for more information about command prefixes.
3. If the link on the PVM side is not active, PVMG will attempt the IUCV connect at 60 second intervals until either communications are established, or a PVMG DROP node command is issued by an operator.



## TRACE



## Purpose

Use the TRACE command to start or stop tracing on a link.

## Operands

### Node *nodeid*

specifies the node name (*nodeid*) associated with the PVMG link for which tracing is requested. The node name specified must be a valid node name as specified by a LINK record. Trace information is written to a wraparound buffer area for a particular link.

PVMG allocates an 8K byte buffer storage area for a wraparound trace table for each node for which tracing is started. Line driver activity is written as trace records to this table.

### System

causes a trace for the PVMG component. Trace information is written to a PVMG managed wraparound table. Specifying OFF terminates only the specified system tracing option. Other system trace options are not terminated.

### WRAP

causes wraparound tracing for system events. Trace records are written to a 16K byte wraparound trace table managed by PVMG.

### GTRACE

writes the output to a GCS trace table.

### ON

initiates the trace. If the GTRACE option is specified, the GCS ITRACE command is invoked.

### OFF

terminates the trace. If the GTRACE option is specified, the GCS ITRACE command is reset.

## Usage Notes

1. This command is restricted to the PVMG operator or an alternate operator specified on an AUTHORIZ record in the PVMG configuration file.
2. When issued from another virtual machine using SMSG, a command prefix can be used with this command to tell PVMG how to respond to the command. See "Using Command Prefixes" on page 149 for more information about command prefixes.

---

## CP Command Used with PVMG

---

### SMSG

---

▶▶ SMsg *vmid* *(prefix)* *command* ▶▶

#### Purpose

Use the CP SMSG command to issue PVMG commands to PVMG from another virtual machine.

#### Operands

*vmid*

is the virtual machine identification of the PVMG user ID.

*prefix*

is any valid PVMG command prefix. See “Using Command Prefixes” on page 149 for more information about command prefixes.

*command*

is any valid PVMG command.

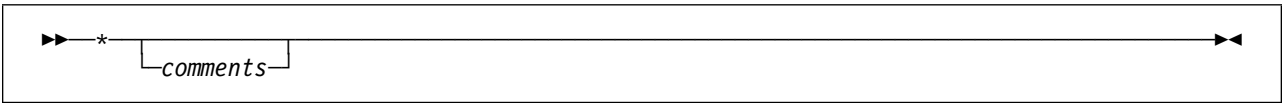
## Appendix E. PVMG Configuration File Records

The PVMG configuration file (PVMG CONFIG) provides installation independent information to the PVMG service machine. It is a fixed length 80 byte file. Except for comments and *string values*, input must be in upper case, and is restricted to 80 columns with no continuation. Operands in a configuration file record must be separated by one or more blanks. During PVMG initialization this file is processed and used to define execution attributes. Use the \* record to imbed comments in the configuration file.

PVMG configuration file records do not have to be sequenced in a particular order. PVMG process records using two passes. If errors are encountered, PVMG terminates following processing of the records.

This chapter includes a description of each record type.

\*



### Purpose

Use the \* record to imbed comments in the PVMG configuration file. This record is ignored during configuration file processing. Place this record anywhere in the PVMG configuration file.

### Operands

*comments*  
is any text message. This information must follow the asterisk(\*).

### Usage Notes

1. The Comment record must begin in column one of the PVMG configuration file.
2. There can be no blank lines in the PVMG configuration file. To insert lines between records, use this record.

---

## APPLN

▶▶—APPLN—*name*—*applid*————▶▶

### Purpose

Use the APPLN record to define alternate names for VTAM applications accessible through PVMG. These alternate names can be specified by users requesting sessions with these applications that are defined in ROUTE table entries at other PVM nodes.

### Operands

**name**

is the name associated with the VTAM application name provided on the second parameter.

**applid**

is the VTAM application name as it is known to VTAM

### Usage Notes

1. All names defined in the PVMG configuration file must be known to the originating PVM node for session requests for the target *applid*. Names are made known to the originating PVM node by defining them in the route table (using the ROUTE record in the PVM configuration file) for that PVM server virtual machine. For example, a local configuration:

- PVM node named PVM1
- PVMG node named PVMG1
- VTAM application named TSO1

The installation wants to refer to TSO1 as OURTSO.

The PVM configuration file will contain a ROUTE record:

```
ROUTE OURTSO PVMG1
```

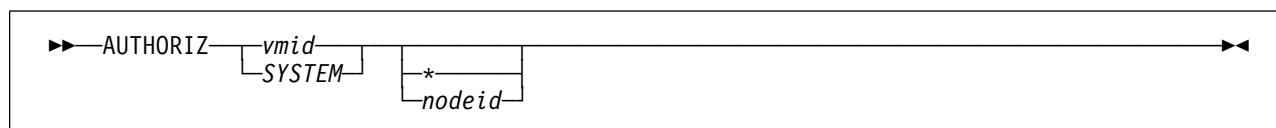
The PVMG configuration file record must contain the APPLN record:

```
APPLN OURTSO TSO1
```

References to the name OURTSO by PVM users are translated to TSO1 application requests. Likewise, references to TSO1 by PVM users will also work if the PVM route table contains an entry for that name.

2. The names appearing in APPLN records are not entered into the selection screen presented to users accessing PVMG through VTAM.

## AUTHORIZ



### Purpose

Use the AUTHORIZ record to identify users who are authorized to issue the restricted subset of PVMG commands and the nodes from which these commands may be issued.

### Operands

*vmid*

is the identification of a VM user (or virtual machine) authorized to issue restricted PVMG commands. Without specifying \* or *nodeid*, this authorization is for the local node only.

#### SYSTEM

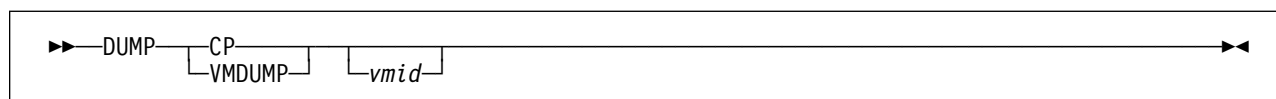
authorizes the PVMG operator to issue commands from nodes (\* or *nodeid*) other than the local node.

\* authorizes the user to issue restricted PVMG commands *from any node*.

*nodeid*

is a specific node where *vmid* is authorized to issue restricted PVMG commands.

## DUMP



### Purpose

Use the DUMP record to indicate whether the dump taken during abnormal termination of PVMG is a CP dump or a VM IPCS dump.

### Operands

#### CP

indicates that on abnormal termination of PVMG the dump is CP DUMP.

#### VMDUMP

indicates that on abnormal termination of PVMG the dump is VMDUMP.

*vmid*

indicates where the dump is to be spooled. The default is written to the printer for a CP dump and spooled to the *vmid* for a VMDUMP.

## Usage Notes

1. The z/VM Dump Viewing Facility require a VMDUMP type dump. Therefore, VMDUMP *must* be specified to use the z/VM Dump Viewing Facility.
2. If the DUMP record is omitted, the default setting is DUMP CP. If more than one DUMP record is entered in the PVMG configuration file, only the last DUMP record in the list is accepted. PVMG will ignore all other DUMP records.

---

## LANG

```
►►—LANG—langid—————►◄
```

## Purpose

Use the LANG record to identify a particular language as the default national language for a node.

During PVMG initialization, the backup national language is considered the national language for a node before the LANG record is processed. It continues as the default national language for the node if the LANG configuration file record is not specified.

## Operands

*langid*

is the identification of the selected language. It is a 1-to-5-character language identifier.

The IBM-provided languages are:

**AMENG** American English

**KANJI** Japanese

**UCENG** Uppercase English

**GER** German

---

## LINK

```
►►—LINK—*—nodeid—type—USER—userid—————►◄
```

## Purpose

Use the LINK record to define the link for connecting with a PVM virtual machine only (PVMG cannot link to another PVMG virtual machine through an IUCV connection). Use separate LINK records for each PVM virtual machine to which a connection is desired.

The same node identifier cannot appear in more than one LINK record.

**Note:** It is very important that the *nodeid* specified in the LINK record be a name that is unique throughout the entire PVM network.

## Operands

\* indicates that a line address is not provided.

*nodeid*

is the node name of the PVM virtual machine to which an IUCV connection is to be provided.

*type*

specifies the type of line driver used to drive the line. The only valid type for PVMG is:

**IUCV** uses the IUCV protocol to communicate with the PVM virtual machine.

**USER** *userid*

indicates that a userid for the PVM virtual machine follows. The *userid* is a 1-to-8-character string identifying the PVM virtual machine to which the IUCV connection is to be made.

## Usage Notes

1. If the link on the PVM side is not active, PVMG will attempt the IUCV connect at 60 second intervals until either communications are established, or a PVMG DROP node command is issued by an operator.

---

## LNODE

```
▶▶—LNODE—nodeid—FORCE—◀◀
```

## Purpose

Use the LNODE record to define the node name that is automatically selected when a user accesses PVMG from an SNA network, bypassing the PVM Selection screen. To go to a different system than the one specified, pass a node name using NODE=*nodeid* as part of the DATA option on the VTAM LOGON command. If any other options are passed using the DATA option on the VTAM LOGON command, they are acted upon by PVMG, but no messages will be issued for errors that are encountered.

## Operands

*nodeid*

is the name of the default selection node.

**FORCE**

forces a session with *nodeid*.

---

## LOCAL

```
▶▶—LOCAL—nodeid—applid—▶▶
```

## Purpose

Use the LOCAL record to define the node by which the PVMG server is known to other nodes in the network, and the application name by which VTAM recognizes PVMG.

## Operands

### *nodeid*

is the name of the local node. The *nodeid* can be 1 to 8 characters. Default is a reserved word and should not be used as the *nodeid*.

### *applid*

is the name by which PVMG will be known to VTAM. The *applid* can be 1 to 8 characters. If not specified, the *nodeid* is used.

## LOGMODE

►►—LOGMODE—*entrydesc*—*entryname*—————◄◄

## Purpose

Use the LOGMODE record to specify VTAM logon mode table entries that are used by PVMG when establishing interactive sessions from PVM to the SNA network.

## Operands

### *entrydesc*

describes the VTAM Logon Mode Table entry. The *entrydesc* must be in the VTAM logmode table. The following operand descriptions are supported:

<i>entrydesc</i>	Operand Description
3277	indicates that the Logon Mode Table Entry Name is for a 3277 terminal
3278-2B	indicates that the Logon Mode Table Entry Name is for a 3278 model 2 terminal
3278-2E	indicates that the Logon Mode Table Entry Name is for a 3278 model 2 terminal with extended features
3278-3B	indicates that the Logon Mode Table Entry Name is for a 3278 model 3 terminal
3278-3E	indicates that the Logon Mode Table Entry Name is for a 3278 model 3 terminal with extended features
3278-4B	indicates that the Logon Mode Table Entry Name is for a 3278 model 4 terminal
3278-4E	indicates that the Logon Mode Table Entry Name is for a 3278 model 4 terminal with extended features
3278-5B	indicates that the Logon Mode Table Entry Name is for a 3278 model 5 terminal
3278-5E	indicates that the Logon Mode Table Entry Name is for a 3278 model 5 terminal with extended features
3279-2B	indicates that the Logon Mode Table Entry Name is for a 3279 model 2 terminal



<i>entrydesc</i>	Operand Description
3279-2E	indicates that the Logon Mode Table Entry Name is for a 3279 model 2 terminal with extended features
3279-3B	indicates that the Logon Mode Table Entry Name is for a 3279 model 3 terminal
3279-3E	indicates that the Logon Mode Table Entry Name is for a 3279 model 3 terminal with extended features
3290	indicates that the Logon Mode Table Entry Name is for a 3290 terminal. The terminal is treated as a 3278 Model 2.

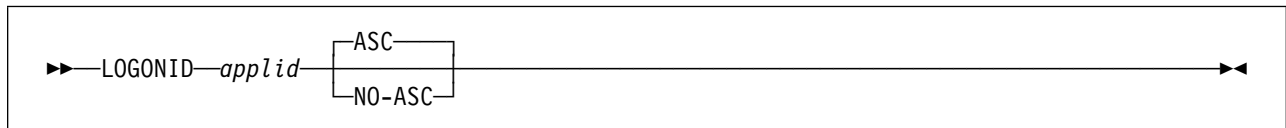
***entryname***

specifies the VTAM Logon Mode Table entry name that applies to the *entrydesc* parameter.

## Usage Notes

1. PVMG looks for each of the *entrydescs* listed in the preceding table during configuration file processing. If one is missing, a message is issued, and PVMG terminates after configuration file processing is complete.

## LOGONID



## Purpose

Use the LOGONID record to define the VTAM APPLID to which the PVMG node is to log the PVM terminals onto when the PVMG node is selected.

## Operands

***applid***

is the name of the application, SAMON, for example, on to which PVMG is to log PVM terminals.

**ASC**

specifies that the application is an Application Selection Concentrator. This is the default if a second operand is not provided.

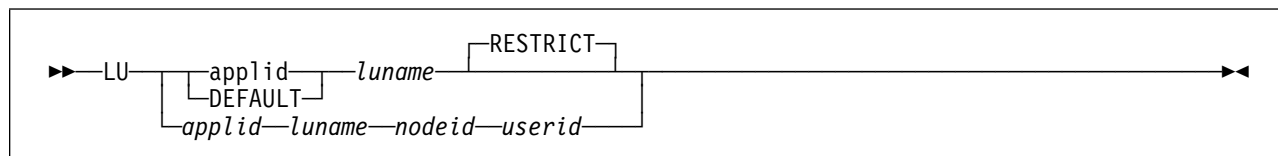
**NO-ASC**

specifies that the application is not an Application Selection Concentrator.

## Usage Notes

1. When SNA users issue the VTAM LOGON command to access a PVM network, the *applid* supplied is the one specified on the PVMG LOCAL configuration file record, not the *applid* on this record.

## LU



## Purpose

Use the LU record to specify LU names that are to be used by PVMG when obtaining interactive VTAM sessions with restricted and secure VTAM APPLIDs. When a PVM user selects an application for a session, PVMG assigns an LU name from the list of LU names in the LU record. If the application has the RESTRICT option, then an available LU is assigned. If the application requires specific users, then the user making the session request must be assigned an LU name in the LU record.

## Operands

### *applid*

is the VTAM APPLID. The *applid* can be 1 to 8 characters.

### *DEFAULT*

the *DEFAULT* operand defines *lunames* to be used for applications that do not have specific *lunames* defined for them. All *LU DEFAULT* records must precede any other LU records for restricted VTAM applications.

### *luname*

is the name of the SNA logical unit associated with the *applid*. The *luname* may be a single name, or it may be a range of SNA logical unit names. The range represents a pool of logical unit names that may be used to access the specified *applid*. If the optional *nodeid* and *userid* are encoded in this record, *luname* must be a single LU. If the third parameter is RESTRICT or omitted, then the second operand can be a range.

The format of a range is given by:

- A base name that also is the lower boundary of the range
- An upper boundary of the range
- A hyphen (-) separates the lower from the upper boundary of the range.

For example,

SAM05000-99

will result in the generation of 100 logical units with addresses SAM05000, SAM05001,...SAM05099.

SAM05050-99

will result in the generation of 50 logical units with addresses SAM05050, SAM05051,...SAM05099.

The following rules apply to ranges:

- The upper boundary must be all digits, and must not exceed a length of three.

- The number of trailing numeric digits in the base must be greater than or equal to the number of numeric digits in the range. For example, the range specification

SAM050A0-99

is not acceptable since the trailing two characters in the base, A0, are not all numeric. The specification

SAM05A20-100

is unacceptable since the trailing three characters in the base, A20, are not all numeric.

- The range of logical unit names that can be generated from a single statement cannot exceed 500.

#### RESTRICT

indicates that only one session may be assigned to one LU. This is the default if no option is specified.

*nodeid*

is the nodeid to which this LU is secured.

*userid*

is the userid to which this LU is secured.

## Usage Notes

1. Unless an application requires specific LU names, or must be secure, one or more LU DEFAULT configuration file records should suffice.
2. The same LU name, or range of LU names, can be specified on multiple restricted LU records for different APPLIDs. However, each LU name can be used by only one user to a single application at a time.
3. An application cannot be both secure and restricted, but an LU name can be used for secure, as well as restricted, applications.
4. If an APPLID must be secure for multiple users, each secure LU record must have a unique LU name associated with it.
5. If an LU is restricted, only one user can use it at a time. Specify a range of restricted LUs in the form:

LU XXX01 XXX0001-50 RESTRICT

50 LUs will be set up to handle 50 users at once, each with their own LU. This saves entering 50 individual LU records.

6. If

LU XXX01 XXX0001 RMTPVM1 JONES

is entered, only user JONES at RMTPVM1 may use the specified LU.

---

## MSGNOH

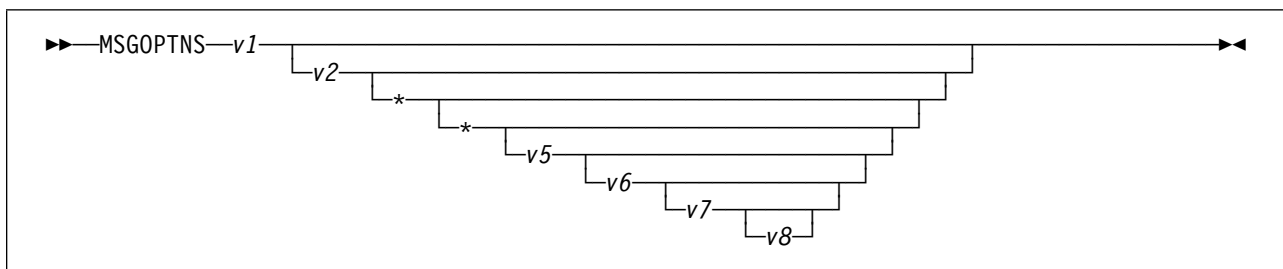
►►MSGNOH◄◄

## Purpose

Use the MSGNOH record to indicate that responses to all commands entered by the CP SMSG command must be returned using the CP MSGNOH command rather than the CP MSG command, unless prohibited by the MSGOPTNS on that record when sending messages to the CP OPERATOR. If there is more than one MSGNOH record in the configuration file, only the last MSGNOH record in the list is accepted. PVMG ignores the other MSGNOH records.

PVMG must have Class B privilege to issue the CP MSGNOH command.

## MSGOPTNS



## Purpose

Use the MSGOPTNS record to indicate what options PVMG will use regarding time stamps and message numbers for messages directed to the PVMG user ID, to users, and to the operator. The installation can choose for each destination whether time stamps and message numbers should be appended to the messages. All parameters (except 3 and 4) must be either a Y for yes or an N for no.

The MSGOPTNS record does not support an audit file, but is compatible with the PVM component's MSGOPTNS record.

## Operands

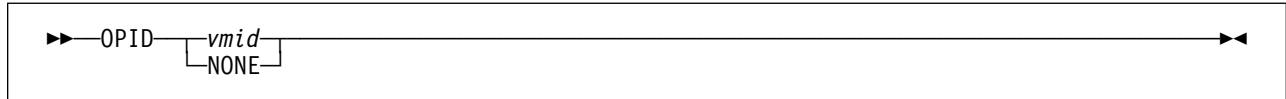
- v1 specifies the time stamps on the messages to the PVMG user ID. The default is N.
- v2 specifies the message numbers on the messages to the PVMG user ID. The default is N.
- \* reserved (use \* as a placeholder). PVMG will ignore any non-blank character(s) used as a placeholder. A placeholder is used to ensure consistency with the PVM MSGOPTNS configuration file record.
- \* reserved (use \* as a placeholder). PVMG will ignore any non-blank character(s) used as a placeholder. A placeholder is used to ensure consistency with the PVM MSGOPTNS configuration file record.
- v5 specifies the time stamps on the messages to users. The default is N.
- v6 specifies the message numbers on the messages to users. The default is N.

*v7* specifies all messages written to the operator. These messages are written by a CP MSG command, overriding the MSGNOH record. The default is N, meaning that the MSGNOH option will take effect.

*v8* specifies the message numbers on the messages to the operator. The default is N.

---

## OPID



## Purpose

Use the OPID record to specify the virtual machine ID that receives messages that normally go to the system operator. Use this record to stop messages from being sent to the system operator.

## Operands

*vmid*

is the identification of the virtual machine to receive operator messages.

**NONE**

suppresses the messages sent to the operator.

## Usage Notes

1. If this record is omitted, operator messages are sent to the OPERATOR user ID.
2. The format of the messages sent to the specified operator depends on the options specified on the MSGOPTNS record and on whether PVMG is authorized to use the CP MSGNOH command.
3. Only one *vmid* may be specified on this record.

