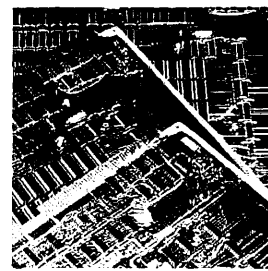
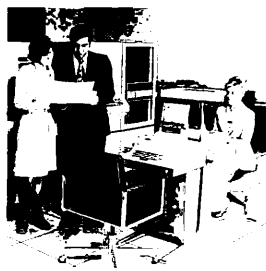
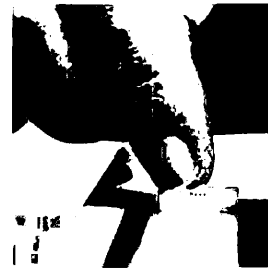
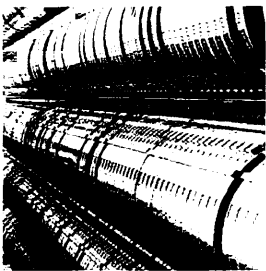


Prime Computer, Inc.

DOC7324-192L

System Operator's Guide Volume II

Revision 19.2



System Operator's Guide, Volume II

DOC 7324-192

Second Edition

**by
James Craig Burley**

Updated for Revision 19.3

**by
Jacki Forbes**

This guide documents the software operation of the Prime Computer and its supporting systems and utilities as implemented at Master Disk Revision Level 19.3 (Rev. 19.3).

**Prime Computer, Inc.
500 Old Connecticut Path
Framingham, Massachusetts 01701**

COPYRIGHT INFORMATION

The information in this document is subject to change without notice and should not be construed as a commitment by Prime Computer Corporation. Prime Computer Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Copyright © 1983, 1984 by
Prime Computer, Incorporated
500 Old Connecticut Path
Framingham, Massachusetts 01701

PRIME and PRIMOS are registered trademarks of Prime Computer, Inc.

PRIMENET, RINGNET, Prime INFORMATION, PRIMACS, MIDASPLUS, Electronic Design Management System, EDMS, PRIMEWAY, and THE PROGRAMMER'S COMPANION are trademarks of Prime Computer, Inc.

PRINTING HISTORY — System Operator's Guide, Volume II

<u>Edition</u>	<u>Date</u>	<u>Number</u>	<u>Software Rel.</u>
First Edition	July 1982	DOC5038-190	19.0
Update 1	January 1983	UPD5038-190	19.0
Second Edition	May 1983	DOC7324-192	19.2
Update 1	January 1984	UPD7324-193	19.3

Changes made to the text since the last printing have been indicated with change bars in the margin. Change bars with numbers indicate technical changes. Those without numbers indicate rewrites for clarification or additional information.

HOW TO ORDER TECHNICAL DOCUMENTS

U.S. Customers

Software Distribution
Prime Computer, Inc.
1 New York Ave.
Framingham, MA 01701
(617) 879-2960 X2053

Prime Employees

Communications Services
MS 15-13, Prime Park
Natick, MA 01760
(617) 655-8000 X4837

Customers Outside U.S.

Contact your local Prime
subsidiary or distributor.

This applies to the 9950, the 2250, and to all 50 Series processors manufactured after October 1, 1983:

WARNING

This equipment generates and uses radio frequency energy and if not installed and used properly, i.e., in strict accordance with the instructions manual, may cause harmful interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment.

Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

This applies to all other processors described in this book:

WARNING

This equipment generates and uses radio frequency energy and if not installed and used properly, i.e., in strict accordance with the instructions manual, may cause harmful interference to radio communications. As temporarily permitted by regulation, it has not been tested for compliance with the limits for Class A computing devices pursuant to Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Contents