## PVMPROP

```
▶▶—PVMPROP—vmid—◀◀
```

### Purpose

Use the PVMPROP record to identify a virtual machine running a programmed operator that receives a copy of PVMG error messages. All messages are transmitted to the virtual machine by the CP Special Message (CP SMSG) Facility. The messages are sent to the PVMPROP virtual machine in the same encoded format as sent to the Command Response Interface. See Chapter 13, “Facilities for Automating PVM and PVMG Operations” on page 145 for more information about PVMPROP.

### Operands

*vmid*

is the virtual machine to receive error messages. All PVMG messages are sent to the *vmid* in encoded form using the CP SMSG command.

## ROUTE

```
▶▶—ROUTE—targetid—nextid  
          DEFAULT—nodeid—▶▶
```

### Purpose

Use the ROUTE record to define the next nodal path to a specified target node from the local PVMG node. For PVMG to PVM communications, routing must be consistently defined on both ends of network paths.

Use the second form of the ROUTE record to define default routing for nodes not defined by LINK or ROUTE records. There can be only one default route record.

### Operands

*targetid*

is the identification of the target system node.

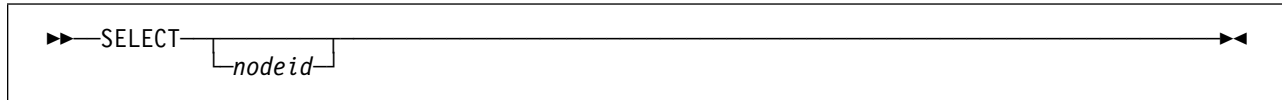
*nextid*

is the identification of the next PVM node defined in a configuration file LINK record, in the routing pattern from the local node to the target system node.

**DEFAULT** *nodeid*

indicates that the *nodeid* is the default node identifier. The *nodeid* is the PVM node through which all unknown nodes are to be routed. This node must be defined in a LINK configuration file record. To avoid difficulties, do not define any PVM or PVMG node identifiers with the name *DEFAULT*.

## SELECT



### Purpose

Use the SELECT record to assign a frequently used node name to a PF key. Specify up to 11 SELECT records to assign 11 PF keys. The first SELECT record *nodeid* is assigned to PF1, the second SELECT record *nodeid* is assigned to PF2, and so forth to PF7. The eighth SELECT record is assigned to PF9 (PF8 is reserved for the SCROLL FORWARD function). The ninth SELECT record is assigned to PF10, and so forth to PF12.

If a SELECT record does not specify the node ID, the PF key associated with this SELECT record is omitted. This allows a PF key to be skipped.

### Operands

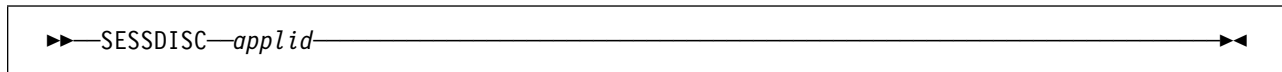
*nodeid*

is the identification of the node assigned to the PF key represented by this SELECT record. The *nodeid* may be 1 to 8 characters long.

### Usage Notes

1. Two SELECT records in the configuration file cannot contain the same node name.

## SESSDISC



### Purpose

Use the SESSDISC record to indicate the VTAM APPLID to which PVMG should issue the VTAM TERMSESS macro with the optional parameter OPTCD=COND, when unexpected PVM session failures occur. If an unexpected PVM session failure occurs for an APPLID not specified on the SESSDISC record, PVMG issues the VTAM TERMSESS macro with OPTCD=UNCOND, to terminate the session.

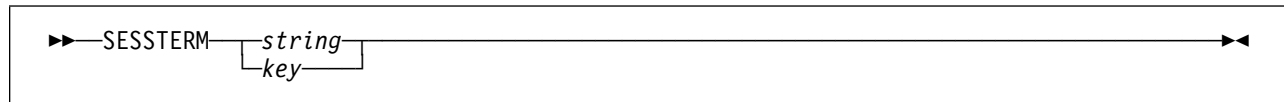
The specific action can occur only when PVMG executes as a secondary logical unit. Determine what session termination actions are provided by an application as the primary logical unit before deciding to specify its APPLID on this record.

### Operands

*applid*

is the name of the VTAM application. The *applid* can be 1 to 8 characters.

## SESSTERM



### Purpose

Use the SESSTERM record to define the default session termination string, *PFn* or *PAn* key used at the local PVMG node for VTAM originating users.

If this record is not entered in the PVMG configuration file, PVMG uses the default string #####.

### Operands

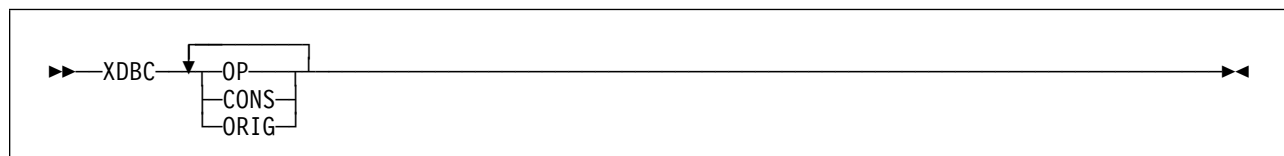
*string*

is a 1-to-8-character, session termination string. Mixed case strings are allowed.

*key*

is the key to assign as the session termination key. Use the form *PFn* or *PAn* to specify the key. For PF keys, *nn* can be a number from 01 (or 1) to 24. For PA keys, *n* can be 1 or 2.

## XDBC



### Purpose

Use the XDBC record to exclude certain users on the local system from receiving, in the national language, any PVMG messages that, in the national language, contain double-byte characters. The identified users receive those messages in the backup language instead. The record may appear anywhere in the file.

The principal reason for the XDBC record is that PVMG cannot tell whether the virtual machine consoles of other virtual machines on a VM system are capable of handling double-byte characters.

### Operands

**OP**

identifies the CP operator.

**CONS**

identifies the PVMG user ID console (display station).



**ORIG**

identifies all virtual machine users on the local VM system who are not PVMG users, other than the CP operator and the PVMG operator. That is, the originators of actions or commands to which PVMG messages are sent in response.

If the LANG record identifies a DBCS language and the PVMG user ID or the CP operator should not receive DBCS messages, the XDBC record should be coded immediately before or after the LANG record.

If PVMG finds an error in the XDBC record, PVMG immediately excludes OP, CONS, and ORIG from receiving DBCS messages, and it is necessary to use the CDBC command to enable them again.



## Appendix F. Setting Up an APPC Link

This appendix describes how to set up an APPC link. The following are sample CP directories, CMS communication directories, configuration files, and PVM profile execs for two PVM nodes, for example, RMTPVM1 and RMTPVM5.

Use these partial samples as models when setting up an APPC link.

SYSTEM -- RMTPVM1	SYSTEM -- RMTPVM5
<p><b>SMITH is a userid on RMTPVM1.</b></p> <p><b>Sample CP Directory for the PVM user ID</b></p> <pre> USER PVM XXXXXXX 5M 8M BG 64 OPTION  BMX MAXCONN 4096 ECMODE IPL CMS IUCV ALLOW IUCV *IDENT RESANY GLOBAL </pre> <p><b>Sample CMS Communications Directory entry for RMTPVM1 (UCOMDIR NAMES)</b></p> <pre> :nick.EAGLE      :luname.PVMGATE1 PVMGATE5                   :tpn.PVM5                   :modename.AVSMODE                   :security.PGM                   :userid.JONES                   :password.IANIAN </pre> <p><b>Partial Configuration File for RMTPVM1 (PVM CONFIG)</b></p> <pre> : LOCAL RMTPVM1 RESID PVM1 LINK EAGLE RMTPVM5 APPC USER SMITH : </pre> <p><b>Partial PVM Profile Exec for RMTPVM1 (PROFILE PVM)</b></p> <pre> : START NODE RMTPVM5 : </pre>	<p><b>JONES is a userid on RMTPVM5.</b></p> <p><b>Sample CP Directory for the PVM user ID</b></p> <pre> USER PVM XXXXXXX 5M 8M BG 64 OPTION  BMX MAXCONN 4096 ECMODE IPL CMS IUCV ALLOW IUCV *IDENT RESANY GLOBAL </pre> <p><b>Sample CMS Communications Directory entry for RMTPVM5 (UCOMDIR NAMES)</b></p> <pre> :nick.HAWK       :luname.PVMGATE5 PVMGATE1                   :tpn.PVM1                   :modename.AVSMODE                   :security.PGM                   :userid.SMITH                   :password.WALTER </pre> <p><b>Partial Configuration File for RMTPVM5 (PVM CONFIG)</b></p> <pre> : LOCAL RMTPVM5 RESID PVM5 LINK HAWK RMTPVM1 APPC USER JONES : </pre> <p><b>Partial PVM Profile Exec for RMTPVM5 (PROFILE PVM)</b></p> <pre> : START NODE RMTPVM1 : </pre>

The following are explanations of the information used in the samples above:

- USER PVM, in the CP directory for PVM on RMTPVM1, identifies PVM as the user ID for that PVM virtual machine.  
USER PVM, in the CP directory for PVM on RMTPVM5, identifies PVM as the user ID for that PVM virtual machine.
- EAGLE, the :nick. entry in UCOMDIR NAMES for RMTPVM1, is a symbolic destination name of the node being connected to, RMTPVM5. EAGLE

corresponds to the parameter on the LINK record in the configuration file for RMTPVM1.

HAWK, the :nick. entry in UCOMDIR NAMES for RMTPVM5, is a symbolic destination name of the node being connected to, RMTPVM1. HAWK corresponds to the parameter on the LINK record in the configuration file for RMTPVM5.

- PVM1GATE and PVM5GATE are the gateway names of the local system and the system to which the connection is being made. PVM1GATE is the gateway name for RMTPVM1, and PVM5GATE is the gateway name for RMTPVM5. These names are the names known to the SNA network.
- PVM5 is the resource name of PVM on system RMTPVM5, which is identified by the RESID record in the configuration file for RMTPVM5. PVM5 is the resource name RMTPVM1 wants to connect to, which is identified by the :tpn. entry in UCOMDIR NAMES for RMTPVM1. If the RESID record is omitted, the resource name would have to match the name on the LOCAL record.

Likewise, PVM1 is the resource name of PVM on system RMTPVM1, which is identified by the RESID record in the configuration file for RMTPVM1. PVM1 is the resource name RMTPVM5 wants to connect to, which is identified by the :tpn. entry in UCOMDIR NAMES for RMTPVM5. If the RESID record is omitted, the resource name would have to match the name on the LOCAL record.

Each resource name must be unique within a PVM network.

- AVSMODE is the mode name. This name is found in the VTAM logon mode table. See a VTAM administrator for the mode name.
- PGM is a keyword identifying the security level. PGM or SAME must be specified.
- JONES, the :userid. entry in UCOMDIR NAMES for RMTPVM1, is a valid user ID on RMTPVM5. This user ID matches the user ID on the LINK record in the configuration file for RMTPVM5.

SMITH, the :userid. entry in UCOMDIR NAMES for RMTPVM5, is a valid user ID on RMTPVM1. This user ID matches the user ID on the LINK record in the configuration file for RMTPVM1.

- IANIAN is the password of JONES at RMTPVM5.  
WALTER is the password of SMITH at RMTPVM1.

## Appendix G. PVM Control Block Formats

The information in this appendix is included to provide further information to help exploitation of the sample access security exits, and for diagnostic purposes.

This appendix contains Product Sensitive Programming Interface information.

### PSPI

#### DVMAAT (Active Application Task Descriptor) Format

Each DVMAAT control block describes an active logical device support task in the PVM (for sessions in which the local PVM is the target) virtual machine.

0	AATFOR				AATBAK			
8	AATDEST							
10	AATDUID							
18	AATUID				AATAD			
20	AALTAD				AAALT			
28	AATLDA				AATREAL		AATCLAS	
30	A*1	A*2	A*3	A*4	AATSTRT			
38	AATICNT				AATOCNT			
40	AATRNODE							
48	AATRADR							
50	A*5	RESERVED						

Field	Length	Use
AATFOR	4	FORWARD POINTER CHAIN
AATBAK	4	BACKWARD POINTER CHAIN
AATDEST	8	DESTINATION NODE ID
AATDUID	8	DESTINATION USER ID
AATUID	4	TASK USERID (PRINTABLE LOGICAL DEVICE ADDRESS)
AATAD	4	ADDRESS OF OWNING TASK (DIAGNOSE DRIVER) TCB
AALTAD	4	ADDRESS OF TCB FOR TASK WHICH IS TO GET DATA BUFFERS
AAALT	4	ADDRESS OF ALT ENTRY
AATLDA	4	LOGICAL DEVICE ADDRESS
AATREAL	2	REAL DEVICE ADDRESS
AATCLAS	2	DEVICE CLASS AND TYPE
A*1 (AATMDL)	1	DEVICE MODEL NUMBER
A*2 (AATTYPE)	1	TYPE FLAGS
Values defined in AATTYPE:		
AATEXTN	X'80'	REMOTE END HAS EXTENDED ATTRIBUTES
AATIOV	X'40'	USER WANTS I/O VERIFICATION
A*3 (AATRFEA)	1	EXTENDED FEATURES
A*4 (AATSTYPE)	1	SESSION TYPE
AATSTRT	4	SESSION START TIME
AATICNT	4	INPUT TRANSACTIONS
AATOCNT	4	OUTPUT TRANSACTIONS
AATRNODE	8	ORIGIN NODE IF R3270
AATRADR	8	EBCDIC REAL TERMINAL ADDRESS
A*5 (AATMARG)	1	MESSAGE ARGUMENT BYTE
	7	RESERVED

**DVMALT (Active Link Table Entry) Format**

Each DVMALT control block defines a link (line driver) to another node.

0	ALTFOR		ALTBK		
8	ALTAUTF		ALTAUTB		
10	ALTAATF		ALTAATB		
18	ALTASK		ALTPORT		
20	ALTNODE				
28	ALDEVAD	ALTNAM	ALACTUS	L*1	L*2
30	ALTREAL	ALTDEL	ALTINBL		
38	ALTOTBL		ALTINBT		
40	ALTOTBT		ALTINTR	ALTOTTR	
48	ALTPSIZ	ALTDDELAV	L*3	L*4	RESERVED
50	ALTGROUP				
58	ALTIADR		ALTDADR		

Field	Length	Use
ALTFOR	4	FORWARD CHAIN POINTER
ALTBK	4	BACKWARD CHAIN POINTER
ALTAUTF	4	CHAIN OF USER TASKS
ALTAUTB	4	CHAIN OF USER TASKS
ALTAATF	4	CHAIN OF APPL TASKS
ALTAATB	4	CHAIN OF APPL TASKS
ALTASK	4	ADDRESS OF OWNING TASK TCB
ALTPORT	4	ADDRESS OF PORT LIST
ALTNODE	8	NODE NAME
ALDEVAD	2	TP LINE ADDRESS
ALTNAM	2	LINK TYPE
ALACTUS	2	NUMBER OF ACTIVE SESSIONS
L*1 (ALTFLG)	1	FLAGS FOR LINK
Values defined in ALTFLG:		
ALTCON	X'80'	LINK WAS OR IS CONNECTED - DETERMINES WHETHER TO RESTART IF LINE FAILS
ALTACTV	X'40'	LINEDRIVER TASK IS STARTED
ALTDOWN	X'20'	INDICATES A DROP HAS BEEN ISSUED
ALTQUIES	X'10'	THE LINE IS QUIESCING
ALTRSTR	X'08'	INDICATES NETWORK LINE RESTARTING
L*2 (ALTTPRMS)	1	TRACING FLAGS
Values defined in ALTTPRMS:		
ALTTRIO	X'80'	I/O TRACING IS ACTIVE
ALTTRDAT	X'40'	DATA TRACING IS ACTIVE
ALTTRWRP	X'20'	USE IN-CORE WRAP TABLE
ALTREAL	2	REAL LINE ADDRESS
ALTDL	2	LINE DELAY IN MILLISECONDS
ALTINBL	4	TRANSMISSION BLOCKS RECEIVED
ALTOTBL	4	TRANSMISSION BLOCKS SENT
ALTINBT	4	BYTES RECEIVED
ALTOTBT	4	BYTES TRANSMITTED
ALTINTR	2	EXTRA RECORDS PER BLOCK SENT
ALTOTTR	2	EXTRA RECORDS PER BLOCK RECEIVED
ALTPSIZ	2	BLOCKSIZE THRESHOLD
ALTDLAV	2	AVERAGED LINK DELAY TIME
L*3 (ALTOPN1)	1	LINK CLASS OPTIONS
Values defined in ALTOPN1:		
ALTOTBND	X'80'	OUT-BOUND SESSIONS ALLOWED
ALTINBND	X'40'	IN-BOUND SESSIONS ALLOWED
ALTNTIME	X'20'	LINK SUPPORTS LINK TIMING
ALTOGRP	X'10'	LINK SUPPORTS LINE GROUPING
L*4 (ALTOPTN2)	1	LINK FEATURE OPTIONS

Field	Length	Use
	Values defined in ALTOPTN2:	
ALTDIA98	X'80'	DIAG 98 WANTED FOR LINK
ALTPACKF	X'40'	PACKING IS REQUESTED
ALTNADDR	X'20'	LINK ADDRESS NOT TO BE CHECKED
ALTNDEF	X'10'	LINK NOT VALID FOR DEFINE COMMAND
ALTNMOD	X'08'	LINK TYPE CANNOT BE ALTERED
	2	RESERVED
ALTGROUP	8	GROUP NAME IF PART OF LINE GROUP
ALTIADR	4	ADDRESS OF I/O TRACE BUFFER
ALTDADR	4	ADDRESS OF DATA TRACE BUFFER

**DVMAUT (Active User Interface Link Descriptor) Format**

Each DVMAUT control block describes an active user support task in this PVM.

0	AUTFOR		AUTBAK			
8	AUTUID					
10	AUTDEST					
18	AUTDUID					
20	AUTAD		AULTAD			
28	AUALT		AUTREAL		AUTNAM	
30	AUTUB	AUTCLAS	U*1	U*2	U*3	U*4
38	AUTSTRT		AUTICNT			
40	AUTOCNT		U*5	U*6	RESERVED	
48	AUTRADR (AUTVMID)					
50	AUTVMID					

Each DVMAUT extension area for the MPVM user support task only applies to an individual MPVM session.

58	AUTSNICK				
60	AUTSBFOR			AUTPHAND	
68	AUTSEGL			AUTSEGB	
70	AUTUPB			AUTSCFOR	
78	AUTSCTIM				
80	AUTSUFOR			AUTSUTIM(8)	
88	AUTSUTIM(cont)			AUTPAFOR	
90	AUTPATIM				
98	AUTPACE			AUTSUORG	
A0	AUTSCR			AUTROWS	AUTCOLS
A8	U*7	U*8	U*9	U*10	

Field	Length	Use
AUTFOR	4	FORWARD CHAIN POINTER
AUTBAK	4	BACKWARD CHAIN POINTER
AUTUID	8	USER ID
AUTDEST	8	DESTINATION NODE ID
AUTDUID	8	DESTINATION USER ID
AUTAD	4	ADDRESS OF OWNING TASK (INTERFACE DRIVER) TCB
AULTAD	4	ADDRESS OF TCB FOR TASK WHICH IS TO GET DATA BUFFERS
AUALT	4	ADDRESS OF ALT ENTRY
AUTREAL	2	REAL DEVICE ADDRESS
AUTNAM	2	MNEMONIC LINK TYPE
AUTUB	2	TUBE DEVICE ADDRESS
AUTCLAS	2	REAL DEVICE CLASS AND TYPE
U*1 (AUTMDL)	1	REAL DEVICE MODEL NUMBER
U*2 (AUTTYPE)	1	TYPE OF INTERFACE TASK

Values defined in AUTTYPE:



Field	Length	Use
AUTGRAF	X'80'	DIRECT GRAPHIC INTERFACE
	X'40'	RESERVED
AUTAAT	X'20'	REMOTE END IS A LOGICAL DEVICE
AUTIUUV	X'10'	IUCV SESSION
AUTEXTN	X'08'	DISPLAY HAS EXTENDED FEATURES
	X'04'	RESERVED
AUTPRT	X'02'	THIS IS A PRINTER TASK
	X'01'	RESERVED
U*3 (AUTSTAT)	1	STATUS OF INTERFACE TASK
	Values defined in AUTSTAT:	
AUTWAIT	X'80'	AWAITING SESSION INITIATION
AUTSEL	X'40'	DISPLAYING SELECTION
AUTCONN	X'20'	SESSION IS ACTIVE
AUTNRDY	X'10'	PRINTER IN NOT READY STATE
AUTSGNON	X'04'	IN PROCESS OF REMOTE SIGNON
AUTIOV	X'02'	USER WANTS I/O VERIFICATION
AUTOFLG	X'01'	THIS IS ORIGIN OF SESSION
U*4 (AUTRFEA)	1	REAL EXTENDED FEATURES
	Values defined in AUTRFEA:	
AUTRECOL	X'80'	DEVICE HAS EXTENDED COLOR
AUTREHLT	X'40'	DEVICE HAS EXTENDED HIGHLIGHTING
AUTRPSS	X'20'	DEVICE HAS PROGRAMMED SYMBOL SETS
AUTDBCS	X'10'	DEVICE SUPPORTS DOUBLE BYTE CHARACTERS
AUTSTRT	4	SESSION START TIME (TOP TOD)
AUTICNT	4	INPUT TRANSACTION COUNT
AUTOCNT	4	OUTPUT TRANSACTION COUNT
U*5 (AUTSTYPE)	1	SESSION TYPE
U*6 (AUTMARG)	1	MESSAGE ARGUMENT BYTE
	2	RESERVED
AUTRADR (AUTVMID)	8	EBCDIC REAL TERMINAL ADDRESS OR IUCV ORIGIN VMID
AUTTVMID	8	IUCV DESTINATION VMID OR MPVM USER'S VMID
AUTSNICK	8	THE MPVM USER'S SESSION NICKNAME
AUTSBFOR	4	NEXT MPVM USER SESSION BLOCK
AUTPHAND	4	MPVM'S SESSION HANDLE
AUTSEGL	4	SEGMENTED DATA LENGTH
AUTSEGB	4	SEGMENTED DATA BUFFERS QUEUE
AUTUPB	4	SCREEN UPDATE BUFFERS QUEUE
AUTSCFOR	4	NEXT TIMER IN CHAIN AFTER SCREEN UPDATE
AUTSCTIM	8	SCREEN UPDATE TIMER VALUE AND CODE
AUTSUFOR	4	NEXT TIMER IN CHAIN AFTER SUSPEND TIMER
AUTSUTIM	8	SUSPEND TIMER VALUE AND CODE
AUTPAFOR	4	NEXT TIMER IN CHAIN AFTER PACING TIMER
AUTPATIM	8	PACING TIMER VALUE AND CODE
AUTPACE	4	PACING OVERFLOW QUEUE OF BUFFERS
AUTSUORG	4	ORIGINAL SUSPEND TIME VALUE WHEN FIRST SET
AUTSCR	4	ADDRESS OF COMPRESSED BUFFER OR SCREEN IMAGE DSECT
AUTROWS	2	NUMBER OF ROWS ON USER'S SCREEN
AUTCOLS	2	NUMBER OF COLUMNS ON USER'S SCREEN
U*7 (AUTPCNT)	1	BUFFERS SINCE LAST PACING
U*8 (AUTPNUM)	1	UNANSWERED PACING COUNT
U*9 (AUTSES1)	1	MPVM USER FLAGS
	Values defined in AUTSES1:	
AUTPAC	X'80'	PACING ACTIVE
AUTWFSC	X'40'	WAITING FOR SESSION TO COMPLETE OR FOR A TERMINATE RECORD
AUTSUS	X'20'	SESSION IS SUSPENDED
AUTST	X'10'	SCREEN FOR SESSION BEING TRANSFERRED
AUTTERM	X'08'	SESSION BEING TERMINATED
AUTSS1	X'04'	FIRST SUSPEND WARNING SET
AUTSS2	X'02'	SECOND SUSPEND WARNING SENT

## PVM Control Block Formats

Field	Length	Use
AUTPCR	X'01'	POSSIBLE CHAINED READ
U*10 (AUTSES2)	1	MPVM USER FLAGS
	Values defined in AUTSES2:	
AUTTDISC	X'80'	USER AND SESSION ARE TEMPORARILY DISCONNECTED
AUTXPAND	X'40'	SCREEN FOR THIS SESSION IS EXPANDED
AUTNONOT	X'20'	NO NOTIFICATION TO MPVM USER FOR SESSION TERMINATE
AUTREPLY	X'10'	NEED TO DO AN IUCV REPLY TO MPVM
AUTNET	X'08'	NET WILL HANDLE SESSION NOW
AUTPRLD	X'04'	PREVIOUS LOAD/COMPRESS PAIR DONE ON CURRENT SCREEN IMAGE

### DVMBUF (Internal Buffer Header)

This data area contains a description of the header portion of the buffered data that is transferred between tasks within the PVM network.

0	BUFTSK	S*1	S*2	BUFLEN
8	BUFORIG			
10	BUFOUID			
18	BUFDEST			
20	BUFDUID			
28	S*3	S*4	BUFSPEC	BUFDATA →

Field	Length	Use
BUFTSK	4	TCB ADDRESS OF OWNING TASK
S*1 (BUFTYPE)	1	RECORD TYPE
	Values defined in BUFTYPE:	
BUFTCTL	X'80'	NETWORK CONTROL RECORD
BUFTIMER	X'40'	LINK TIMING RECORD
BUFTGDAT	X'20'	GRAPHIC DATA RECORD
BUFTDATA	X'10'	NON-GRAPHIC DATA RECORD
BUFTCMD	X'04'	LOCAL COMMAND BUFFER
S*2 (BUFSTYP)	1	RECORD SUBTYPE
	Values defined in BUFSTYP when BUFTYPE= X'80':	
BUFSNET	X'20'	NETWORK PATH CONTROL
BUFSIGN	X'10'	SESSION INITIATE REQUEST
BUFSGNR	X'08'	SESSION COMPLETE
BUFSOFF	X'04'	SESSION TERMINATION
BUFCMSG	X'02'	'CMD', 'MSG', AND REPLIES
	Values defined in BUFSTYP when BUFTYPE= X'40':	
BUFRETRN	X'40'	THIS IS THE RETURN BUFFER
	Values defined in BUFSTYP when BUFTYPE= X'20':	
GRAFRDBF	X'20'	DATA WAS RETURNED FROM A READ-BUFFER COMMAND
GRAFMSG	X'10'	MESSAGE TEXT TO BE DISPLAYED ON THE SCREEN
GPRTSTAT	X'08'	STATUS RETURNED FROM I/O
GSRDBUF	X'04'	SHORT READ-BUFFER REQUEST
GSTATRQ	X'02'	STATUS MUST BE RETURNED
GCHNCCW	X'01'	PART OF COMMAND CHAIN SEQUENCE
	Values defined in BUFSTYP when BUFTYPE= X'10' and bit 0 of BUFLG1= 0:	
BUFRECV	X'20'	IUCV REPLY
	Values defined in BUFSTYP when BUFTYPE= X'10' and bit 0 of BUFLG1= 1:	
BUFRJCT	X'40'	IUCV REJECT OF TWO-WAY SEND
BUFRECV	X'20'	IUCV REPLY
BUFPRTY	X'10'	DATA IS PRIORITY DATA
BUFPMSG	X'08'	DATA IS VIA PRMDATA
BUFOQUIS	X'04'	THIS IS AN IUCV QUIESCE
BUFORESU	X'04'	THIS IS AN IUCV RESUME
BUFLEN	2	LENGTH OF DATA IN BUFFER

Field	Length	Use
BUFORIG	8	ORIGIN NODE ID
BUFOUID	8	ORIGIN USER ID
BUFDEST	8	DESTINATION NODE ID
BUFDUID	8	DESTINATION USER ID
S*3 (BUFFLG1)	1	FLAG BYTE
	Values defined in BUFFLG1:	
BUFSFORM	X'80'	NEW FORMAT SESSION PACKET
BUFDIRF	X'40'	PACKET IS FROM SESSION ORIGIN
BUFPACE	X'20'	PACING FLAG
BUFAPAC	X'10'	RESPONSE TO PACING
BUFSEGF	X'0F'	NUMBER OF 256 BYTE SEGMENTS IN BUFFER
	Values for commands and messages:	
BUFNLSR4	X'80'	THE ORIGIN IS RELEASE 4
BUFENCOD	X'40'	BUFFER CONTAINS AN ENCODED MESSAGE
S*4 (BUFFLG2)	1	SPANNED RECORD FLAGS
	Values defined in BUFFLG2:	
BUFSFLG	X'80'	THIS IS A SPANNED RECORD
BUFSPFE	X'40'	THIS IS THE LAST RECORD SEGMENT
BUFSCNT	X'3F'	SPANNED RECORD NUMBER 0-63
BUFSPEC	2	SPECIAL DEFINITION
	Values defined in BUFSPEC for COMMAND packets (BUFTYPE= X'80' and BUFSTYP= X'02'):	
SYSCMD	X'40'	COMMAND FROM CONSOLE
CMDRPLY	X'20'	REPLY TO COMMAND
CPCMD	X'10'	CP COMMAND
SYSUSER	X'08'	SYSTEM USER
CMDSMSG	X'04'	REPLY BY SMSG
BUFSIGNA	X'02'	SIGNATURE SET IN BUFTAG
BUFSUBST	X'01'	SPECIAL SUBSET COMMAND
BUFDATA	—	START OF DATA AREA
	Buffer overlay page spool buffer pointers	
BUFTRPSI	4	Printer subchannel identifier
—	4	Reserved
BUFTRPRT	2	Address of printer
BUFLEFT	2	Length left in page
BUFCNXT	4	Address of next slot
PREVCCW	4	Address of last ccw
BUFCCW1	8	First CCW in chain

### **DVMCOMN (PVM Multitasker Common Area) Format**

The DVMCOMN control block contains information commonly required by PVM tasks.

0	BUFRTCB		CONSTCB	
8	NETWTCB		ALTANC	
10	ROUTCNT		ROUTANC	
18	LOCID			
20	USRLIST		SYSMHDR	
28	SYSMHDR (cont.)		COMSPNM	
30	COMSANC		COMSTRT	
38	COMLEN		COMNBUFFS	
40	COMNPOOL		COMBFLG	
48	COMTIMOT		COMNDISC	
50	COMADFN			
58	COMADFT			
60	COMADFM	COMNPROT	COMPID	
68				LASTMSG
70				
78	LASTCMD			
80	COMSNODE		COMLMSG	
88	STATTCB		COMNSTAT	
90	SYSVMID			
98	TRACNODE			
A0	TRACUSER			
A8	SYSROUT			
B0	PROPV MID			
B8	SYSOPID			
C0	SYSZONE		COMNAUTO	
C8	COMNAUDN		SYSSCPN	SYSPP
D0	SYSBUFHI		C*1	C*2
D8	SYSSGND A			
E0	SYSLNODE			
E8	SYSRESID			
F0	SUBSANC		SUBSCNT	
F8	GRPANC		COMIUCV	
100	COMNUSR1		COMNUSR2	
108	COMNUSR3		ILTANC	
110	SLOTADDR		HASTSIZE	
118	NSCTSLOT		NSCTUSED	
120	COMIMAX		COMUTRAC	
128	C*3	COMNREL	RSVDWRD1	
130	TCPCTLBK			
138	SYSSCHIB			
158				
160				RSVDWRD3
168	SYSORB			
170				
178				
180				
188	SYSIRB			
1E0	SYSTRORB			
1E8				
1F0				
1F8				
200				
208	SYSTRIRB			
260				
268				
	CONSID		PRTSID	

## PVM Control Block Formats

Field	Length	Use	
BUFRTCB	4	BUFFER TCB ADDRESS	
CONSTCB	4	CONSOLE INPUT TCB ADDRESS	
NETWTCB	4	NETWORK ROUTER TCB ADDRESS	
ALTANC	4	ALT ANCHOR	
ROUTCNT	4	NUMBER OF ENTRIES IN ROUT TABLE	
ROUTANC	4	ADDRESS OF ROUTE TABLE	
LOCID	8	LOCAL NODE ID	
USRLIST	4	ADDRESS OF USER CONNECT LIST	
SYSMHDR	8	CP MESSAGE COMMAND OP CODE	
COMSPNM	4	NUMBER OF ENTRIES IN SYSTEM PROGRAMMER LIST	
COMSANC	4	ADDRESS OF SYSTEM PROGRAMMER LIST	
COMSTRT	4	ADDRESS OF START OF TABLE AREA	
COMLEN	4	LENGTH OF TABLE AREA	
COMNBUFFS	4	NUMBER OF BUFFER PAGES	
COMNPOOL	4	ADDRESS OF BUFFER POOL START	
COMBFLG	4	ADDRESS OF BUFFER FLAG TABLE	
COMTIMOT	4	TP LINE TIMEOUT CHECK TIME	
COMNDISC	4	USER TEMP DISCONNECT TIME	
COMADFN	8	FILENAME OF AUDIT FILE	
COMADFT	8	FILETYPE OF AUDIT FILE	
COMADFM	2	FILEMODE OF AUDIT FILE	
COMNPROT	3	LINE DRIVER PROTOCOL LEVEL	
COMPID	9	COMPONENT ID	
LASTMSG	10	LAST LOCAL PVM MESSAGE ISSUED	
LASTCMD	8	LAST PVM COMMAND ISSUED	
COMSNODE	4	POINTER TO LOCAL NODES	
COMLMSG	4	POINTER TO LOGMSG	
STATTCB	4	TCB OF STATISTICS TASK (DVMSTA)	
COMNSTAT	4	STATISTICS TIME INTERVAL	
SYSVMID	8	SYSTEM VMID	
TRACNODE	8	TRACE NODENAME	
TRACUSER	8	TRACE USERID	
SYSROUT	8	DEFAULT LINK ROUTING	
PROPVVID	8	DEFAULT VMID FOR PVMPROP	
SYSOPID	8	DEFAULT VMID FOR OP MESSAGES	
SYSZONE	4	TIME ZONE OFFSET IN MINUTES	
COMNAUTO	4	AUTO EXEC TIME INTERVAL	
COMNAUDN	4	MAXIMUM RECORDS FOR AUDIT FILE	
SYSSCPN	2	SYSTEM NAME FROM DIAG 0	
SYSPP	2	FIRST 16 BITS OF PP FLAGS	
SYSBUFHI	4	HIGH BUFFER PAGE ALLOCATED	
C*1 (COMNFLG1)	1	MESSAGE OPTIONS FLAGS	
	Values defined in COMNFLG1:		
	CFLG1CT	X'80'	CONSOLE - TIMESTAMP
	CFLG1CM	X'40'	CONSOLE - MESSAGE NUMBER
	CFLG1AT	X'20'	AUDIT - TIMESTAMP
	CFLG1AM	X'10'	AUDIT - MESSAGE NUMBER
	CFLG1UT	X'08'	USER - TIMESTAMP
	CFLG1UM	X'04'	USER - MESSAGE NUMBER
	CFLG1ON	X'02'	OPID - MSG, NOT MSGNOH
	CFLG1OM	X'01'	OPID - MESSAGE NUMBERS
C*2 (SYSCOMPT)	1	COMPATIBILITY LEVEL	
SYSSGNLN	2	SESSION TERMINATE LENGTH	
SYSSGNDA	8	SESSION TERM STRING	
SYSLNODE	8	LOCAL NODE SELECT	
SYSRESID	8	LOCAL RESOURCE ID	
SUBSANC	4	ANCHOR FOR SUBSET COMMANDS	
SUBSCNT	4	NUMBER OF SUBSET ENTRIES	
GRPANC	4	ANCHOR FOR LINE GROUPING	

Field	Length	Use
COMIUCV	4	ADDRESS OF IUCV SLOT TABLE
COMNUSR1	4	USER AREA
COMNUSR2	4	USER AREA
COMNUSR3	4	USER AREA
ILTANC	4	ANCHOR FOR PCCF TASKS
SLOTADDR	4	ADDRESS OF HASH SLOT TABLE
HASTSIZE	4	SIZE IN BYTES OF SLOT TABLE
NSCTSLOT	4	NUMBER OF 6-BYTE SLOTS
NSCTUSED	4	NUMBER OF USED SLOTS
COMIMAX	4	MAXIMUM NUMBER OF IUCV/APPC PATHS
COMUTRAC	4	ADDRESS OF USER TRACING BUFFER
C*3 (COMNVER)	1	PVM VERSION NUMBER
COMNREL	3	PVM RELEASE, SERVICE LEVEL
RSVDWRD1	4	RESERVED FULL WORD
TCPCTLBK	4	Address of TCP/IP Control Block
SYSSCHIB	52	Subchannel Information Block
RSVDWRD3	4	Reserved
SYSORB	32	Operation Request Block
SYSIRB	96	Interruption Response Block
SYSTRORB	32	Trace Operation Request Block
SYSTRIRB	96	Trace Interruption Response Block
CONSID	4	Console subchannel identifier
PRTSID	4	Trace printer subchannel identifier

### DVMROUT (Entry in Route List) Format

Each DVMROUT control block identifies the target node, followed by the node through which the routing occurs.

0	ROUTERM
8	ROUTNEXT

Field	Length	Use
ROUTERM	8	TARGET NODE NAME
ROUTNEXT	8	NEXT NODE IN LINE

**DVMSYSB (PVM Multitasker System Control Block) Format**

DVMSYSB is the main communication control block for the multitasking supervisor.

0	SYSRUN				SYSWAIT			
8	SYSTIM				SYSDEV			
10	SYSPQE				SPFLGP			
18	SYSPQPAG		SYSEVLN		SYSEVMX		SYSAUDMX	
20	SCHBUSY				SYSEXT			
28	SYSEXTLN		SYSEXTMX		SYSSVC			
30	SEXTDEF				STRACAD			
38	MABORT				MPOST			
40	MSCHED				MATCHK			
48	MGETPQE				MPUTPQE			
50	CMSSVCP							
58	CMSEXTP							
60	CMSIOP							
68	MSVCNP							
70	MEXTNP							
78	MIONP							
80	VMCMPTR				SPQESAV			
88	SYSSAV							
96	SYSSAV2							
104	STRACBUF							
160	SWTRAD				S*1	S*2	S*3	S*4
168	CMSPGMP							
170	MPGMNP							
178	SYSNDMP							
180	SYSNDID							
188	STRCLGS		STIOLW		STIOHI		STIUCLOW	
190	STIUCHI		STLDSLOW		STLDSHI		SAPPCPTH	
198	S*5	S*6	S*7	S*8	SYSLANGS			
1A0	SYSNATL				SYSNETL			
1A8	SYSAAMTD				SYSEAMTD			
1B0	SYSAPMTD				SYSEPMTD			
1B8	SYSA000E				SYSE000E			
1C0	SYSTYPCS				RESERVED			
1C8	RESERVED				RESERVED			
1D0	RESERVED							



Field	Length	Use
SYSRUN	4	RUN Q ANCHOR
SYSWAIT	4	WAIT Q ANCHOR
SYSTIM	4	TIMER Q ANCHOR
SYSDEV	4	POINTER TO SYSTEM DEVICE TABLE
SYSPQE	4	POINTER TO PQE POOL
SPFLGP	4	POINTER TO PQE POOL FLAGS
SYSPQPAG	2	NUMBER OF PQE PAGES
SYSDEVLN	2	DEVICE TABLE CURRENT ENTRY COUNT
SYSDEVMX	2	MAXIMUM DEVICE TABLE ENTRY COUNT
SYSAUDMX	2	MAX AUDIT FILE RECORD LENGTH
SCHBUSY	4	CHANNEL BUSY BITS FOR IOS
SYSEXT	4	POINTER TO EXT HANDLE TABLE
SYSEXTLN	2	CURRENT EXTTAB ENTRY COUNT
SYSEXTMX	2	MAXIMUM EXTTAB ENTRY COUNT
SYSSVC	4	POINTER TO SVC TABLE
SEXTDEF	4	TCB ADDRESS FOR EXT DEFAULTS
STRACAD	4	ADDRESS OF TRACE ROUTINE
MABORT	4	ADDRESS OF ABORT ROUTINE
MPOST	4	ADDRESS OF POST ROUTINE
MSCHED	4	ADDRESS OF SCHEDULER
MATTCHK	4	ADDRESS OF CHECK IF ATTACHED ROUTINE
MGETPQE	4	ADDRESS OF PQE GET ROUTINE
MPUTPQE	4	ADDRESS OF PQE PUT ROUTINE
CMSSVCP	8	CMS SVC NEW PSW
CMSEXT	8	CMS EXTERNAL NEW PSW
CMSIOP	8	CMS I/O NEW PSW
MSVCNP	8	MULTITASKING SUPERVISOR SVC NEW PSW
MEXTNP	8	MULTITASKER EXTERNAL NEW PSW
MIONP	8	MULTITASKER I/O NEW PSW
VMCMPTR	4	POINTER TO VMCF INTERRUPT ELEMENT
SPQESAV	4	POINTER TO SAVED PQE
SYSSAV	64	REGISTER SAVE AREA FOR SUPERVISOR ROUTINES
SYSSAV2	64	ANOTHER REGISTER SAVE AREA FOR SUPERVISOR ROUTINES
STRACBUF	88	TRACE BUFFER
SWTRAD	4	ADDRESS OF OUTPUT WRITER TASK
S*1 (SYSFLAG)	1	SYSTEM FLAGS
Values defined in SYSFLAG:		
SYSETRAC	X'80'	TRACE USING ETRACE
SYSCTRAC	X'40'	TRACE TO VIRTUAL CONSOLE
SYSPTRAC	X'20'	TRACE TO PRINTER (00F)
SYSTRACE	X'E0'	ANY TRACING IS ACTIVE
SMSSSW	X'10'	USE *MSS FACILITY
SYSTRAP	X'08'	ENABLE TRAP CODE
SYSSHUT	X'04'	SYSTEM IS SHUTTING DOWN
SYSQUIS	X'02'	SYSTEM IS IN A QUIESCE STATE
SYSAUDIT	X'01'	LOG THE CONSOLE
S*2 (SYSFLAG2)	1	SYSTEM FLAGS
Values defined in SYSFLAG2:		
SAUTONOD	X'80'	FORCE AUTO-NODE SELECT
SRESUM	X'40'	TASK-MUST-COMPLETE FLAG
SYSDISC	X'20'	CONSOLE IS DISCONNECTED
SRECMS	X'10'	TASK RESUME FROM CMS
SYSXA	X'04'	VM/XA RUNNING
SYSXASP	X'02'	VM/XA SP RUNNING