ABOUT THIS BOOK	xiii
Hardware Operations	xiii
System Software and Maintenance	xiii
Other Prime Documentation You Should Know About	xv
Prime Documentation Conventions	xvi
PART I -- CONCEPTS	
1 WORKING WITH PRIMOS	
Introduction	1-1
PRIMOS II	1-2
PRIMOS	1-3
The Peripherals	1-11
2 THE FILE SYSTEM	
Introduction	2-1
Files and Directories	2-1
Partitions	2-6
Online Maintenance	2-8
Offline Maintenance	2-20
3 THE USER COMMUNITY	
Introduction	3-1
Responding to User Requests	3-5
Monitoring User Status	3-8
4 SYSTEM RESOURCES	
Introduction	4-1
The Spooler Subsystem	4-1
The Batch Subsystem	4-5
Magnetic Tapes	4-8
The File Transfer Service (FTS)	4-15
PRIMENET	4-17

5 MONITORING THE SYSTEM

Introduction	5-1
Monitoring System Performance	5-1
System and Network Event Logging	5-12
Supervisor Terminal Messages	5-14

PART II -- PROCEDURES

6 FORMATTING DISK DEVICES

Introduction	6-1
What to Do Before Running MAKE	6-2
Running MAKE	6-3
Special Message From MAKE	6-12
What to Do After Running MAKE	6-13
Examples	6-14

7 BACKING UP

Introduction	7-1
System Preparation Procedures	7-2
Disk-to-disk Backup (COPY_DISK)	7-15
Logical Disk-to-tape Backup (MAGSAV)	7-32
Logical Tape Verification (MAGSAV)	7-38
Physical Disk-to-tape Backup (PHYSAV)	7-41
Physical Tape Verification (PHYRST)	7-46

8 RESTORING FILES FROM BACKUPS

Introduction	8-1
System Preparation Procedures	8-2
Partial Restore From Disk (COPY)	8-2
Partition Restore From Disk (COPY_DISK)	8-6
Partition Restore From Tape (MAGRST)	8-13
Partial Restore From Tape (MAGRST)	8-17
Partition Restore From Tape (PHYRST)	8-24
Emergency Restore of Command Disk (COPY_DISK)	8-29
Emergency Restore of Command Disk (MAGRST)	8-31

9 REPAIRING FILE PARTITIONS

Introduction	9-1
What Is FIX_DISK?	9-1
When Should FIX_DISK Be Used?	9-2
What to Do Before Running FIX_DISK	9-4
The Procedure for Running FIX_DISK	9-6
Running FIX_DISK	9-6
What to Do After Running FIX_DISK	9-11
Examples of FIX_DISK	9-12
How Does FIX_DISK Work?	9-18

10 WORKING WITH PRINTERS AND PLOTTERS

Introduction	10-1
Printer Environments	10-1
The PROP Command	10-8
Dealing With Mechanical Problems	10-19

11 WORKING WITH BATCH

Introduction	11-1
Monitoring Batch	11-1
Starting the Batch Monitor	11-5
Pausing and Continuing the Batch Monitor	11-6
Stopping the Batch Monitor	11-7
Controlling Batch Jobs	11-9
Handling Batch Queues	11-9
Using FIXBAT	11-12

12 WORKING WITH FTS

Introduction	12-1
Monitoring the File Transfer Service	12-1
Managing File Transfer Requests (FTR)	12-4
Controlling the File Transfer Servers (FTOP)	12-4

PART III — REFERENCE

13 PRIMOS COMMANDS REFERENCE

Introduction	13-1
PRIMOS Commands	13-2

14 PRIMOS II COMMANDS REFERENCE

Introduction	14-1
Using PRIMOS II	14-1
Types of Commands Used Under PRIMOS II	14-3
PRIMOS II Commands	14-5

APPENDIXES

A PHYSICAL DEVICE NUMBERS

Introduction	A-1
Disks	A-1
Physical Device Numbers	A-3
Partitioning of Specific Disks	A-8

B FIXRAT

Introduction	B-1
File Structure	B-2
Running FIXRAT	B-3
FIXRAT Error Messages	B-10

C FUTIL

Introduction	C-1
FUTIL Subcommands	C-2

D LOGPRT

Introduction	D-1
Invoking LOGPRT	D-1
System Event Types	D-4
Network Event Types	D-4
LOGPRT Messages	D-5
Purging Input Event Logging Files	D-6
Information Printed in the Event Logging File	D-6
Example of Event Logging Session	D-6
LOGLST Messages	D-8
NETLST Messages	D-14