## PVM Control Block Formats

Field	Length	Use
S*3 (SYSFLAG3)	1	SYSTEM FLAGS
	Values define in SYSFLAG3:	
SIUCVSW	X'80'	IUCVIS ALLOWED
SDIAG98	X'40'	DIAG 98 IS ALLOWED
SLDSCSW	X'20'	CP INDICATE READ CHAINED TO WRITE
SIUCLIST	X'10'	LIST FORM ALLOWED WITH IUCV
S327EDS	X'08'	CP HAS EXTENDED LSDF
SRESMGR	X'04'	PVM is *IDENT RESOURCE MANAGER
SRESCON	X'02'	SUCCESSFUL IUCV CONNECT *IDENT
SNAPPC	X'01'	SYSTEM CAN'T HANDLE APPC
S*4 (SYSXDBC)	1	DBCS EXCLUSIONS FLAGS FOR PSWS
CMSPGMP	8	CMS PGM NEW PSW
MPGMNP	8	MULTITASKER PROGRAM NEW PSW
SYSNDMP	8	TYPE OF DUMP
SYSNDID	8	VMID FOR DUMP
STRCFLGS	2	TRACING FLAGS
	Values defined in STRCFLGS:	
SDISPTR	X'80'	TRACE TASK DISPATCHING
SPOSTTR	X'40'	TRACE POSTING OF TASKS
SIOTR	X'20'	TRACE START I/O's AND I/O INTERRUPTS
SVMCFTR	X'10'	TRACE VMCF DIAGNOSE AND INTERRUPTS
SLDSFTR	X'08'	TRACE LSDF DIAGNOSE AND INTERRUPTS
SIUCVTR	X'04'	TRACE IUCV INSTRUCTIONS AND INTERRUPTS
STIMRTR	X'02'	TRACE TIMER REQUESTS AND INTERRUPTS
SSTOTR	X'01'	TRACE STORAGE ALLOCATIONS
	Values defined in STRCFLGS+1:	
SATTCTR	X'80'	TRACE ATTACH AND DETACH
SSVCTR	X'40'	TRACE SVC INSTRUCTIONS
SDSTOTR	X'20'	TRACE DATA BUFFER STORAGE ALLOCATION
STIOLOW	2	LOW LIMIT FOR I/O TRACING
STIOHI	2	HIGH LIMIT FOR I/O TRACING
STIUCLOW	2	LOW LIMIT FOR IUCV TRACING
STIUCHI	2	HIGH LIMIT FOR IUCV TRACING
STLDSLOW	2	LOW LIMIT FOR LDSF TRACING
STLDSHI	2	HIGH LIMIT FOR LDSF TRACING
SAPPCPTH	2	IUCV PATHID TO LOCAL *IDENT RESIDE
S*5 (SYSFLAG5)	1	SYSTEM FLAGS
	Values defined in SYSFLAG5:	
SDIAGEC	X'80'	SYSTEM SUPPORTS DIAG X'EC'
SISFSW	X'40'	SYSTEM SUPPORTS ISF
S*6 (SYSFLAG6)	1	RESERVED
S*7 (SYSFLAG7)	1	RESERVED
S*8 (SYSFLAG8)	1	RESERVED
SYSLANGS	4	BASE OF ENABLED LANGUAGE LIST
SYSNATL	4	NATIONAL LANGUAGE BASE ADDRESS
SYSNETL	4	BACKUP LANGUAGE BASE ADDRESS
SYSAAMTD	4	NATIONAL LANGUAGE A.M. TIMESTAMP
SYSEAMTD	4	BACKUP LANGUAGE A.M. TIMESTAMP
SYSAPMTD	4	NATIONAL LANGUAGE P.M. TIMESTAMP
SYSEPMTD	4	BACKUP LANGUAGE P.M. TIMESTAMP
SYS000E	4	ABR's 000E NATIONAL MESSAGE ADDRESS
SYSE000E	4	ABR's 000E NETWORK MESSAGE ADDRESS
SYSTYPCS	4	10, 50, 68, & 127 TYPE CHARACTERS
RESERV1	4	RESERVED
RESERV2	4	RESERVED
RESERV3	4	RESERVED
RESERV4	4	RESERVED

## DVMTCB

Each task in PVM has a task control block (DVMTCB) with which it is associated. The DVMTCB has two parts:

1. A common area common to each task, and
2. An extension area that is defined by, and used exclusively by, each task.

The format of the common area is:

0	TFOR	TBAK
8	TNAME	
10	TPTASK	
18	TPARG	
40		TPFLAG
48	WFLAG	PCHAIN
50	TPSW	
58	TREGS	
98	TLINK	RESERVED
A0	TUSER	
B8	TSYSPIRA	TSESSCB
C0	TTIMER	
C8	TFW	TBW
D0	TPARM	
E0	TCBID	
E8	TPRIOR	T*1 T*2

Figure 37. DVMTCB Control Block Format

Field	Length	Use
TFOR	4	FORWARD CHAIN POINTER
TBAK	4	BACKWARD CHAIN POINTER
TNAME	8	LINK NAME
TPTASK	4	TCB ADDRESS OF POSTING TASK
TPARG	48	POSTED ARGUMENT
TPFLAG	4	FLAG FOR CURRENT POST
WFLAG	4	WAIT ENABLED FLAGS
PCHAIN	4	POSTED ELEMENT CHAIN (contents may change whenever system is enabled)
TPSW	8	OLD PSW FOR THIS TASK
TREGS	64	REGISTER SAVE AREA FOR THIS TASK
TLINK	4	ADDRESS OF THE LINK CONTROL BLOCK FOR THIS TASK
RESERVED	4	RESERVED
TUSER	24	USER AREA
TSYSPIRA	4	ADDRESS OF PROGRAM CHECK INTERRUPT ROUTINE ADDRESS
TSESSCB	4	ADDRESS OF SESSION CONTROL BLOCK
TTIMER	8	CLOCK VALUE FOR TIMER INTERRUPT
TFW	4	TIMER FORWARD CHAIN
TBW	4	TIMER BACKWARD CHAIN
TPARM	16	ARGUMENTS TO MULTITASKING SUPERVISOR SERVICE ROUTINES
TCBID	8	TASK ID
TPRIOR	2	TASK SCHEDULING PRIORITY

## PVM Control Block Formats

Field	Length		Use
T*1 (TATTR)	1		TASK ATTRIBUTES
	Values defined in TATTR:		
	TDMSFR	X'80'	MUST DMSFRE TCB WHEN DONE
	TWAIT	X'40'	TASK NOW WAITING
	TPEND	X'20'	TASK EXECUTION PENDING
	TIOW	X'10'	TASK WAITING - SIO BUSY
T*2	1		RESERVED

The format of the TCB extension for the network session manager (DVMNET) is:

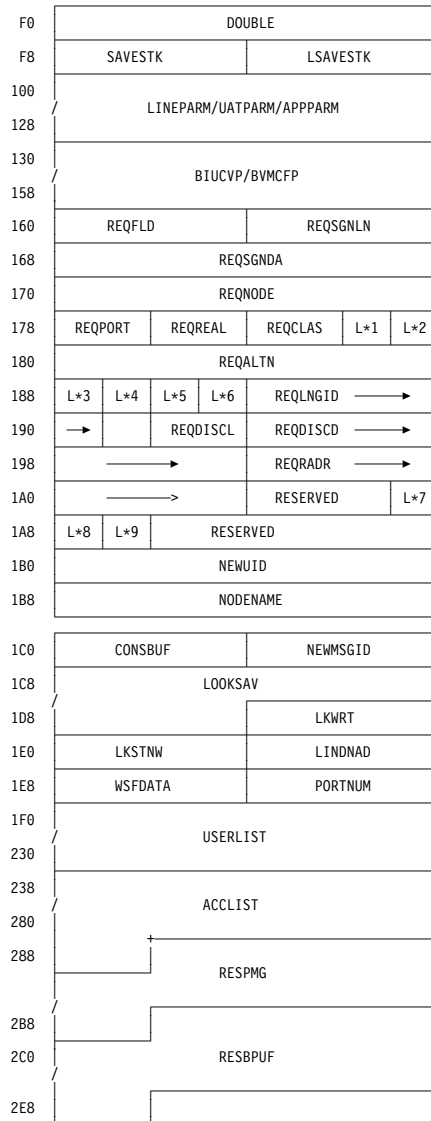


Figure 38. DVMNTCB Control Block Format

Field	Length	Use
DOUBLE	8	WORK AREA
SAVESTK	4	POINTER TO STACK AREA
LSAVESTK	4	LENGTH OF STACK IN DWs
LINEPARM/ UATPARM/ APPPARM	48	TASK ATTACH PARAMETER AREA
BIUCVP/BVMCFD	48	IUCV OR VMCF PARAMETER AREA.
REQFLD	4	CONTAINS WORD 'NETW'
REQSGNLN	4	SESSION DISC LENGTH

Field	Length	Use
REQSGNDA	8	SESSION DISC DATA
REQNODE	8	ANY SPECIFIC NODE
REQPORT	2	PORT NUMBER OR X'FF'
REQREAL	2	TERMINAL ADDRESS
REQCLAS	2	TERMINAL CLASS
L*1 (REQMDL)	1	TERMINAL MODEL NUMBER
L*2 (REQTYPE)	1	ATTRIBUTE FLAGS
	Values defined in REQTYPE:	
	REQEXTN	X'80'
	REQDBCS	X'01'
REQALTN	8	ALTERNATE ROUTING
L*3 (REQFLGS)	1	SPECIALTY FLAGS
	Values defined in REQFLGS:	
	REQIOV	X'80'
L*4 (REQMARG)	1	NLS COMPATIBILITY LEVEL
L*5 (REQLNNGM)	1	NLS LANGUAGE NUMBER
	Values defined in REQLNNGM:	
	REQLNGEX	X'01'
L*6 (REQNPKEY)	1	NOTEPAD KEY
REQLNGID	5	NLS LANGUAGE NAME
	1	RESERVED
REQDISCL	2	TEMPORARY DISCONNECT LENGTH
REQDISCD	8	TEMPORARY DISCONNECT STRING/KEY
REQRADR	8	EBCDIC TERMINAL ADDRESS
RESERVED	3	RESERVED
L*7 (SCAN)	1	SCAN NUMBER
L*8 (UMDL)	1	DISPLAY MODEL
L*9 (PMDL)	1	PORT MODEL
RESERVED	6	RESERVED
NEWUID	8	NEW USERID
NODENAME	8	SAVE AREA FOR NODE ID
CONSBUF	4	ADDRESS OF CONSOLE COMMAND BUFFER
NEWMSGID	4	MESSAGE I.D. FROM NEW REQUEST
LOOKSAVE	20	SAVE AREA FOR LOOKAUT ROUTINE
LKWRT	4	LINK SAVE AREA FOR WRTCON ROUTINE
LKSTNW	4	LINK SAVE AREA FOR START NEW ROUTINE
LINDNAD	4	SAVE FOR TCB TO GO TO FREE STORE ... WHEN RECYCLING LINE DRIVER
WSFDATA	4	ADDRESS OF WRITE STRUCTURED FIELD DATA
PORTNUM	4	SAVE AREA FOR PORT NUMBER
USERLIST	64	USER EXIT PARAMETER LIST
ACCLIST	82	STATISTICS EXIT PARAMETER LIST
RESPMSG	48	RESPONSE MESSAGE AREA
RESPBUF	48	RESPONSE MESSAGE AREA

**PSPI end**



## Appendix H. PVM Access Security Exits

### DMTI

This chapter describes the security exits sample code provided for the PVM component. These security exit routines will enable an installation to control sessions originating at a local node, outside the local node, and commands which are destined for or originated at another node. This will be accomplished by using a table called REMOTE AUTH which must be located on a minidisk that is accessed by PVM's user ID.

#### Notes:

1. The REMOTE AUTH file can also have a file name of the PVM local node id. The access security exits looks for file *nodeid* AUTH first and if not found, then REMOTE AUTH. This file will be referred to as REMOTE AUTH throughout this appendix.
2. These exits are included in the PVM 2.1.1 module supplied by IBM on the product tape and built by VMSES/E. If an installation does not require the use of these exits, then place a dummy REMOTE AUTH file on PVM's 191 minidisk or a disk the PVM user id has access to.

Local exit routines have been added to PVM to prevent users from accessing nodes outside the scope of their authorization. PVM maintains an in-storage copy of the REMOTE AUTH file. Entries in this file define the valid destinations for the specified user IDs. A technique known as **generics** allows security personnel to control vast numbers of IDs and terminals with a one-line record.

PVM provides entry points for locally defined exits from the DVMNUE module. This module has been updated to call a new module, DVMSEC, which can be used to restrict users from accessing nodes outside their authority. Routines have been designed and implemented in DVMSEC:

- For the DVMNUE3 exit to restrict local users
- For the DVMNUE4 exit to restrict remote users
- For the DVMNUE5 exit to restrict commands
- To initialize the security exits table in storage from the REMOTE AUTH file
- To reinitialize this table from an updated REMOTE AUTH file by using a command, without shutting PVM down.

For more information, see "Developing Exits" on page 114.

## The REMOTE AUTH File

The REMOTE AUTH file contains fixed-length, 80-character records with one record per line. Input must be in uppercase alphanumeric characters. At least one blank must exist between each field of a record. Comments may appear after the last parameter, and any entry not preceded with an opcode will be considered a comment. Opcodes must appear in column 1. Valid opcodes are:

**GATELINK**      Defines adjacent links to a node

<b>REM-CMD</b>	Restricts users from issuing commands at local or remote nodes if the target is not the origin node
<b>REM+USER</b>	Allows a user to continue past a remote node
<b>SNA-USER</b>	Restricts local users with SNA terminals
<b>VM-USER</b>	Restricts local users with non-SNA terminals.

In general, when the first match is found for a given set of conditions, the security exits allow the session to continue even if the next record disallows access. Therefore, the REMOTE AUTH table must be coded carefully. The exception to this rule is REM-CMD which will work just the opposite.

**Note:** User sessions and commands are not allowed if a REMOTE AUTH file is not found.

**Note:** All **unauthorized** attempts to access PVM are recorded in the SECAUDIT LOG file. All **valid** attempts are recorded by PVM in the PVM audit file if AUDIT is specified in the PVM CONFIG file. For examples of the SECAUDIT file see pages 331 and 334.

## Using Generics

To minimize the size of the REMOTE AUTH file (and the search time required at runtime), a technique known as **generics** can be used in each opcode parameter. Use an asterisk \* to indicate that any valid character may appear in the indicated position. For example, specifying NYC\*\*\* as a node name, the following are valid because they all will match a user's node: NYCVM1, NYCDA, NYCZ, and NYC.

However, node NYCPUBS would not match as a user's node because there are more than three characters after NYC.

For an example of records using generics, see page 326

## Reloading the REMOTE AUTH File

The REMOTE AUTH file can be reloaded without shutting PVM down. To do this, first update the REMOTE AUTH file. Then, the file can be reloaded into PVM by issuing the following command to PVM:

```
SEC RELOAD
```

This is a privileged command, authorization is required to execute it.

**Note:** The disk might need to be accessed again to read the updated file.



---

## Definitions of the REMOTE AUTH File Entries

The following section explains REMOTE AUTH records introduced with this support.

---

### GATELINK

▶▶—GATELINK—*linkid*————▶▶

#### Purpose

To apply restrictions to remote sessions and commands across adjacent PVM links, use the GATELINK record.

#### Operands

*linkid*  
is the node adjacent to the node where the GATELINK record resides.

#### Examples

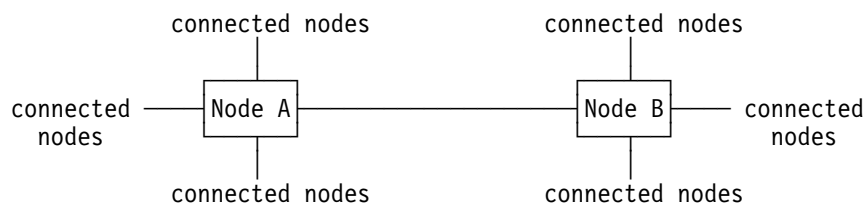


Figure 39. Example of Adjacent Nodes using the GATELINK Record

The REMOTE AUTH file at Node B contains a GATELINK NODEA record.

Node A is an adjacent node to Node B; they are connected by a PVM link. No commands will be allowed to or from Node A or nodes which use the link between Node A and Node B. All remote users in the network must have a REM+USER record to access any node which uses the link between Node A and Node B.

## REM-CMD

►►—REM-CMD—*userid*—*origin*—◄◄

### Purpose

Use REM-CMD to restrict users from issuing commands at a specified local or remote node. Use this record if an adjacent node has not been defined by the GATELINK record.

### Operands

*userid*

is the virtual machine identification of the local or remote user to restrict from issuing commands into the PVM network.

*origin*

are the nodes from which commands from *userid* are restricted.

**Note:** This exit is not applicable if the target node ID for executing the command is also the origin node ID.

### Examples

Below are two examples showing REM-CMD restrictions:

Command	Meaning
REM-CMD LES NYC***	<ul style="list-style-type: none"><li>• Restricts user LES from issuing commands at any node of the form NYC***.</li><li>• User LES may issue commands at any node other than of the form NYC***.</li><li>• All other users may issue commands at any node.</li></ul>
REM-CMD LES *****	<ul style="list-style-type: none"><li>• Restrict user LES from issuing commands at any node.</li><li>• All other users may issue commands at any node.</li></ul>

## REM+USER

►—REM+USER—*userid*—*origin*—*destination*—◄

### Purpose

Use the REM+USER record to identify users who are authorized to have sessions across a PVM link that has been defined with a GATELINK record. Unauthorized users will be denied access to the network.

### Operands

#### *userid*

are the remote users allowed to continue to the next adjacent node or to logon to this node.

The session identification is in the form of:

- GRAFxxx for PVM dialed users
- The user's user ID for PASSTHRU, MPVM and IUCV users
- *luname* if the session origin is an SNA application

#### *origin*

are the starting nodes available to remote users.

#### *destination*

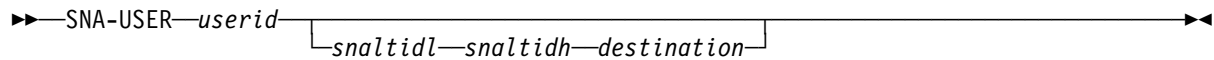
are the target nodes available to remote users.

### Examples

The following examples assume that a GATELINK has been defined.

Command	Meaning
REM+USER TOM LOS*** NYCVM1	<ul style="list-style-type: none"><li>• Permits user TOM from nodes of the form LOS*** access to node NYCVM1.</li><li>• User TOM may not access any nodes other than NYCVM1, nor access NYCVM1 from nodes other than of the form LOS***.</li><li>• All other users will be not be allowed access to any nodes because of the explicit GATELINK record.</li></ul>
REM+USER TOM ***** NYCVM1	<ul style="list-style-type: none"><li>• Permits user TOM from any node access to node NYCVM1.</li><li>• All other users will not be allowed access to any nodes because of the explicit GATELINK record.</li></ul>

## SNA-USER



### Purpose

Use the SNA-USER record to restrict local users with SNA terminals. Local users can be restricted to a specific terminal address range, and/or restricted to specific destination nodes

The exit code differentiates an SNA user from a non-SNA user through the users' real terminal address supplied to it by PVM. SNA is assumed if the address is greater than five characters, or the first character is not an *r* or *l*.

Disconnected MPVM or IUCV users can be restricted with the SNA-USER record with a terminal ID of DSC.

### Operands

*userid*

are the local users to be restricted.

The session identification is in the form of:

- GRAFxxx for PVM dialed users
- The user's user ID for PASSTHRU, MPVM, and IUCV users

*snaltidl*

is the SNA real terminal lower address boundary available to the local users being restricted. It is alphanumeric and up to eight characters long.

*snaltidh*

is the SNA real terminal higher address boundary available to the local users being restricted. It is alphanumeric and up to eight characters long.

*destination*

are the target nodes available to the local users being restricted.

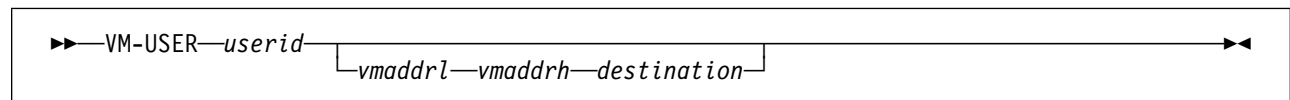
### Examples

These are examples of the restrictions that SNA-USER places on users:

Command	Meaning
SNA-USER JIM H436***** H446***** NYC*****	<ul style="list-style-type: none"><li>• Restricts local user ID JIM on an SNA terminal to address range H436AAAA-H4469999 and access to target nodes of the form NYC***** only.</li><li>• All other SNA terminal attached users may access any target node.</li></ul>

Command	Meaning
SNA-USER ***** H123**** H124**** NYCVM1	<ul style="list-style-type: none"> <li>Restricts all SNA terminal attached users to address range H123AAAA-H1249999 and access to target node NYCVM1 only.</li> <li>Any SNA terminal attached users outside address range H123AAAA-H1249999 will not be allowed access to any target nodes.</li> </ul>
SNA-USER MAINT DSC DSC NYCVM2	<ul style="list-style-type: none"> <li>Restricts MAINT user ID running a disconnected application accessed to target node NYCVM2 only.</li> <li>All other SNA terminal attached users may access any target node.</li> </ul>
SNA-USER VEN**** AAAAAAA 99999999 CHICAGO	<ul style="list-style-type: none"> <li>Restricts all SNA terminal attached user IDs of the form VEN**** access to target node CHICAGO only.</li> <li>All other SNA terminal attached users may access any target node as long as the VM user ID does not start with VEN.</li> </ul>
SNA-USER *****	<ul style="list-style-type: none"> <li>Denies all SNA terminal attached users access to any node on the network no matter what address they are logged on to.</li> </ul> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>This is the only time the parameters <i>snaltidl</i>, <i>snaltidh</i>, and <i>destination</i> are optional. They are optional because by omitting one or more of the parameters the example above is accomplished. All other times they are required.</li> <li>This can be used to disallow all SNA terminal attached users access to the SEC PVM virtual machine.</li> </ul>

## VM-USER



## Purpose

Use the VM-USER record to restrict local users with non-SNA terminals. Local users can be restricted to a specific terminal address range, and/or restricted to specific destination nodes.

## Operands

### *userid*

are the local users to be restricted.

The session identification is in the form of:

- GRAFxxx for PVM dialed users
- The user's user ID for PASSTHRU, MPVM, and IUCV users

### *vmaddrl*

is the non-SNA real terminal lower address boundary available to the local users being restricted in the format of *hhh* or *hhhh* for local devices, and *Lhhh* or *Lhhhh* for logical devices. The *h* is any hexadecimal character. The use of generics is not allowed with the *vmaddrl* parameter.

### *vmaddrh*

is the non-SNA real terminal higher address boundary available to the local users being restricted in the format of *hhh* or *hhhh* for local devices, and *Lhhh* or *Lhhhh* for logical devices. The *h* is any hexadecimal character. The use of generics is not allowed with the *vmaddrh* parameter.

### *destination*

are the target nodes available to the local users being restricted.

## Examples

These are examples of restrictions that VM-USER places on users:

Command	Meaning
VM-USER DAVID 100 1FF VNETCMC	<ul style="list-style-type: none"><li>• Restricts local user ID DAVID on a non-SNA terminal to address range 100-1FF and access to target node VNETCMC only.</li><li>• All other non-SNA terminal attached users may access any target node.</li></ul>
VM-USER ***** 100 1FF **VM1	<ul style="list-style-type: none"><li>• Restricts all non-SNA terminal attached users to address range 100-1FF and access to target nodes of the form **VM1 only.</li><li>• Any non-SNA terminal attached users outside address range 100-1FF will not be allowed access to any target nodes.</li></ul>
VM-USER GRAF*** 000 FFF NYC***	<ul style="list-style-type: none"><li>• Restricts all non-SNA terminal attached users having used the DIAL command access to target nodes of the form NYC*** only.</li><li>• All other local non-SNA terminal attached PVM users will be allowed access to any target node as long as the VM user ID does not start with GRAF.</li></ul>

Command	Meaning
VM-USER *****	<ul style="list-style-type: none"> <li>Denies all non-SNA terminal attached users access to any node on the network no matter what address they are logged on to.</li> </ul> <p><b>Note:</b></p> <ul style="list-style-type: none"> <li>This is the only time the parameters <i>vmaddr</i>, <i>vmaddrh</i>, and <i>destination</i> are optional. They are optional because by omitting one or more of the parameters the example above is accomplished. All other times they are required.</li> <li>This can be used to disallow all non-SNA terminal attached users access to the SEC PVM virtual machine.</li> </ul>

## Messages

### Messages Returned to the User:

The sample security exits return the following messages to the user:

---

*nodeid* **SEC01: NOT ALLOWED TO DESTINATION**

**Explanation:** The user is not authorized to access the desired destination.

---

*nodeid* **SEC04: NOT ALLOWED TO ISSUE COMMAND**

**Explanation:** Indicates that the user/origin combination is not authorized to issue PVM commands at this node.

---

*nodeid* **SEC06: NO REMOTE AUTH TABLE FOUND**

**Explanation:** Indicates that a REMOTE AUTH table could not be found on any disk accessed by PVM's user ID. Thus, no user sessions or commands will be allowed.

## Usage Notes

1. The node ID at the beginning of the messages above is the node ID of the PVM machine that issued that message.
2. No syntax error messages are produced for invalid records in the REMOTE AUTH file.

### Messages Returned to the PVM Console

The sample security exits return the following messages to the PVM console:

---

**INVALID SUBCOMMAND ON SEC COMMAND**

**Explanation:** The SEC command had a subcommand other than RELOAD.

---

**COULD NOT LOCATE REMOTE AUTH FILE**

**Explanation:** The REMOTE AUTH file is not on any disks accessed by the PVM user ID.

---

**COULD NOT ALLOCATE STORAGE FOR REMOTE AUTH TABLES**

**Explanation:** Not enough storage is available for loading the in-storage tables of the REMOTE AUTH file.

---

**SECAUDIT LOG FILE CANNOT BE WRITTEN**

**Explanation:** An error occurred while attempting to write an entry to the SECAUDIT LOG file.

## Usage Notes

1. Messages are sent to the PVM user ID using the CMS macro LINEWRT.
2. Use of LINEWRT (and other CMS functions) should be limited to avoid possible conflicts between CMS and the PVM operating environment.

---

## Format of SECAUDIT LOG Records

Invalid session or command execution attempts are recorded in the SECAUDIT LOG file on the SEC PVM's filemode A minidisk. SECAUDIT is a fixed-length file created and updated by DVMSEC. SECAUDIT LOG contains these types of records which are created for the following situations, unsuccessful attempts:

- By a user to gain access to the PVM network through a local system (DVMNUE3 entry point).
- To connect to or through a remote PVM node (DVMNUE4 entry point).
- To issue commands to or through a remote PVM node or from a user ID which has been excluded from issuing commands in the PVM network (DVMNUE5 entry point).

Review the SECAUDIT LOG file periodically because it will continue to grow as unauthorized attempts are added to it.

Unauthorized attempts are all recorded in the SECAUDIT LOG file. A 3, 4, or 5 is written to the SECAUDIT LOG after the DATE/TIME indicating the type of attempt, where the attempt originated, and the destination node attempted in the case of sessions. In the case of commands, where the command originated and what command execution was attempted appears on the lines following the 5.



Table 21. SECAUDIT Record Formats

Record Format Type	Column	Contents
3 and 4	1	Date
	10	Time
	19	SEC record format type 3 or 4
	21	Origin node ID
	30	Origin user ID
	39	Terminal address
	48	Destination node ID (attempted)
<b>Note:</b> The terminal address will be blank for unauthorized access attempts by an IUCV user at a remote node. The terminal address will be DSC for unauthorized access attempts by an MPVM or IUCV user at the local node.		
5	1	Date
	10	Time
	19	SEC record format type 5
	21	Origin node ID of user issuing command
	30	Origin user ID of user issuing command
	39	Node ID where command was to be executed
	48	Command to be executed (up to column 80)

See pages 331 and 334 for more examples of a SECAUDIT LOG file.

## Controlling Local Users Using the REMOTE AUTH Table

The following REMOTE AUTH table controls the access to PVM for local users on the CAMBRIDG system.

```

VM-USER GATEDOC 0404 040F PARISVM
VM-USER GUEST* 0410 0410 NYCVMX
VM-USER GUEST** 0400 0410 NYCVMT
VM-USER GUEST4 0000 1FFF CAMBVM2
VM-USER XXXPAR* 0000 1FFF
REM-CMD XXXPAR* CAMBRIDG
REM-CMD GUEST* CAMBRIDG

```

Figure 40. REMOTE AUTH Table for Local Users

User ID	Access
GATEDOC	Can only access node PARISVM from a terminal with an address between 0404 and 040F.
GUEST* GUEST**	<p>Represents user IDs GUESTA through GUESTZ, and user IDs GUEST0 through GUEST9. The two asterisks ** of user ID GUEST** can be any combination of A-Z and 0-9.</p> <p>User ID GUEST* and GUEST** can access NYCVMT from a terminal on addresses 400 through 410. User ID GUEST* can also access NYCVMX from a terminal on address 410.</p> <p><b>Note:</b> The above list represents all the nodes that GUEST** and GUEST* can access and from which address.</p>

User ID	Access
GUEST4	<ul style="list-style-type: none"> <li>Can access node CAMBVM2 from any address between 000 and 1FFF.</li> <li>Can access NYCVMT from a terminal on address between 400 through 410.</li> <li>Can access NYCVMX from a terminal on address 410.</li> </ul> <p><b>Note:</b> The above list represents all the nodes that GUEST4 can access and from which address.</p>
XXXXPAR*	<p>Represents user IDs XXXPARA through XXXPARZ, and user IDs XXXPAR0 through XXXPAR9. None of these user IDs can access PVM because no destination node ID is specified. This indicates that this user ID is not to have access to PVM no matter what address is specified on the REMOTE AUTH record. Since no destination was specified, the terminal address range has no effect because there will always be a mismatch on the destination.</p> <p><b>Note:</b> User IDs XXXPAR* and GUEST* Are unable to execute PVM commands on remote nodes.</p>

## DVMNUE3 Testing

The following are test cases using the REMOTE AUTH table controlling local users.

Table 22 (Page 1 of 2). REMOTE AUTH Table Controlling Local Users

User ID	Address	PASSTHRU Node or Command to PVM	Access
GATEDOC	40F	CAMBVM2 NYCVMX NYCVMT PARISVM SM PVM CMD PARIS Q SYS	Disallowed Disallowed Disallowed Allowed Allowed
	410	CAMBVM2 NYCVMX NYCVMT PARISVM	Disallowed Disallowed Disallowed Disallowed
	82A	CAMBVM2 NYCVMX NYCVMT PARISVM	Disallowed Disallowed Disallowed Disallowed
GUEST3	410	NYCVMX NYCVMT CAMBVM2 PARISVM SM PVM CMD LOSVMV Q SYS	Allowed Allowed Disallowed Disallowed Disallowed
	82A	CAMBVM2 NYCVMX NYCVMT PARISVM SM PVM CMD LONVM0 Q SYS	Disallowed Disallowed Disallowed Disallowed Disallowed

Table 22 (Page 2 of 2). REMOTE AUTH Table Controlling Local Users

User ID	Address	PASSTHRU Node or Command to PVM	Access
GUEST4	40F	CAMBVM2 CAMBVM3 NYCVMT NYCVMX SM PVM CMD NYCVMV Q SYS	Allowed Disallowed Allowed Disallowed Disallowed
	410	NYCVMX NYCVMT NYCGATE	Allowed Allowed Disallowed
	82A	CAMBVM2 NYCVMT NYCVMX	Allowed Disallowed Disallowed
XXPAR3	410	PARISVM NYCGATE SM PVM CMD PARVMV Q SYS	Disallowed Disallowed Disallowed

## SECAUDIT LOG

The following is the SECAUDIT LOG associated with the REMOTE AUTH table controlling local users.

```

09/21/91 06:35:24 3 CAMBRIDG GUEST4 040F CAMBVM3
09/21/91 06:35:45 3 CAMBRIDG GUEST4 040F NYCVMX
09/21/91 06:36:45 5 CAMBRIDG GUEST4 NYCVMV CMD NYCVMV Q SYS
09/21/91 06:36:55 3 CAMBRIDG GUEST4 0410 NYCGATE
09/21/91 06:37:44 3 CAMBRIDG GUEST3 0410 CAMBVM2
09/21/91 06:37:50 3 CAMBRIDG GUEST3 0410 PARISVM
09/21/91 06:38:13 5 CAMBRIDG GUEST3 LOSVMV CMD LOSVMV Q SYS
09/21/91 06:38:41 3 CAMBRIDG XXPAR3 0410 PARISVM
09/21/91 06:38:47 3 CAMBRIDG XXPAR3 0410 NYCGATE
09/21/91 06:38:51 5 CAMBRIDG XXPAR3 PARVMV CMD PARVMV Q SYS
09/21/91 06:40:30 3 CAMBRIDG GUEST4 082A NYCVMT
09/21/91 06:40:34 3 CAMBRIDG GUEST4 082A NYCVMX
09/21/91 06:40:52 3 CAMBRIDG GUEST3 082A CAMBVM2
09/21/91 06:40:58 3 CAMBRIDG GUEST3 082A NYCVMX
09/21/91 06:41:02 3 CAMBRIDG GUEST3 082A NYCVMT
09/21/91 06:41:06 3 CAMBRIDG GUEST3 082A PARISVM
09/21/91 06:41:15 5 CAMBRIDG GUEST3 LONVM0 CMD LONVM0 Q SYS
09/21/91 06:41:32 3 CAMBRIDG GATEDOC 040F CAMBVM2
09/21/91 06:41:39 3 CAMBRIDG GATEDOC 040F NYCVMX
09/21/91 06:41:41 3 CAMBRIDG GATEDOC 040F NYCVMT
09/21/91 06:41:45 3 CAMBRIDG GATEDOC 0410 CAMBVM2
09/21/91 06:42:52 3 CAMBRIDG GATEDOC 0410 NYCVMX
09/21/91 06:42:56 3 CAMBRIDG GATEDOC 0410 NYCVMT
09/21/91 06:42:02 3 CAMBRIDG GATEDOC 0410 PARISVM
09/21/91 06:42:25 3 CAMBRIDG GATEDOC 082A CAMBVM2
09/21/91 06:42:32 3 CAMBRIDG GATEDOC 082A NYCVMX
09/21/91 06:42:39 3 CAMBRIDG GATEDOC 082A NYCVMT
09/21/91 06:43:19 3 CAMBRIDG GATEDOC 082A PARISVM

```

Figure 41. SECAUDIT LOG for Local Users

---

## Controlling Remote Users by Using the REMOTE AUTH Table

The following REMOTE AUTH table at a remote gateway node controls access between systems which use PVM links defined at this node. This machine is not the PVM machine that users normally access. Traffic through this node is destined to or from another location. Users are not allowed to access PVM locally. Commands cannot be sent on a link which goes through this gateway node.

```
VM-USER  *****
REM+USER DRBIM    BOSTON  CAMBRIDG
REM+USER DRBIM    BOSTON  NEWYORK
REM+USER DAVID*   CAMBRIDG BOSTON
REM+USER DAVID*   CAMBRIDG LONDON
REM+USER BILL     CAMBRIDG BOSTON
REM+USER JARRELL  CAMBRIDG LONDON
REM+USER GATEDOC  CAMBRIDG *****
REM+USER NODEADMN CAMBRIDG *****
REM-CMD  ***** GATEWAY
GATELINK BOSTON
GATELINK CAMBRIDG
```

Figure 42. REMOTE AUTH Table at a Remote Gateway Node

Figure 43 shows how the PVM network is connected:

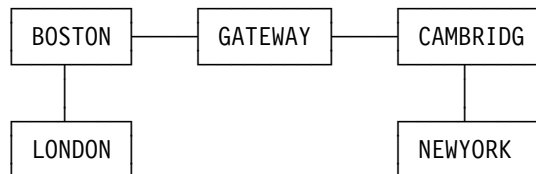


Figure 43. The PVM Network

The REMOTE AUTH table shown in Figure 42 disallows access from any local terminal to PVM by using the first VM-USER statement. Since no destination node ID is specified, no access is allowed by any local user. Also, no commands can be issued into the PVM network by any local user ID by using the REM-CMD statement.

Two gateway links are defined: CAMBRIDG and BOSTON. Any traffic routed between these two links is checked against the REMOTE AUTH table. Also, any commands routed between these two links cannot be executed.

User DRBIM on node BOSTON can only access the CAMBRIDG and NEWYORK nodes. No other user from BOSTON or LONDON can access the CAMBRIDG, NEWYORK, or GATEWAY nodes.

The following CAMBRIDG user IDs have access to node BOSTON:

- DAVID0 thru DAVID9
- DAVIDA thru DAVIDZ
- BILL
- GATEDOC
- NODEADMN.

The following CAMBRIDG user IDs are given access to node LONDON:

- DAVID0 thru DAVID9
- DAVIDA thru DAVIDZ
- GATEDOC
- JARRELL
- NODEADMN.

User ID NODEADMN can also access the GATEWAY node.

No other user from CAMBRIDG or NEWYORK can access the BOSTON, LONDON, or GATEWAY nodes.

## DVMNUE4 Testing

The following are test cases using the REMOTE AUTH table controlling remote users.

Table 23. REMOTE AUTH Table Controlling Remote Users

From Node	From User ID	PASSTHRU Node	Access
<b>CAMBRIDG</b>	DAVID2A	BOSTON LONDON GATEWAY NEWYORK	Disallowed Disallowed Disallowed Allowed
<b>CAMBRIDG</b>	DAVID2 or GATEDOC	BOSTON LONDON GATEWAY NEWYORK	Allowed Allowed Disallowed Allowed
<b>CAMBRIDG</b>	BILL	BOSTON LONDON GATEWAY NEWYORK	Allowed Disallowed Disallowed Allowed
<b>CAMBRIDG</b>	JARRELL	BOSTON LONDON GATEWAY NEWYORK	Disallowed Allowed Disallowed Allowed
<b>CAMBRIDG</b>	NODEADM	BOSTON LONDON GATEWAY NEWYORK	Allowed Allowed Allowed Allowed
<b>NEWYORK</b>	OTHER	BOSTON LONDON GATEWAY CAMBRIDG	Disallowed Disallowed Disallowed Allowed
<b>BOSTON</b>	DRBIM	CAMBRIDG NEWYORK GATEWAY LONDON	Allowed Allowed Disallowed Allowed
<b>LONDON</b>	OTHER	CAMBRIDG NEWYORK GATEWAY BOSTON	Disallowed Disallowed Disallowed Allowed

## SECAUDIT LOG

The following is the SECAUDIT LOG associated with the REMOTE AUTH table controlling remote users.

```
09/21/91 09:39:49 4 CAMBRIDG DAVID2A 082A BOSTON
09/21/91 09:40:51 4 CAMBRIDG DAVID2A 082A LONDON
09/21/91 09:41:22 4 CAMBRIDG DAVID2A 082A GATEWAY
09/21/91 09:41:30 4 CAMBRIDG GATEDOC 082A GATEWAY
09/21/91 09:41:35 4 CAMBRIDG BILL 082A LONDON
09/21/91 09:41:39 4 CAMBRIDG BILL 082A GATEWAY
09/21/91 09:41:35 4 CAMBRIDG JARRELL 082A BOSTON
09/21/91 09:41:39 4 CAMBRIDG JARRELL 082A GATEWAY
09/21/91 09:43:45 4 NEWYORK OTHER 082A BOSTON
09/21/91 09:43:51 4 NEWYORK OTHER 082A LONDON
09/21/91 09:43:56 4 NEWYORK OTHER 082A GATEWAY
09/21/91 09:44:00 4 BOSTON DRBIM 082A GATEWAY
09/21/91 09:44:14 4 LONDON OTHER 082A CAMBRIDG
09/21/91 09:46:58 4 LONDON OTHER 082A NEWYORK
09/21/91 09:47:16 4 LONDON OTHER 082A GATEWAY
```

*Figure 44. SECAUDIT LOG for Remote Users*

**DMTI end**

## Appendix I. CVIEW Product Panels

CVIEW displays menu and data-entry panels to its users during setup mode. In addition, consultants and moderators receive control-mode panels during CVIEW sessions by pressing the PA1 key. If your establishment uses the NOTIFY panel, CVIEW displays it to you at preset intervals during session mode. This appendix presents all the CVIEW product panels and indicates when they are displayed.