E COPY_DISK ERROR MESSAGES

Introduction	E-1
COPY_DISK Error Messages	E-1

F FIX_DISK ERROR MESSAGES

Introduction	F-1
FIX_DISK Error Messages	F-1

G	BATCH MESSAGES	
	Introduction	G-1
	Batch Messages	G-1
	FIXBAT Messages	G-19
H	PRINT_SYSLOG MESSAGES	
	Introduction	H-1
	PRINT_SYSLOG Event Messages	H-1
I	PRINT_NETLOG MESSAGES	
	Introduction	I-1
	PRINT_NETLOG Event Messages	I-1
J	DISK ERRORS	
	Introduction	J-1
	Fully Supported Disks	J-1
	Floppy Disks	J-3
K	FAM I	
	Introduction	K-1
	Starting Up FAM	K-1
	Special FAM I Command Lines	K-1
	FAM I Messages	K-4
	INDEX	X-1

About This Book

The System Operator's Guide provides guidelines for the successful daily operation of a Prime computer. In order to use this information most effectively, the reader should be familiar with the topics covered in the Prime User's Guide.

The System Operator's Guide is issued in a two-volume set. Volume I is primarily concerned with hardware operations, such as system startup and shutdown. Volume II covers system maintenance tasks, such as monitoring system resources and performing backups and restorations. It provides guidelines for performing these tasks and describes the utilities used to accomplish them.

HARDWARE OPERATIONS

Volume I of the System Operator's Guide includes information on system operation, system startup, system halt and hang handling, and system shutdown. Before beginning the tasks discussed in Volume II, the reader should be familiar with the preface and introductory chapter of Volume I. The reader should also have skimmed the portions of Volume I pertaining to the system that he or she will be using.

SYSTEM SOFTWARE AND MAINTENANCE

This book, Volume II of the System Operator's Guide, is divided into three parts and an appendix section.

Part I provides an overview of PRIMOS and the operator's tasks. Those readers who have not used Prime systems previously are encouraged to read this part before beginning work on this system. The topics covered in Part I are:

- Working with PRIMOS (Chapter 1)
- The PRIMOS file system (Chapter 2)
- The user community (Chapter 3)
- System resources (Chapter 4)
- System monitoring (Chapter 5)

Part II describes the general procedures an operator uses to keep the system running smoothly. The topics discussed in this part are:

- Disk formatting (Chapter 6)
- Performing backups (Chapter 7)
- Restoring data from backups (Chapter 8)
- Repairing disk partitions (Chapter 9)
- Monitoring print and plot requests (Chapter 10)
- Monitoring batch jobs (Chapter 11)
- Monitoring file transfer requests (Chapter 12)

Part III contains reference material on commands which are primarily used by the system operator. This part includes two chapters:

- PRIMOS commands (Chapter 13)
- PRIMOS II commands (Chapter 14)

The appendixes for Volume II provide additional reference material on various aspects of system operation. The subjects discussed are:

- Physical device numbers (Appendix A)
- PRIMOS II utilities (Appendixes B, C, and D)
- Error messages (Appendixes E, F, G, and J)
- Event log messages (Appendixes H and I)
- FAM I (Appendix K)

OTHER PRIME DOCUMENTATION YOU SHOULD KNOW ABOUT

Several Prime documents are mentioned in this text. For information on ordering these books, see page iii.

Frequent reference is made to the System Administrator's Guide, DOC5037-190. This book contains information about system building, resource allocation, and system security. This book has three updates, UPD5037-191, UPD5037-192, and UPD5037-193.

Two basic references necessary to any user of a Prime system are the New User's Guide to EDITOR and RUNOFF, FDR3104-101, which provides information on Prime's text editor and formatter; and the Prime User's Guide, DOC4130-190, containing information on PRIMOS (the operating system of every Prime computer), Prime's file system, utilities, compilers, and subroutine libraries.

The Magnetic Tape User's Guide, DOC5027-183, and its two update packages, UPD5027-184 and UPD5027-190, provide a complete description of working with Prime's magnetic tape software.

The PRIMOS Commands Reference Guide, FDR3108-190, is a dictionary of PRIMOS commands.