### Setup-mode Panels

Setup mode begins after you enter 'DIAL CVIEW'. The first panel displayed is the **HOME** panel:

```

HOME                COOPERATIVE VIEWING FACILITY (CVIEW)
-----
Enter an item number.

    Consult Session

        1  Receive assistance
        2  Give assistance

    Conference Session

        3  Attend a conference
        4  Moderate a conference

Press ENTER after you enter an item number.
-----
PF:  1= Help      2=          3= Quit    4=          5=          6=
PF:  7=          8=          9=          10=         11=         12=

===>

```

Enter a number (1-4) to select your role in the CVIEW session. After you select your role, you see one of the following panels:

1. A **recipient** sees the **RECEIVE** panel.
2. A **consultant** sees the **GIVE** panel.
3. An **attendee** sees the **ATTEND** panel.
4. A **moderator** sees the **MODERATE** panel.

## Consult Session Setup (Recipient)

```

RECEIVE                                CVIEW - CONSULTANT SELECTION
-----
Contact a consultant by telephone before entering an item number.

1  Consultant not listed
2  Consultant #1          Product 1          555-0001
3  Consultant #2          Product 2          555-0002
4  Consultant #3          Product 3          555-0003
5  Consultant #4          Product 4          555-0004
6  Consultant #5          Product 5          555-0005
7  Consultant #6          Product 6          555-0006
8  Consultant #7          Product 7          555-0007
9  Consultant #8          Product 8          555-0008
10 Consultant #9          Product 9          555-0009
11 Consultant #10         Product 10         555-0010
12 Consultant #11         Product 11         555-0011

Press ENTER after you enter an item number.
-----
PF: 1= Help    2=          3= Quit    4= Home    5=          6=
PF: 7=          8=          9=          10=         11=         12=

===>

```

The **RECEIVE** panel appears after the HOME panel when you are setting up a consult session as a recipient. Enter a number from 1 - 12 to select a consultant.

```

IDRECV                                CVIEW - UNLISTED CONSULTANT SELECTION
-----
Enter the identification provided by the consultant.

Consultant's identification: ===>          1 to 8 characters (no blanks)

The consultant identification you enter must also be entered by the
consultant. This identification allows CVIEW to associate your
terminal with the consultant's terminal in a consult session.

Press ENTER after you enter the identification.
-----
PF: 1= Help    2=          3= Quit    4= Home    5=          6=
PF: 7=          8=          9=          10=         11=         12=

===>

```

CVIEW displays the **IDRECV** (ID RECEIVE) panel if you entered 1 to select a consultant who is not already listed with CVIEW. Now, you must enter the consultant's identification as he gives it to you.



```

RECVWAIT                                CVIEW - RECIPIENT WAIT
-----
You and the consultant should be in contact by telephone.  You are waiting
for the consultant to establish a consult session with you.

You selected to receive assistance from consultant #XX: CONSULTANTID.
The phone number for this consultant is XXXXXXXXXXXXXXXX.

The consultant must do the following to establish a consult session with you:

1) Start from the VM system logo (where CVIEW is running)
2) Clear the screen and issue the command: DIAL CVIEW
3) Select option 2 from the Home panel
4) Specify the same consultant identification you selected (see above)

5) The VM system logo will be displayed when the consult session starts

If you do not wish to wait for the consultant, press the Home PF key to
return to the Home panel.

No further action is required from you.
-----
PF:  1= Help      2=          3= Quit    4= Home    5=          6=
PF:  7=          8=          9=          10=         11=         12=

==>

```

CVIEW displays the **RECVWAIT** (RECEIVE WAIT) panel when you have responded to all the recipient setup panels. You are waiting for the consultant to complete his setup panels. The logo for your VM/ESA system will appear when the consultant completes his setup panels.

## Consult Session Setup (Consultant)

```

GIVE                                CVIEW - CONSULTANT IDENTIFICATION
-----
Enter an item number.  The recipient must enter the same item number.

1  Consultant not listed
2  Consultant #1          Product 1          555-0001
3  Consultant #2          Product 2          555-0002
4  Consultant #3          Product 3          555-0003
5  Consultant #4          Product 4          555-0004
6  Consultant #5          Product 5          555-0005
7  Consultant #6          Product 6          555-0006
8  Consultant #7          Product 7          555-0007
9  Consultant #8          Product 8          555-0008
10 Consultant #9          Product 9          555-0009
11 Consultant #10         Product 10         555-0010
12 Consultant #11         Product 11         555-0011

Press ENTER after you enter an item number.
-----
PF: 1= Help    2=          3= Quit    4= Home    5=          6=
PF: 7=          8=          9=          10=         11=         12=

===>

```

The **GIVE** panel appears after the HOME panel when you are setting up a consult session as a consultant. Enter a number from 1 - 12 to select your consultant identification.

```

IDGIVE                             CVIEW - UNLISTED CONSULTANT IDENTIFICATION
-----
Enter the data.

Consultant identification: ===>          1 to 8 characters (no blanks)

Display input PF key:    ===>          PF1,PF2....,PF24 or NONE
(for use during consult session)

The consultant identification you specify must also be entered by the
recipient.  This identification allows CVIEW to associate your
terminal with the recipient's terminal in a consult session.

You may press the PA1 key during the consult session to display the
Consult Session Control panel.

Press ENTER after you enter the data.
-----
PF: 1= Help    2=          3= Quit    4= Home    5=          6=
PF: 7=          8=          9=          10=         11=         12=

===>

```

CVIEW displays the **IDGIVE** (ID GIVE) panel if you entered 1 to indicate that you are an unlisted consultant. You must enter your identification and tell it to the recipient. You can use this panel to define a PF key to display input before it is processed.

CONSPASS	CVIEW - CONSULTANT PASSWORD
<hr style="border-top: 1px dashed black;"/>	
Enter the data.	
Consultant password:	===>                      1 to 8 characters (no blanks)
Display input PF key:	===>                      PF1,PF2...,PF24 or NONE
(for use during consult session)	
You may press the PA1 key during the consult session to display the Consult Session Control panel.	
Press ENTER after you enter the data.	
<hr style="border-top: 1px dashed black;"/>	
PF: 1= Help	2=
PF: 7=	8=
3= Quit	9=
4= Home	10=
5=	11=
6=	12=
===>	

CVIEW displays the **CONSPASS** (CONSULTANT PASSWORD) panel if you are a listed consultant with a password. Enter your password. You can also use this panel to define a PF key to display input before it is processed.

REDISP	CVIEW - DISPLAY INPUT
<hr style="border-top: 1px dashed black;"/>	
Enter the data.	
Display Input PF key:	===>                      PF1,PF2...,PF24 or NONE
(for use during consult session)	
You may press the PA1 key during the consult session to display the Consult Session Control panel.	
Press ENTER after you enter the data.	
<hr style="border-top: 1px dashed black;"/>	
PF: 1= Help	2=
PF: 7=	8=
3= Quit	9=
4= Home	10=
5=	11=
6=	12=
===>	

CVIEW displays the **REDISP** (RE-DISPLAY) panel if you are a listed consultant without a password. You can use this panel to define a PF key to display input before it is processed.

GIVEWAIT	CVIEW - CONSULTANT WAIT												
<p>You and the recipient should be in contact by telephone. You are waiting for the recipient to establish a consult session with you.</p> <p>You specified your consultant identification as #XX: CONSULTANTID.</p> <p>The recipient must do the following to establish a consult session with you.</p> <ol style="list-style-type: none"> <li>1) Start from the VM system logo (where CVIEW is running)</li> <li>2) Clear the screen and issue the command: DIAL CVIEW</li> <li>3) Select option 1 from the Home panel</li> <li>4) Select the same consultant identification you specified (see above)</li> <li>5) The VM system logo will be displayed when the consult session starts</li> </ol> <p>If you do not wish to wait for the recipient, press the Home PF key to return to the Home panel.</p> <p>No further action is required from you.</p>													
<table border="0" style="width: 100%;"> <tr> <td style="width: 16.6%;">PF: 1= Help</td> <td style="width: 16.6%;">2=</td> <td style="width: 16.6%;">3= Quit</td> <td style="width: 16.6%;">4= Home</td> <td style="width: 16.6%;">5=</td> <td style="width: 16.6%;">6=</td> </tr> <tr> <td>PF: 7=</td> <td>8=</td> <td>9=</td> <td>10=</td> <td>11=</td> <td>12=</td> </tr> </table> <p style="text-align: center; margin-top: 10px;">==&gt;</p>		PF: 1= Help	2=	3= Quit	4= Home	5=	6=	PF: 7=	8=	9=	10=	11=	12=
PF: 1= Help	2=	3= Quit	4= Home	5=	6=								
PF: 7=	8=	9=	10=	11=	12=								

CVIEW displays the **GIVEWAIT** panel when you have responded to all the consultant setup panels. You are waiting for the recipient to complete his setup panels. The logo for your VM/ESA system will appear when the recipient completes his setup panels.

## Conference Session Setup (Attendee)

ATTEND	CVIEW - CONFERENCE PARTICIPANT IDENTIFICATION												
<p>Enter the data.</p> <p>Conference name:      ==&gt;                      1 to 8 characters (no blanks)</p> <p>Your Name:              ==&gt;                      1 to 16 characters</p> <p>If you wish to leave the conference after it has started, you may press the PA1 key. If the moderator has assigned you to be a presenter in the conference, the PA1 key will perform the function normally associated with it.</p> <p>Press ENTER after you enter the data.</p>													
<table border="0" style="width: 100%;"> <tr> <td style="width: 16.6%;">PF: 1= Help</td> <td style="width: 16.6%;">2=</td> <td style="width: 16.6%;">3= Quit</td> <td style="width: 16.6%;">4= Home</td> <td style="width: 16.6%;">5=</td> <td style="width: 16.6%;">6=</td> </tr> <tr> <td>PF: 7=</td> <td>8=</td> <td>9=</td> <td>10=</td> <td>11=</td> <td>12=</td> </tr> </table> <p style="text-align: center; margin-top: 10px;">==&gt;</p>		PF: 1= Help	2=	3= Quit	4= Home	5=	6=	PF: 7=	8=	9=	10=	11=	12=
PF: 1= Help	2=	3= Quit	4= Home	5=	6=								
PF: 7=	8=	9=	10=	11=	12=								

The **ATTEND** panel appears after the HOME panel when you are setting up a conference session as an attendee. Enter the conference name provided by the moderator, and enter your own name.

PASSWORD	CVIEW - CONFERENCE PASSWORD
-----	
Enter the password for the conference.	
Conference password: ===>	1 to 8 characters (no blanks)
Press ENTER after you enter the password.	
-----	
PF: 1= Help	2=
PF: 7=	8=
3= Quit	4= Home
9=	10=
5=	11=
6=	12=
===>	

CVIEW displays the **PASSWORD** panel if the moderator assigns a password to the conference. Enter the conference password now.

CONFWAIT	CVIEW - CONFERENCE WAIT
-----	
A VM system logo will be displayed when the conference starts.	
You are waiting for conference XXXXXXXX to start.	
If you wish to leave the conference after it has started, you may press the PA1 key. If the moderator has assigned you to be a presenter in the conference, the PA1 key will perform the function normally associated with it.	
A Notify panel may be displayed occasionally by CVIEW. In order for the conference to continue, you must press ENTER when the Notify panel is displayed.	
If a conference password is required, you will be prompted for the password on the Conference Password panel.	
No further action is required from you.	
-----	
PF: 1= Help	2=
PF: 7=	8=
3= Quit	4= Home
9=	10=
5=	11=
6=	12=
===>	

CVIEW displays the **CONFWAIT** (CONFERENCE WAIT) panel after you complete all the setup panels to attend a conference. You are waiting for the moderator to start the conference. The logo for your VM/ESA system will appear when the moderator starts the conference.

## Conference Session Setup (Moderator)

MODERATE		CVIEW - MODERATOR IDENTIFICATION	
-----			
Enter the data.			
Conference Name:	====>	1 to 8 characters (no blanks)	
Your Name:	====>	1 to 16 characters	
Display Input PF key:	====>	PF1,PF2,...,PF24 or NONE	
(for use during conference)			
Conference Password:	====>	1 to 8 characters (no blanks)	
(optional)			
You may press the PA1 key during the conference to display the Conference Control panel.			
Press ENTER after you enter the data.			
-----			
PF: 1= Help	2=	3= Quit	4= Home
PF: 7=	8=	9=	10=
			11=
			12=
====>			

Enter the conference name on this panel and tell it to the attendees. Enter your name also. You can use this panel to define a PF key to display input before it is processed. In addition, you can assign a password to the conference. If you do so, tell the password to the attendees.

XXXXXXX		CVIEW - CONFERENCE PARTICIPANTS	
-----			
You may change a participant's status by entering one of the following item numbers next to the participant's name and pressing ENTER:			
1 Allow a participant to present in the conference (presenter)			
2 Allow a participant to only observe the conference (attendee)			
3 Drop a participant from the conference			
Participant	Status	Participant	Status
XXXXXXXXXXXXXXXX	Moderator		
====>		====>	
====>		====>	
====>		====>	
====>		====>	
====>		====>	
Press ENTER if you enter an item number next to a participant's name.			
Press the SESSION PF key to start the conference session.			
-----			
PF: 1= Help	2=	3= Quit	4= Home
PF: 7=	8=	9=	10=
			11=
			12= Session
====>			

The name you entered for the conference appears in the upper left-hand corner of this panel. You can now assign presenter status to any of the attendees or drop attendees from the conference. When you are ready to start the conference session, press the PF key for SESSION. After you press the SESSION PF key,

the logo for your VM/ESA system will appear, indicating that the session has begun.

## Notify-mode Panels

Notify mode occurs during a consult or conference session if the use of the NOTIFY panel was selected for your establishment. The session cannot continue until every CVIEW user who receives a NOTIFY panel has responded to it.

NOTIFY	CVIEW - NOTIFY					
-----						
This Notify panel is being displayed by the Cooperative Viewing Facility (CVIEW) to make you aware you are running under the control of CVIEW.						
You are currently in a CVIEW session, someone else is watching the same screen images you are displaying.						
If you press the QUIT or HOME PF key while this panel is displayed, the session will be terminated.						
If you were not aware you were using this facility, contact your Computer Center Management.						
Every participant who receives a Notify panel must press ENTER for the session to continue						
Press ENTER to continue.						
-----						
PF: 1= Help	2=	3= Quit	4= Home	5=	6=	
PF: 7=	8=	9=	10=	11=	12=	
===>						

CVIEW displays the **NOTIFY** panel at preset intervals during consult and conference sessions. Press ENTER to continue the CVIEW session. If you do not want to continue the session, press the PF key assigned to either HOME or QUIT.

NTFYHOLD	CVIEW - NOTIFY HOLD				
-----					
You are waiting for other participants who have received the Notify panel to press ENTER.					
Every participant who receives the Notify panel must press ENTER for the session to continue.					
No further action is required from you.					
-----					
PF: 1= Help	2=	3= Quit	4= Home	5=	6=
PF: 7=	8=	9=	10=	11=	12=
===>					

CVIEW displays the **NTFYHOLD** (NOTIFY HOLD) panel after you press ENTER in response to the NOTIFY panel. The NTFYHOLD panel indicates that other session participants have not yet pressed ENTER in response to their NOTIFY panels.



## Control-mode Panels

A consultant or moderator enters control mode during a CVIEW session by pressing the PA1 key to access a control panel. During control mode, the consult or conference session is suspended.

### Consultant Control

```

CONSCTL                      CVIEW - CONSULT SESSION CONTROL
-----
You may change the data below by entering a new value.

Currently, your Display Input PF key is set to XXXX.

      Display Input PF key:      ==>          PF1,PF2,...,PF24 or NONE
      (for use during consult session)

If you had intended the PA1 key to perform the function normally
associated with it, press the PA1 key again now.

The consult session cannot continue while this panel is being displayed.


Press ENTER to continue.
-----
PF:  1= Help      2=          3= Quit    4= Home    5=          6=
PF:  7=          8=          9=          10=         11=         12=

      ==>

```

After you, as a consultant, press the PA1 key, CVIEW displays the **CONSCTL** (CONSULT SESSION CONTROL) panel. You can now change the PF key assignment for input display. In addition, you can press PA1 again while this panel is on the screen and the host or application program will perform the function normally associated with the PA1 key. The consult session continues when you press ENTER or PA1 in response to this panel.

## Moderator Control

```

CONFCTL                      CVIEW - CONFERENCE CONTROL
-----
You may change the data below by entering a new value.

Currently, your Display Input PF key is set to XXXX.

    Display Input PF key:  ===>          PF1,PF2,...,PF24 or NONE
    (for use during conference)

    Display the Conference
    Participants panel:    ===>          Yes or No

If you had intended the PA1 key to perform the function normally
associated with it, press the PA1 key again now.

The conference cannot continue while this panel is being displayed.

Press ENTER to continue.
-----
PF:  1= Help      2=          3= Quit    4= Home    5=          6=
PF:  7=          8=          9=          10=         11=         12=

    ===>

```

After you, as a moderator, press the PA1 key, CVIEW displays the **CONFCTL** (CONFERENCE CONTROL) panel. You can now change the PF key assignment for input display or request the CONFERENCE PARTICIPANTS panel. In addition, you can press PA1 again and the host or application program will perform the function normally associated with the PA1 key.

```

XXXXXXXXX                    CVIEW - CONFERENCE PARTICIPANTS
-----
You may change a participant's status by entering one of the following
item numbers next to the participant's name and pressing ENTER:

    1 Allow a participant to present in the conference (presenter)
    2 Allow a participant to only observe the conference (attendee)
    3 Drop a participant from the conference

    Participant      Status      Participant      Status
    XXXXXXXXXXXXXXXX Moderator    ===>
    ===>
    ===>
    ===>
    ===>
    ===>
    ===>

Press ENTER if you enter an item number next to a participant's name.
Press the SESSION PF key to return to the conference session.
-----
PF:  1= Help      2=          3= Quit    4= Home    5=          6=
PF:  7=          8=          9=          10=         11=         12= Session

    ===>

```

CVIEW displays the **CONFERENCE PARTICIPANTS** panel when you request it from the CONFERENCE CONTROL panel. The conference session can resume while this panel is displayed, if a presenter exists. From this panel, you can

change a participant's status to attendee or presenter, and drop attendees from the conference session. Type the appropriate number by a participant's name and press ENTER. You return to the conference in progress by pressing the SESSION PF key.



---

## Notices

This information was developed for products and services offered in the US. This material might be available from IBM (TM) in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it. p. 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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US

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

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

IBM may use or distribute any of the information you provide 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 Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

#### **COPYRIGHT LICENSE:**

This information may contain sample application programs in source language, which illustrate 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

---

## **Programming Interface Information**

This information primarily documents intended Programming Interfaces that allow the customer to write programs to obtain services of z/VM.

This information also documents information that is NOT intended to be used as Programming Interfaces of z/VM. This information is identified where it occurs in the following way:

**NOT-PI**

Non-programming interface information...

**NOT-PI end**

---

## **Trademarks**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at IBM copyright and trademark information - United States ([www.ibm.com/legal/us/en/copytrade.shtml](http://www.ibm.com/legal/us/en/copytrade.shtml)).

---

## **Terms and Conditions for Product Documentation**

Permissions for the use of these publications are granted subject to the following terms and conditions.

## Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

## Personal Use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

## Commercial Use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

## Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

---

## IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM Online Privacy Statement Highlights at <http://www.ibm.com/privacy> and the IBM Online Privacy Statement at <http://www.ibm.com/privacy/details> in the section entitled "Cookies, Web Beacons and Other Technologies", and the IBM Software Products and Software-as-a-Service Privacy Statement at <http://www.ibm.com/software/info/product-privacy>.





---

## Bibliography

This bibliography lists the books that provide additional information about your system.

---

### VM/Pass-Through Facility Library

The following table lists, by task-oriented category, the VM/Pass-Through Facility library books and their order numbers. Books cited in this document are marked with a box (■).

Title	Order Number
<b>Evaluation</b>	
VM/Pass-Through Facility: VM/Pass-Through Facility: Licensed Program Specifications	GC24-5591
<b>Administration, Operation and Service</b>	
VM/Pass-Through Facility: VM/Pass-Through Facility: Administration and Operation	SC24-5557
<b>End Use</b>	
■ VM/Pass-Through Facility: VM/Pass-Through Facility: User's Guide	SC24-5555
<b>Diagnosis and Reference</b>	
■ VM/Pass-Through Facility: VM/Pass-Through Facility: Messages	SC24-5648
■ VM/Pass-Through Facility: VM/Pass-Through Facility: Programming Reference	SC24-5556
■ VM/Pass-Through Facility: VM/Pass-Through Facility: Auto-Signon Reference	SC24-5656

---

### RSCS Library

The following table lists, by task-oriented category, the RSCS library books and their order numbers. Books cited in this document are marked with a box (■).

Title	Order Number
<b>Evaluation</b>	
&dmta5.	&dmta5nu.
&dmta0.	&dmta0nu.
<b>Planning, Installation and Service</b>	
z/VM: RSCS Networking Planning and Configuration	SC24-6320
<b>End Use</b>	
z/VM: RSCS Networking Operation and Use	SC24-6319
&dmta9.	&dmta9nu.
<b>Diagnosis and Reference</b>	
z/VM: RSCS Networking Diagnosis	GC24-6316
z/VM: RSCS Networking Messages and Codes	GC24-6318
<b>Miscellaneous</b>	
&dmta6.	&dmta6nu.
z/VM: RSCS Networking Exit Customization	SC24-6317
RSCS 3270 Secure Printing	SH24-5255

---

## z/VM Release 2.1 Library

The following table lists, by task-oriented category, each z/VM Release 2.1 book and its order number. Books cited in this document are marked with a box (■).

Title	Order Number
<b>Evaluation</b>	
&dmsa0.	&dmsa0nu.
z/VM: General Information	GC24-6286
&hcpd6.	&hcpd6nu.
<b>Installation and Service</b>	
■ z/VM: VMSES/E Introduction and Reference	GC24-6336
z/VM: Installation Guide	GC24-6292
■ z/VM: Service Guide	GC24-6325
<b>Planning and Administration</b>	
&hpcp9.	&hpcp9nu.
■ z/VM: Connectivity	SC24-6267
■ z/VM: CP Planning and Administration	SC24-6271
z/VM: Running Guest Operating Systems	SC24-6321
&hcpa8.	&hcpa8nu.
&gcta.	&gctanu.
z/VM: Performance	SC24-6301
z/VM: CMS File Pool Planning, Administration, and Operation	SC24-6261
&hcpd8.	&hcpd8nu.
<b>Operation</b>	
z/VM: Virtual Machine Operation	SC24-6334
z/VM: System Operation	SC24-6326
<b>Application Programming</b>	
z/VM: CMS Application Development Guide	SC24-6256
z/VM: CMS Callable Services Reference	SC24-6259
z/VM: CMS Application Development Guide for Assembler	SC24-6257
z/VM: CMS Macros and Functions Reference	SC24-6262
&dmsa7.	&dmsa7nu.
&dmsa8.	&dmsa8nu.
z/VM: REXX/VM User's Guide	SC24-6315
z/VM: REXX/VM Reference	SC24-6314
■ z/VM: CP Programming Services	SC24-6272
z/VM: Enterprise Systems Architecture/Extended Configuration Principles of Operation	SC24-6285
z/VM: CPI Communications User's Guide	SC24-6273
&dmsa9.	&dmsa9nu.
z/VM: CMS Application Multitasking	SC24-6258
&cn2v1.	&cn2v1nu.
External Security Interface (RACROUTE) Macro Reference for MVS and VM	GC28-1366
&cn1v0.	&cn1v0nu.
<b>End Use</b>	
z/VM: Glossary	GC24-6288

Title	Order Number
z/VM: CMS Primer	SC24-6265
■ z/VM: CMS User's Guide	SC24-6266
■ z/VM: CMS Commands and Utilities Reference	SC24-6260
z/VM: XEDIT User's Guide	SC24-6338
z/VM: XEDIT Commands and Macros Reference	SC24-6337
■ z/VM: CP Commands and Utilities Reference	SC24-6268
z/VM: CMS User's Guide and Reference	SC24-6252
z/VM: CMS User's Guide and Reference	SC24-6252
<b>Diagnosis</b>	
&dmsb8.	&dmsb8nu.
■ &dmsb7.	&dmsb7nu.
z/VM: Diagnosis Guide	GC24-6280
&hcpc2.	&hcpc2nu.
&hcpc7.	&hcpc7nu.
&hcpc0.	&hcpc0nu.
■ z/VM: Dump Viewing Facility	GC24-6284
<b>Reference Summaries</b>	
&hcpc4.	&hcpc4nu.
&dmsb9.	&dmsb9nu.
&dmsc2.	&dmsc2nu.
&dmsc0.	&dmsc0nu.
&dmsc1.	&dmsc1nu.
z/VM: Quick Reference	SC24-6111
&dmsc4.	&dmsc4nu.
&hcpc5.	&hcpc5nu.

#### Bill-of-Forms

To receive a set of books, use one of the following order numbers:

SBOF-3286	One set of unlicensed VM/ESA Release 2.1 books
LBOF-3287	One set of VM/ESA Release 2.1 books to help you diagnose system problems
LBOF-3288	One set of licensed VM/ESA Release 2.1 books

## Related Books

The following table lists other books, outside the VM/Pass-Through Facility and z/VM libraries, that are helpful when using this book. Books cited in this document are marked with a box (■).

Title	Order Number
<b>VM/ESA 370 Feature</b>	
■ &kagd.	&kagdnu.
■ &kage.	&kagenu.
■ &dmki.	&dmkinu.
■ &dmkb.	&dmkbn.

Title	Order Number
<ul style="list-style-type: none"> <li>■ &amp;csia.</li> <li>■ z/VM: CP Programming Services</li> <li>■ &amp;dmka.</li> <li>■ &amp;dmmma.</li> <li>■ z/VM: CMS Application Development Guide for Assembler</li> </ul>	<ul style="list-style-type: none"> <li>&amp;csianu.</li> <li>SC24-6272</li> <li>&amp;dmkanu.</li> <li>&amp;dmmmanu.</li> <li>SC24-6257</li> </ul>
<b>VM/XA System Product</b>	
<ul style="list-style-type: none"> <li>■ VM/XA SP: Dump Viewing Facility Operation Guide and Reference</li> <li>■ VM/XA SP: CP Programming Services</li> <li>■ VM/XA SP: CMS Application Program Development Reference</li> <li>■ VM/XA SP: Product Installation and Service</li> </ul>	<ul style="list-style-type: none"> <li>SC23-0359</li> <li>SC23-0370</li> <li>SC23-0370</li> <li>SC23-0364</li> </ul>
<b>Customer Information Control System/VM</b>	
CICS/VM: Problem Determination	SC33-0593
<b>Other</b>	
Dictionary of Computing	SC20-1699

---

# Index

## Special Characters

- ?CONFIG nickname entry 52
- ?DEFAULT nickname entry 52
- ?FINAL nickname entry 53
- ?INITIAL nickname entry 52
- \* (comment) command 182
- \* (comment) record
  - in a configuration file
    - description 233
    - placement in the configuration file 30
    - use in the configuration file 30, 44
  - in a message repository 134
- \* comment record, PVMG
  - Comment record, PVMG 283

## Numerics

- 3088 line driver
  - defining 31
    - DEFINE command (PVM) 189
    - LINK record 242
  - description of 3

## A

- accessing systems
  - PVM 4
- adding
  - links
    - DEFINE command (PVM) 189
    - LINK record 242
- administrative considerations 15
  - performance 19
  - requirements 15
  - restrictions 20
  - system security 19
- after initialization exit (DVMCUEAI) 125
- after read exit (DVMCUEAR) 126
- after write exit (DVMCUEAW) 127
- alternate operator commands 63
  - \* (comment) 182
  - AUDIT command, PVM 182
  - AUTHORIZ, PVMG 262
  - CDBC command, PVM 185
  - CMD command, PVM 186
  - CMS command, PVM 186
  - CP command, PVM 187
  - CPQ command, PVM 188
  - DEFINE 63, 189
  - DROP 67, 195
  - EXEC, PVM 196
  - LMSG 65, 197
  - alternate operator commands (*continued*)
    - LMSG (PVMG) 71
    - LOGMSG (PVM) 198
    - MSG 65, 202
    - MSG(PVMG) 71
    - PAUSE command, PVM 207
    - PMSG 65, 71, 208
    - QUERY 61
    - QUIESCE 66, 214
    - RESUME 66, 217
    - ROUTE command, PVM 218
    - SELECT command, PVM 220
    - SHOW 64, 222
    - START command, PVM 224
    - STATUS 225
    - summary 180, 261
    - TRACE command, PVM 228
    - VARY 230
- alternate route
  - defining 32, 252
  - sample use in the configuration file 33
- AMSTAMP records, in a message repository 135
- APPC/VM line driver
  - defining 31, 299
    - CMS communications directory 37
    - DEFINE command (PVM) 189
    - LINK record 242
  - description 3
- application programming 145—151
  - command response interface 146
  - programmable operator 145
  - PVMPROP/CRI message protocol 146
- APPLN record, PVMG 284
- applying
  - exits to PVM 128
- AUDIT command, PVM 182
- audit file 34
  - AUDIT command, PVM 182
  - AUDIT record 233
- AUTHORIZ
  - command, PVM 183
  - command, PVMG 262
  - record, PVMG 285
- authorizing PVM operators 33
- authorizing PVMG operators 46
- automatic node selection 33
- automating PVM initialization
  - PROFILE EXEC for CP and CMS commands 36
  - PVM profile exec for PVM commands 38
- auxiliary storage
  - requirements 15

## B

- backup language
  - changing 130
  - description of 129
- before final exit (DVMCUEBF) 128
- before initialization exit (DVMCUEBI) 125
- before read exit (DVMCUEBR) 126
- before write exit (DVMCUEBW) 127
- bibliography 353
- bill-of-forms 355

## C

- calculating values for
  - BUFFERS record 236
  - EXT record 239
  - I/O record 241
  - SET RESERVE value 107
- CCWs restriction, logical device support 21
  - cview.Multiple CVIEW virtual machines 25
- CDBC command, PVM 185
- CDBC command, PVMG 264
- changing the backup language 130
- CINIT session initiation data exit (DVMNUE6) 121
- class authority
  - class B 15
  - for MSGNOH use 248, 292
  - when configuration file contains a MSGNOH record 15
- class, privilege
  - class B 15
  - for MSGNOH use 248, 292
  - when configuration file contains a MSGNOH record 15
- CMD command, PVM 186
  - CMD command, PVM 186
- CMD command, PVMG 265
  - CMD command 265
- CMS command, PVM 186
- CMS communications directory 37
  - using with APPC/VM links 37, 299
- CMS GLOBALV command 206
- CMS subset
  - CMS command, PVM 186
  - identifying CMS subset commands to PVM 33
  - SUBSET configuration file record 254
  - using subset commands 65
- CMS SYNONYM file 206
- CMS user exits 124—128
  - after initialization (DVMCUEAI) 125
  - after read (DVMCUEAR) 126
  - after write (DVMCUEAW) 127
  - before final (DVMCUEBF) 128
  - before initialization (DVMCUEBI) 125
  - before read (DVMCUEBR) 126

- CMS user exits (*continued*)
  - before write (DVMCUEBW) 127
- command and response exit (DVMNUE5) 120
- command prefix
  - example 150
  - format 149
  - use in application programming 146
- command response
  - for application programs
    - command response interface 146
    - deciphering encoded messages 147
    - PVMPROP/CRI message protocol 146
  - specifying a language for 149
- command summary 180, 261
- commands
  - CP commands used with PVM 231—232
  - CVIEW commands 76
  - descriptions 182
  - in CMS subset mode 65
  - invoking PVM under CMS 203
  - MPVM command 199
  - operator examples 57
  - prefixes 149
  - PVM commands 182
  - specifying a language for responses 149
  - summary 180, 261
- comment (\*) command 182
- commenting
  - audit file 182
  - configuration file 30, 44, 233
  - message repository 134
- communications directory
  - using with APPC/VM links 37, 299
- CONFIG nickname entry 52
- connection types 3
- considerations
  - auxiliary requirements 15
  - performance 19, 107
  - routing 32
  - time out 35
  - TRACE command 94
- continuation character 135
- control blocks
  - DVMAAT 301
  - DVMALT 302
  - DVMAUT 304
  - DVMBUF 306
  - DVMCOMN 308
  - DVMSYSB 312
- control unit
  - support requirements on non-PVM systems 15
- CP command, PVM 187
- CP commands used for PVM performance
  - SET RESERVE command 107
- CP DIAL command
  - description 231

- CP parameter on DUMP record 238
- CP SET command 206
- CP SMSG command 232, 282
- CPQ command, PVM 188
- creating
  - audit file
    - AUDIT command, PVM 182
    - AUDIT record 34, 233
  - configuration file 28
  - message repository 132
- CTCA line driver
  - defining 31
    - DEFINE command (PVM) 189
    - LINK record 242
  - description of 3
  - restrictions 21
- customizing PVM
  - applying exits to PVM 128
  - changing the backup language 130
  - CMS user exits 124
  - enhancing performance 107
  - network management exits 122
  - programmable operator 145
  - session control exits 117
  - using a message repository 132
- CVIEW
  - accessing the CVIEW virtual machine 9
  - attendee 340
  - conference session setup
    - ATTEND panel 340
    - CONFWAIT panel 341
    - moderator 342
    - PASSWORD panel 341
  - consult session setup
    - CONSPASS panel 339
    - consult session setup (consultant) 338
    - GIVE panel 338
    - GIVEWAIT panel 340
    - IDGIVE 338
    - REDIS panel 339
  - control mode panels
    - consultant control 345
    - moderator control 346
  - control-mode panels 345
  - customizing CVIEW 165
    - changing the PK keys 166
    - concurrent users 167
    - CONSULT IDNAME 165
    - defining listed consultants 165
    - DXGINIT exec 165
    - listed consultants 165
    - NOTIFY panel 167
    - PF key functions 166
    - privileged users 168
    - to accept a default 168
    - to change a default 168
    - to run DXGINIT 168

- CVIEW (*continued*)
  - CVIEW commands 76
    - CMS command, CVIEW 76
    - CP command, CVIEW 76
    - DROP command, CVIEW 77
    - HELP command, CVIEW 78
    - PASSWORD command, CVIEW 78
    - QUERY command, CVIEW 79
    - SHUTDOWN command, CVIEW 81
    - STOP command, CVIEW 82
  - device type and model assignment 88
  - different display station types 12
  - directory entry 23
  - display-input option 11
  - DXGINIT EXEC 167
  - establishing a CVIEW session
    - conference session 83
    - consult session 83
  - establishing a session 82
  - installation planning 23
  - joining a conference 88
  - log file 89
  - machine requirements 24
  - managing a CVIEW session 85
    - conference sessions 86
    - consult sessions 85
  - messages 12
  - modes of operation 6
    - control mode 8
    - notify mode 8
    - session mode 7
    - setup mode 6
  - networking considerations 11
  - networking sessions 9
  - notify-mode 343
  - operating environment 24
  - PASSTHRU considerations 11
  - product overview 5
  - product panels 335
    - HOME panel 335
    - setup-mode panels 335
  - PROFILE EXEC 89
  - setup-mode
    - consult setup (recipient) 336
    - IDRECV panel 336
    - RECEIVE panel 336
    - RECWAIT panel 337
  - starting CVIEW 75
    - automatically 75
    - manually 75
  - technical overview 8
  - using CVIEW 75
  - virtual machine requirements 23
    - block multiplexer (BMX) 23
    - channel mode 23
    - disk space 23
    - nonshared virtual control units 23

## CVIEW (continued)

### virtual machine requirements (continued)

- privilege class 23
- user ID 23
- VCUNOSHR option 23
- virtual storage 23

## D

### DASD requirements 15

### data length restrictions

- data length restrictions 20

### default

- routing 32

### DEFAULT nickname entry 52

### DEFINE command

- description, PVM 189
- sample 63—64

### defining

- alternate routes 32, 252
- line groups 32, 239

### links

- DEFINE command (PVM) 189
- LINK record 242
- permanent 31, 242
- temporary 63, 189

### message repository 137

### message severity characters 135

### PVM configuration 27

- creating a configuration file 28
- planning 27
- sample configuration file 28

### PVM menus 137

### time stamps 135

### developing exits 114—128

- applying to PVM 128
- caution note 111
- CMS user 124—128
- network management 122—124
- session control 117
- time interval between calls to link statistics exit 254

### DIAL command 231

### DISC command, PVM 194

### DISC command, PVMG 266

#### DISC command 266

### disconnecting the PVM operator's console 194

### disconnecting the PVMG operator's console 266

### dispatching priority 108

### double-byte character restriction 256

### DROP command, PVM 68

- description 195
- DROP command, PVM 195
- samples 67

### DROP command, PVMG 72, 266

- DROP command 266
- samples 72

### dropping lines, sessions, and tasks (PVMG DROP command) 266

### dropping sessions and tasks (PVM DROP command) 195

### dump

- destination of 238
- DUMP record 34
- specifying type of 238

### dump facilities 97—98

- full dumps 97
- partial dumps 98

### DUMP record, PVM

- description 238
- example 29
- placement in the configuration file 30
- use in the configuration file 34

### DUMP record, PVMG 285

- DUMP record, PVMG 285
- use in the configuration file 47

### DVMAAT control block 301

### DVMALT control block 302

### DVMAUT control block 304

### DVMBUF (Internal Buffer Header) 306

### DVMCOMN control block 308

### DVMCUEAI (after initialization exit) 125

### DVMCUEAR (after read exit) 126

### DVMCUEAW (after write exit) 127

### DVMCUEBF (before final exit) 128

### DVMCUEBI (before initialization exit) 125

### DVMCUEBR (before read exit) 126

### DVMCUEBW (before write exit) 127

### DVMENR macro 116

### DVMNMEL (link statistics exit) 122

- setting time interval between calls 254

- time interval between calls 35

### DVMNMEL (link statistics exit), time interval between calls 35

### DVMNMES (session statistics exit) 122, 124

### DVMNUE1 (session initiation exit) 117

### DVMNUE2 (session termination exit) 118

### DVMNUE3 (origin node session request exit) 118

### DVMNUE4 (intermediate node session request exit) 118

### DVMNUE5 (command and response exit) 120

### DVMNUE6 (CINIT session initiation data exit) 121

### DVMNUE6 (session initiation exit) 121

### DVMRETN macro 116

### DVMSYSB control block 312

### DVMXTRN macro 117

## E

### EFGDCSS exec 108

### EFGTRACE exec 100

### EFGZAP exec 109



- encoded messages
  - deciphering 147
  - placement in a command response message
    - header 146
- end highlighting character 135
- enhancing PVM performance
  - SET RESERVE command 107
- entering commands in CMS subset mode 65
- entering CP commands within PVM (CP command) 187
- ERASE ALL UNPROTECTED command
  - restrictions 20
- ERRLOG files 101
- examples
  - BUFFERS record calculation 236
  - CMS communications directory 38
  - configuration file 28—30
  - EXT record calculation 239
  - I/O record calculation 241
  - message repository 141
  - operator commands 57
  - PROFILE EXEC 36
  - PROFILE PVM 39
  - SET RESERVE calculation 107
- excluding double-byte character messages 31, 45
- EXEC command, PVM 196
- exits 114—128
  - applying to PVM 128
  - caution note 111
  - CMS user 124—128
  - network management 122—124
  - session control 117
  - time interval between calls to link statistics exit 254

## F

- facilities for application programming 145—151
  - command response interface 146
  - programmable operator 145
  - PVMPROP/CRI message protocol 146
- files
  - See also* audit file
  - CMS communications manager 37
  - CMS SYNONYM 206
  - configuration file
    - See* PVM configuration file
  - created by the message compiler 210
  - PROFILE EXEC
    - for general users 206
    - for the PVM user ID 36
  - PROFILE PVM 38, 39
  - PVMCONF.DAT 29
  - UCOMDIR NAMES 37
- FINAL nickname entry 53

## G

- general use commands
  - \* (comment) 182
  - CMD command, PVM 186
  - command prefixes 149
  - CP commands
    - DIAL 231
    - MSG 232, 282
  - CPQ command, PVM 188
  - MSG 202
  - PASSTHRU 203
  - PMSG (PVM) 208
  - QUERY 211
  - SHOW 222
  - summary 180, 261
- groups
  - DEFINE command (PVM) 189
  - GROUP record 239

## H

- HELP minidisk
  - storage requirements 15
- HELP online 179

## I

- identifying
  - CMS subset commands 33
  - local node name 31
  - system language 30
  - unique resource name 31
- INIT command, PVMG 267
- INIT command 267
- Initial Menu
  - bypassing 244
  - defining PF key node settings 253
- INITIAL nickname entry 52
- installing PVM
  - administrative considerations 15
  - considerations
    - performance 107
    - routing 32
    - storage 15
    - time out 35
  - TRACE command 94
  - exits 114—128
    - See also* developing exits
  - performance options
    - dispatching priority 108
    - SET RESERVE 107
  - planning 15
  - requirements 15
  - restrictions 20
    - Logical Device Support 21
  - storage considerations 15

- intermediate node
  - description 3
  - in determining a routing path 32
- intermediate node session request exit (DVMNUE4) 118
- invoking MPVM 199
- IUCV paths, specifying maximum for PVM 245