The PRIMENET Guide, DOC3710-193, explains PRIMENET, Prime's networking system.

The Prime 50 Series Technical Summary, DOC6904-191, describes the features of the 50 Series systems, including the advanced architecture concepts, and the software and hardware products they support.

If you are interested in writing programs to help run your system, the CPL User's Guide, DOC4302-190, will be of particular interest to you. This book describes Prime's Command Procedure Language (CPL), a powerful and flexible tool for program development. Additionally, you may want one or more of the language reference guides, the Subroutines Reference Guide, DOC3621-190, and the SEG and LOAD Reference Guide, DOC3524-192.

PRIME DOCUMENTATION CONVENTIONS

The following conventions are used in command formats, statement formats, and in examples throughout this document. Examples illustrate the uses of these commands and statements in typical applications. Terminal input may be entered in either uppercase or lowercase.

<u>Convention</u>	<u>Explanation</u>	<u>Example</u>
UPPERCASE	In command formats, words in uppercase indicate the actual names of commands, statements, and keywords. They can be entered in either uppercase or lowercase.	SLIST
lowercase	In command formats, words in lowercase indicate items for which the user must substitute a suitable value.	LOGIN user-id
abbreviations	If a command or statement has an abbreviation, it is indicated by underlining. In cases where the command or directive itself contains an underscore, the abbreviation is shown below the full name, and the name and abbreviation are placed within braces.	<u>LOGOUT</u> { SET_QUOTA } SQ
<u>underlining</u> in examples	In examples, user input is underlined but system prompts and output are not.	OK, <u>RESUME MY_PROG</u> This is the output of MY_PROG.CPL OK,
Brackets	Brackets enclose a list of two or more optional items. Choose none, one, or more of these items.	SPOOL [-LIST -CANCEL]
Braces	Braces enclose a list of items. Choose one and only one of these items.	CLOSE { filename } ALL
Ellipsis ...	An ellipsis indicates that the preceding item may be repeated.	item-x[,item-y]...

Parentheses
()

In command or statement formats, parentheses must be entered exactly as shown.

DIM array (row,col)

Hyphen
-

Wherever a hyphen appears as the first letter of an option, it is a required part of that option.

SPOOL -LIST

PART I

Concepts

1

Working With PRIMOS

INTRODUCTION

The role of the operator is a crucial one in any data processing installation. Specific assignments and responsibilities as defined by the System Administrators may vary from one computer room to another. Nevertheless, there are a number of tasks that normally "belong" to the operator.

The operations you perform will generally fall within one of the following task groups:

- Monitoring hardware and computer room conditions.
- Starting up and shutting down the system. (See Volume I of this book.)
- Monitoring system and subsystem resources. (See Chapters 4 and 5.)
- Preparing disks for use by the system. (See Chapter 6.)
- Performing backups. (See Chapter 7.)
- Restoring files and directories from backup copies. (See Chapter 8.)
- Repairing the partitions on a disk. (See Chapter 9.)

- Monitoring line printers and replacing printer paper and ribbons. (See Chapter 10.)
- Assisting users with magnetic tape assignments. (See Chapter 4.)
- Monitoring the Batch subsystem. (See Chapter 11.)
- Making sure data communications systems are functioning properly. (See Chapter 12.)
- Preventing and recovering from system halts and hangs. (See Volume I of this book.)
- General system responsibilities.

In carrying out these tasks, the operator has three areas of concern:

- The system
- The computer room
- The logbook

Volume I of this book provides introductory material in each of these areas, with particular respect to the CPU-specific aspects of the system. This volume concentrates on the software supplied by Prime for computer operations, and on the operator's interactions with the user community.

Since the operating system lies at the base of all system software, this chapter discusses Prime's operating systems, PRIMOS and PRIMOS II, and explains some of the methods by which PRIMOS expedites users' work on the system.

PRIMOS II

PRIMOS II is Prime's single-user operating system. It functions primarily as an offline environment for PRIMOS. Therefore, it is ordinarily used only during system startup. However, PRIMOS II may also be needed when PRIMOS cannot be started up or when some activity not allowed by PRIMOS must be performed.

PRIMOS II procedures for system activities such as backup and restore operations are described in Chapters 7, 8, and 9. For a general discussion of PRIMOS II, including a description of PRIMOS II commands, see Chapter 14.

PRIMOS

PRIMOS is the time-sharing operating system used by all Prime systems. It allows each user to work independently of other users and their activities. It provides:

- Time-shared access for up to 128 users per CPU
- Segmented virtual address space for programs up to 32 megabytes per user
- Access to programming languages
- Input/output control
- File system
- Interactive and noninteractive (phantom) user jobs
- Communications systems
- System utilities
- Database management

The majority of the operator's work, and the entirety of users' work, is performed under PRIMOS control.

Terminology

Many of the terms used in describing PRIMOS concepts are peculiar to Prime systems. As you read Part I, therefore, you may meet a number of new terms. Much of this terminology is described in the Prime User's Guide and its glossary. Other terms which are used in Part I of this book are defined as follows:

- access

From a user's point of view, access defines whether he or she has the right to perform some function on the system (for example, whether he or she can read a particular file). From the operator's point of view, the term access also describes the mechanisms by which the system decides who has what rights.

The primary tools for determining access on Prime systems are Access Control Lists (ACLs). ACLs are discussed in Chapter 2 and in the Prime User's Guide.

PRIMOS grants special access privileges to operators and to the System Administrator. To gain this privilege, you must log in as SYSTEM, as the System Administrator, or at the supervisor terminal.

- assigned

When a user has exclusive access to a system device, such as a magnetic tape drive, that user is said to have assigned the device. To discontinue exclusive access, the user unassigns the device.

- asynchronous line

19.1

Asynchronous lines connect terminals or Letter Quality Printers (LQPs) with the CPU. The term asynchronous refers to the ability of the line (and the terminal or LQP) to send and receive data simultaneously. Because many Prime systems use the Asynchronous Multi-Line Controller, past documentation often used the term AMLC line to describe asynchronous lines. Prime now offers both AMLC and ICS controllers that support asynchronous lines. Therefore, the term asynchronous line refers to a line connected to an AMLC or ICS controller.

- login

Before any user can use the system, he or she must log in. The procedure of logging in is called a login. During login, PRIMOS identifies the user as authorized to use the system.

- logout

When a user has finished his or her work on the system, the user must log out. The procedure of logging out is called a logout. Once you have logged out, anyone wishing to use your terminal must log in before using the system. If you leave your terminal unattended without logging out, another person could begin using the terminal under your identity; PRIMOS would be unable to tell the difference.

- mount

When a disk pack is made available for use, or a tape is set up on a tape drive, it is said to be mounted. Usually, mounting a pack or tape includes the issuing of the command that tells PRIMOS that the disk or tape is available.

When the pack or tape is removed, it is dismounted.

- network

A network is an organization of several systems into a community. These systems (such as PRIMENET) communicate via communications devices and related system software. In many installations, the system operator is responsible for several systems connected in a network.

- node

Synonymous with the term system. In this book, the term refers to a computer system, other than your own, that forms part of your network. The term is often used to distinguish your system from other systems in the network. For example: "Message from <nodename>" indicates that the message came from any other system on the network, whereas "Message from <systemname>" identifies the message as originating on the local system.

- port

A port is an information outlet connected to the CPU. Most of the ports on a system are connected to user terminals; one port on each system connects to a supervisor terminal. Some systems also have synchronous line ports, printer ports, and so on.

- subsystem

A subsystem is a collection of programs and/or processes that cooperate in providing a service to users of a system. On Prime systems, subsystems include MAGSAV/MAGRST, Batch, Spool, the File Transfer Service, EMACS, and others.

- supervisor terminal

The supervisor terminal is recognized by PRIMOS as the most privileged terminal on the system. On systems with a Virtual Control Panel (VCP), the supervisor terminal is connected to the VCP, and the VCP in turn connects to the supervisor terminal port of the CPU.

Unlike user terminals, the supervisor terminal is always logged into PRIMOS. (An exception occurs when the supervisor terminal is used as a user terminal on a Prime 2250 or 9950.)