## L

- L. command prefix 149
- LANG record, PVMG 286
- language
  - backup language 129
  - changing the backup language 130
  - default national 241
  - feature tape 129
  - identifying the system language 30
  - national 129
  - specifying
    - with command prefixes 149
- line
  - addressing 38
  - DEFINE command (PVM) 189
  - defining 242
  - defining groups 239
  - performance considerations 107
  - QUIESCE command 215
  - RESUME command 217
  - routing 32
  - START command, PVM 224
  - VARY command 230
- line driver
  - data transfer 96
  - defining 31
    - DEFINE command (PVM) 189
    - LINK record 242
  - status checks 35
  - time out handling 35
  - tracing
    - I/O activity 95
  - types 3
    - 3088
    - APPC
    - CTCA
    - TCPIP
- line groups
  - defining 32
    - DEFINE command (PVM) 189
    - GROUP record 239
- lines, tracing activity 95
- link
  - addressing 38
  - DEFINE command (PVM) 189
  - defining 242
  - defining groups 239

- link (*continued*)
  - QUIESCE command 215
  - RESUME command 217
  - routing 32
  - START command, PVM 224
  - VARY command 230
- LINK record, PVMG 286
- link statistics exit (DVMNMEL) 122
  - setting time interval between calls 254
  - time interval between calls 35
- LMSG command, PVM
  - description 197
  - LMSG command, PVM 197
  - sample 65
  - sample (PVMG) 71
- LMSG command, PVMG 268
- LMGS command 268
- local PVM system name 245
- LOCAL record
  - LOCAL record, PVMG 287
  - SESSDISC record, PVMG 295
  - use in the PVMG configuration file 45
- LOCAL record, PVMG 287
- logging messages and screen images 101
- Logical Device Support Facility 3
- logical devices
  - restrictions 21
- LOGMODE record, PVMG 288
- LOGMSG command 198
- LOGMSG command, PVM
  - LOGMSG command, PVM 198
- LOGMSG command, PVMG 269
- logo 84
- logon logo 84
- LOGONID record, PVMG 289
- LU record, PVMG 290

## M

- M. command prefix 149
- MAINT user ID
  - DASD requirements 15
  - storage requirements 15
- master record, in a message repository 134
- message commands
  - CP SMSG 232, 282
  - LMSG 65
  - LMSG (PVM) 197
  - LMSG (PVMG) 71
  - LOGMSG 198
  - MSG 65
  - MSG (PVM) 202
  - MSG(PVMG) 71
  - PMSG 65
  - PMSG (PVM) 208
  - PMSG(PVMG) 71

- message compiler
  - changing the backup language 130
  - compiling the message repository 133
  - description 210
  - files created by 210
- message record (message repository) 137
- message repository
  - AMSTAMP record 135
  - checking the accuracy of the message file 133
  - comment record 134
  - compiling the message file 133
  - continuation character 135
  - creating 132
  - creating a message file 132
  - defining message severity characters 135
  - end highlighting character 135
  - making available to users 133
  - master record 134
  - message records 137
  - PMSTAMP record 135
  - samples 141
  - screen records 137
  - special considerations 139
  - start highlighting character 135
  - start variable character 135
  - substitution variables 139
  - time stamp records 135
  - using 132
- message severity characters 135
- messages
  - defining in a message repository 137
  - LMSG command, PVM 197
  - LOGMSG command, PVM 198
  - MSG command, PVM 202
  - PMSG command, PVM 208
  - specifying a destination for operator messages 250
  - specifying format of 248
- messages, identifying 101
- messages, logging 101
- modifying links, DEFINE command (PVM) 189
- MPVM
  - command 199
  - command, return codes from 201
  - NAMES file 49
- MPVMSYS NAMES file 49
- MSG command, PVM
  - description 202
  - MSG command, PVM 202
  - sample 65
  - sample(PVMG) 71
- MSG command, PVMG 269
- MSGNOH record, PVMG 291
- MSGOPTNS record, PVMG 292

## N

- NAMES file 37, 49, 206
- national language
  - backup language 129
  - changing the backup language 130
  - default national 241
  - feature tape 129
  - identifying the system language 30
  - national 129
  - specifying
    - with command prefixes 149
- network line driver, time interval between line status checks 256
- network management exits 122—124
  - link statistics (DVMNMEL) 122
  - session statistics (DVMNMES) 122, 124
- networking terms 3
- non-PVM node, security considerations 19
- notepad function
  - PASSTHRU command description 203
  - restrictions 20
  - security considerations 19

## O

- online HELP 179
- operating PVM 57
- operator
  - commands and execs
    - See *also* alternate operator commands
    - DISC command, PVM 194
    - QUIT command, PVM 217
    - RUNPVM command 219
    - RUNPVMG command 275
    - SHUTDOWN command, PVM 223
    - summary 180, 261
  - tasks 57
- OPID record, PVMG 293
- origin node session request exit (DVMNUE3) 118
- override variable settings for current session 201

## P

- PASSTHRU command
  - description 203
  - setting defaults 206
- path
  - defining 218, 252
  - displaying 64
  - ROUTE command, PVM 218
  - ROUTE record 252
  - SHOW command, PVM 222
- PAUSE command, PVM 207
- PCCF support task
  - defining 242
  - querying 211

- PCCF support task (*continued*)
  - restrictions 21
  - sample LINK record for 31
  - sample PVMCONF.DAT file 29
  - starting 224
  - terminating 195
  - tracing 96, 228
- performance
  - commands
    - SET RESERVE 107
  - considerations 19, 107
  - dispatching priority 108
- PF keys
  - general user tasks
    - starting a PVM session 206
  - operator tasks
    - adding to PVM menus 220
    - SELECT command, PVM 220
    - settings for PVM menus 34, 253
- planning a PVM configuration 27
- PMSG command, PVM
  - description 208
  - PMSG command, PVM 208
  - sample 65
  - sample(PVMG) 71
- PMSG command, PVMG 270
  - PMSG command, PVMG 270
- PMSTAMP records, in a message repository 135
- prefix 149
- prerequisite programming 15
- privilege class
  - class B 15
  - for MSGNOH use 248, 292
  - when configuration file contains a MSGNOH record 15
- problem determination 93—98
  - gathering installation information 93
  - TRACE facility 94
- PROFILE PVM 38—39
- programmable operator
  - description 145
  - messages sent to a 146
  - PVMPROP record 250
  - PVMPROP/CRI message protocol 146
  - specifying user ID for 250
- PVM
  - accessible system types 3
  - accessing 4
  - connection types 3
  - DASD requirements 15
  - line driver types 3
  - networking terms 3
  - operating 57
  - privilege class 15
  - restrictions 20
    - CTCA 21
    - data length 20
- PVM (*continued*)
  - restrictions (*continued*)
    - Logical Device Support 21
    - Notepad function 20
    - SYS REQUEST key 20
  - routing considerations 32
  - security considerations 19—20
    - eliminating exposures 20
    - notepad function 19
    - session end function 19
    - temporary disconnect function 19
    - user session security 19
  - storage requirements 15
  - time out considerations 35
- PVM AUDIT record
  - description 233
  - example 29
  - placement in the configuration file 30
  - use in the configuration file 34
- PVM AUTHORIZ record
  - description 235
  - example 29
  - placement in the configuration file 30
  - use in the configuration file 33
- PVM AUTO record
  - description 235
  - placement in the configuration file 30
- PVM BUFFERS record
  - calculating value for 236
  - description 236
  - placement in the configuration file 30
  - use in the configuration file 36
- PVM commands 182
  - See *also* commands
  - prefixes 149
  - summary 180, 261
- PVM comment (\*) record
  - in a configuration file
    - description 233
    - placement in the configuration file 30
    - use in the configuration file 30, 44
  - in a message repository 134
- PVM configuration file
  - authorizing operators 33
  - automatically connecting users to a node 33
  - commenting 233
  - commenting in 30, 44
  - creating 28
  - defining a link 31
  - defining alternate routes 32
  - defining line groups 32
  - excluding double-byte character messages 31, 45
  - identifying a unique resource name 31
  - identifying CMS subset commands 33
  - identifying the local node name 31
  - identifying the system language 30

- PVM configuration file (*continued*)
  - order of records 30
  - records 233—257
  - sample 28
  - saving PVM information, audit file 34
  - saving PVM information, dumps 34
  - setting PF keys for PVM menus 34
  - specifying message requirements 36, 47
  - specifying storage requirements 36
  - specifying time restrictions 35
- PVM EXT record
  - calculating entry size 239
  - description 238
  - placement in the configuration file 30
  - use in the configuration file 36
- PVM GROUP record
  - description 239
  - example 28
  - placement in the configuration file 30
  - use in the configuration file 32
- PVM I/O record
  - calculating entry size 241
  - description 240
  - placement in the configuration file 30
  - use in the configuration file 36
- PVM LANG record
  - description 241
  - example 28
  - placement in the configuration file 30
  - use in the configuration file 30
- PVM LINK record
  - description 242
  - placement in the configuration file 30
  - time out conditions 35
  - use in the configuration file 31
  - used to determine routing 32
- PVM LNODE record
  - description 244
  - example 29
  - placement in the configuration file 30
  - use in the configuration file 33
- PVM LOCAL record
  - description 245
  - example 28
  - placement in the configuration file 30
  - use in the configuration file 31
- PVM MAXCONN record
  - description 245
  - placement in the configuration file 30
- PVM MAXIDLE record
  - description of 246
- PVM MAXSESS record
  - description of 246
- PVM MAXSUSP record
  - description 246
- PVM MAXTDISC record
  - description of 247
- PVM MAXUSERS record
  - description of 248
- PVM message compiler
  - changing the backup language 130
  - compiling the message repository 133
  - description 210
  - files created by 210
- PVM messages
  - defining in a message repository 137
  - LMSG command, PVM 197
  - LOGMSG command, PVM 198
  - MSG command, PVM 202
  - PMSG command, PVM 208
  - specifying a destination for operator messages 250
  - specifying format of 248
- PVM MSGNOH record
  - description 248
  - example 29
  - placement in the configuration file 30
  - use in the configuration file 36, 47
- PVM MSGOPTNS record
  - description 248
  - example 29
  - placement in the configuration file 30
  - use in the configuration file 36, 47
- PVM OPID record
  - description 250
  - placement in the configuration file 30
- PVM profile exec 38—39
- PVM PVMPROP record
  - description 250
  - placement in the configuration file 30
  - PVMPROP/CRI message protocol 146
  - use in application programming 145
- PVM SESSTERM record
  - description of 253
  - placement in the configuration file 30
- PVM SET command
  - description of 220
- PVM STATTIME record
  - description of 254
  - example 29
  - placement in the configuration file 30
  - use in the configuration file 35
- PVM SUBSET record
  - description of 254
  - placement in the configuration file 30
  - use in the configuration file 33
- PVM TCP record
  - description 255
- PVM TDISC record
  - description 255
  - example 29
  - placement in the configuration file 30

PVM TDISC record (*continued*)  
     use in the configuration file 35

PVM TIMEOUT record  
     description 256  
     example 29  
     placement in the configuration file 30  
     use in the configuration file 35

PVM XDBC record  
     description 256  
     example 28  
     placement in the configuration file 30  
     use in the configuration file 31, 45

PVMBLD command 209

PVMCMP command 210  
     changing the backup language 130  
     compiling the message repository 133  
     description 210  
     files created by 210

PVMCONF.DAT file 29

PVMG  
     \* Comment record 283  
     AUTHORIZ 285  
     AUTHORIZ record 285  
     command  
         AUTHORIZ 262  
         CDBC 264  
         CMD 265  
         DISC 266  
         DROP 266  
         INIT 267  
         LMSG 268  
         LOGMSG 269  
         MSG 269  
         QUERY 271  
         QUIESCE 273  
         RESUME 274  
         ROUTE 274  
         SELECT 276  
         SHOW 276  
         SHUTDOWN 73, 277  
         SPOOL 279  
         START 280  
         TRACE 281  
     configuration record, APPLN 284  
     LOCAL 287  
     PMSG 270  
     record  
         DUMP 285  
         LANG 286  
         LINK 286  
         LOGINID 289  
         LOGMODE 288  
         LU 290  
         MSGNOH 291  
         MSGOPTNS 292  
         OPID 293  
         PRVPROP 294

PVMG (*continued*)  
     record (*continued*)  
         ROUTE 294  
         SELECT 295  
         SESSDISC 295  
         SESSTERM 296  
         use in the configuration file 45  
         XDBC 296

PVMG LNODE record  
     description 287

PVMPROP record, PVMG 294

PVMPROP user ID  
     description 145  
     messages sent to a 146  
     PVMPROP record 250  
     PVMPROP/CRI message protocol 146  
     specifying user ID for 250

PVMSNAP EXEC 151

## Q

QUERY command, PVM  
     description 211  
     QUERY command, PVM 211  
     samples 61—63

QUERY command, PVMG 271  
     PVMG samples 69  
     QUERY command, PVMG 271

QUIESCE command  
     description 214  
     QUIESCE command, PVM 214, 217  
     reversing effect of a PVM RESUME command 217  
     samples 66, 67

QUIESCE command, PVMG 273  
     QUIESCE command, PVMG 273

QUIT command  
     description 217  
     usage by a PVM operator 69

## R

remote node  
     routing 32

remote systems  
     accessible types of 3

requirements 15  
     DASD 15

RESID record, PVM  
     description 252  
     example 28  
     use in the configuration file 31  
     used with CMS communication directory 37

resource identifier, specifying 252

restricted commands 182

restrictions  
     CTCA 21

- restrictions (*continued*)
  - data length 20
  - general 20
  - notepad function 20
  - SYS REQUEST key 20
- RESUME command, PVM
- RESUME Command, PVMG 274
- RESUME record, PVM
  - description 217
  - samples 66, 67, 72
- return codes, from MPVM command 201
- reversing a QUIESCE command (RESUME command) 217
- route
  - defining 218, 252
  - displaying 64
  - ROUTE command, PVM 218
  - ROUTE record 252
  - SHOW command, PVM 222
- ROUTE command 218
- ROUTE command, PVM
  - ROUTE command, PVM 218
- ROUTE command, PVMG 274
  - ROUTE command, PVMG 274
- ROUTE record, PVM
  - description 252
  - example 29
  - placement in the configuration file 30
  - use in the configuration file 32
  - used to determine routing 32
- ROUTE record, PVMG 294
  - ROUTE record, PVM 294
- routing 32
- RUNPVM command
  - description 219, 275
  - samples 57—59

## S

- S. command prefix 149
- samples
  - BUFFERS record calculation 236
  - CMS communications directory 38
  - configuration file 28—30
  - EXT record calculation 239
  - I/O record calculation 241
  - message repository 141
  - operator commands 57
  - PROFILE EXEC 36
  - PROFILE PVM 39
  - SET RESERVE calculation 107
- saving information
  - AUDIT record 34
  - DUMP record 34
- screen images, logging 101

- screen record, in a message repository 137
- security considerations
  - eliminating exposures 20
  - notepad function 19
  - session security 19
- SELECT command
  - PVM 220
  - PVMG 276
- SELECT record, PVMG 295
- Selection Menu
  - defining PF keys for (SELECT record) 253
  - setting the LOGMSG area (PVM LOGMSG command) 198
- sending messages
  - CP SMSG 232, 282
  - LMSG 65
  - LMSG (PVM) 197
  - LMSG (PVMG) 71
  - LOGMSG 198
  - MSG 65
  - MSG (PVM) 202
  - MSG(PVMG) 71
  - PMSG 65
  - PMSG (PVM) 208
  - PMSG(PVMG) 71
- SESSDISC record, PVMG 295
- session control exits 117
  - command and response (DVMNUE5) 120
  - intermediate node session request (DVMNUE4) 118
  - origin node session request (DVMNUE3) 118
  - session initiate data exit (DVMNUE6) 121
  - session initiation (DVMNUE1) 117
  - session termination (DVMNUE2) 118
- session initiation exit (DVMNUE1) 117
- session initiation exit (DVMNUE6) 121
- session statistics exit (DVMNMES) 122, 124
- session terminate function
  - changing the system default 253
  - security exposure 19
  - setting with PASSTHRU command 204
- session termination exit (DVMNUE2) 118
- SESSTERM record, PVMG 296
- SET command, PVM 220
- SET RESERVE command 107
- setting defaults
  - alternate routing 252
  - dump types 238
  - for PASSTHRU parameters
    - CMS GLOBALV command 206
    - CMS SYNONYM file 206
    - CP SET command 206
    - NAMES file 206
  - LANG record 241
  - message options 248
  - MSGOPTNS record 248

- setting defaults (*continued*)
  - national language 241
  - ROUTE record 252
  - routing 252
  - session terminate function 253
  - SESSTERM record 253
- severity characters 135
- SHOW command, PVM
  - description 222
  - sample use 64
  - SHOW command, PVM 222
- SHOW command, PVMG 276
  - sample use 70
- SHUTDOWN command
  - PVM 69, 223
  - PVMG 73, 277
- MSG command 232, 282
- special considerations for a message repository 139
- specifying
  - a command response language with a command prefix 149
  - an alternate operator
    - AUTHORIZ command, PVM 183
    - AUTHORIZ record 235
  - CMS Subset Mode commands (SUBSET record) 254
  - default national language (LANG record) 241
  - dump types (DUMP record) 238
  - message requirements 36, 47
  - notepad function (PASSTHRU command) 203
  - session language (PASSTHRU command) 203
  - session terminate string (PASSTHRU command) 203
  - storage requirements 36
  - target system (PASSTHRU command) 203
  - temporary disconnect (PASSTHRU command) 203
- specifying a pvmid 199
- SPOOL command, PVMG
  - description 279
- START command
  - description 224
  - PVM 224
  - PVMG 280
- start highlighting character 135
- start variable character 135
- starting
  - lines 224
  - PVM 57
  - RUNPVM command description 219
  - START command description, PVM 224
- starting MPVM 199
- STATUS command 225
- stopping
  - PVM 69
    - QUIT command, PVM 217
    - SHUTDOWN command, PVM 223

- storage considerations
  - auxiliary requirements 15
  - calculating BUFFERS record value 236
  - calculating EXT record value 239
  - calculating I/O record value 241
- storage requirements 15
- substitution variables in a message repository 139
- syntax diagrams 177, 260
- system NAMES file 49
- system programmer
  - commands
    - PVMBLD 209
    - PVMCMP 210
    - summary 180, 261
  - tasks
    - changing message destinations 130—131
    - changing the PVM translate table 131—132
    - compiling a PVM message repository 210
    - developing exits 114—128
    - enhancing PVM performance 107
    - modifying PVM messages 129—143
    - providing system applications 145—151
    - rebuilding the PVM MODULE 209
- System Request key restriction 20

## T

- T. command prefix 149
- target subtag 50
- TCP command
  - PVM 227
- TCPIP line driver
  - defining 155
    - DEFINE command (PVM) 189
    - LINK record 242
- temporary disconnect function
  - disconnect time limit 35
  - security considerations 19
  - setting defaults
    - CMS GLOBALV command 206
    - CP SET PF<sub>n</sub> command 206
    - setting maximum disconnect time allowed 255
- terminating
  - PVM 69
    - QUIT command, PVM 217
    - SHUTDOWN command, PVM 223
- time intervals
  - before terminating a temporary disconnected PVM user 255
  - between calls to the link statistics exit 254
  - between line status checks 256
- time out
  - considerations 35
  - handling 35
- time restrictions 35



- time stamp
  - controlling the display of 249
  - defining format of 135
  - in the PVM audit file 234
  - message repository records 135
  - MSGOPTNS record 249
- TRACE command, PVM
  - TRACE command, PVM 228
- TRACE command, PVMG 281
  - APPLN record, PVMG 284
  - TRACE command, PVMG 281
- trace facility
  - line driver data transfer 96
  - line driver I/O 95
  - PCCF 96
  - PVM system activity 96
  - table 95
  - table, FICON 95
  - the PVM multitasking supervisor 96
  - TRACE command description (PVM) 228
- tracing
  - line driver data transfer 96
  - line driver I/O 95
  - PCCF 96
  - PVM system activity 96
  - table 95
  - table, FICON 95
  - the PVM multitasking supervisor 96
  - TRACE command description (PVM) 228

## U

- UCOMDIR NAMES file 37
  - using with APPC/VM links 37, 299
- unrestricted commands 182
- updating PVM
  - adding links
    - DEFINE command, PVM 189
    - LINK record 242
    - permanent 31
    - temporary 63
  - changing message destinations 130
  - changing the backup language 130
  - changing the PVM translate table 131
  - creating a message repository 132—143
  - developing exits 114—128
    - CMS user exits 124—128
    - network management exits 122—124
    - session control exits 117
  - enhancing PVM performance
    - dispatching priority 108
    - SET RESERVE command 107
  - storage 18
    - BUFFERS record calculation 236
    - EXT record calculation 238
    - I/O record calculation 240

- USER control statement 108
- user NAMES file 49
- using national languages on PVM 129

## V

- VARY command 230
- VM/Pass-Through Facility
  - See PVM
- VMDUMP type dump
  - specifying on DUMP record 238

## X

- XDBC record, PVMG 296



Program Number: 5684-100



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

SC24-5557-02