- synchronous line

A synchronous line is a communications line. Such lines usually connect two systems, or a system and a Public Data Network (PDN). The term synchronous means that the line is able to send data only at certain times. The controllers at each end of the line agree on when to send data so as to provide bidirectional communication.

User terminals are not connected to synchronous lines; they are connected to asynchronous lines, defined earlier.

- system

In this book, the term system refers to the computer system with which the user (or operator) is currently dealing. The term is often used to distinguish one system from other systems in the network.

- user terminal

This term is used to identify any terminal other than the supervisor terminal. User terminals can be used only when PRIMOS is running.

PRIMOS Commands

The basic unit of work for any user of PRIMOS is the command. You use commands to tell PRIMOS what you want it to do for you. To learn more about how to communicate with PRIMOS via commands, read the Prime User's Guide. For complete information on PRIMOS commands for ordinary users, see the PRIMOS Commands Reference Guide.

Special commands are available for the operator that are not available, or not useful, for ordinary users. These commands are fully described in Chapter 13. In addition, the procedures by which these commands are used to perform particular tasks are discussed in the various chapters of Part II.

Some commands provide the operator with information on the status of various parts of the system. Since it is the operator's task to ensure the system's smooth operation, he or she should use these commands periodically.

The following list shows system parameters and values. It also tells which commands will print information on their status. For specific details on the individual commands, see Chapter 13.

<u>Status Item</u>	<u>PRIMOS Command</u>
Access groups	LIST_GROUP
ACL protection	LIST_ACCESS
Active Batch jobs	JOB -STATUS or JOB -DISPLAY
Assigned devices, user	STATUS USERS
Assigned mag tape drives	STATUS DEVICES
Asynchronous (user) line	STATUS USERS
Available records	AVAIL
Command device	STATUS or STATUS DISKS
Batch jobs, active	JOB -STATUS or JOB -DISPLAY
Batch jobs, executing	BATCH -DISPLAY
Batch jobs, specific	JOB job-id -DISPLAY
Batch queue names	BATGEN -STATUS
Batch queue parameters	BATGEN -DISPLAY
Batch subsystem usage	BATCH -DISPLAY
Current rev number	STATUS SYSTEM
Deferred spool files	SPOOL -LIST DEFER
Device, command	STATUS or STATUS DISKS
Devices mounted	STATUS DISKS
Devices, assigned, user	STATUS USERS
Devices, logical	STATUS DISKS
Devices, physical	STATUS DISKS
Devices, physical, user	STATUS USERS
Devices, remote	STATUS DISKS
Disk usage	USAGE -DISK
Disks mounted	STATUS DISKS
Disks, remote	STATUS DISKS
Executing Batch jobs	BATCH -DISPLAY
File transfer server(s)	FTOP -LSVS
File transfer requests	FTR -STATUS or FTR -DISPLAY
File units in use	STATUS UNITS
Free records	AVAIL
Line, user (asynchronous)	STATUS USERS
Local nodename	STATUS NET or STATUS UNITS
Logical devices	STATUS DISKS
Logins, remote	STATUS USERS
Mag tape drives, assigned	STATUS DEVICES
Mounted devices	STATUS DISKS
Mounted disks	STATUS DISKS
Network	STATUS NET
Network, type	STATUS NET
Node condition	STATUS NET
Nodename, local	STATUS NET or STATUS UNITS
Number of users	USERS
Number, user	STATUS USERS
Packnames	STATUS DISKS
Phantom users	STATUS USERS
Physical devices	STATUS DISKS
Physical devices, user	STATUS USERS
Plot files, spool	SPOOL -LIST PLOT
Print files, spool	SPOOL -LIST PRINT
Printer names	PROP -STATUS
Printer environment parameters	PROP -DISPLAY

<u>Status Item</u>	<u>PRIMOS Command</u>
Priority, user	STATUS USERS
Protection, file	LIST_ACCESS
Quotas	LIST_QUOTA or LD -SIZE
Records available	AVAIL
Records used	AVAIL
Remote devices	STATUS DISKS
Remote disks	STATUS DISKS
Remote logins	STATUS USERS
Remote systems, logins to	STATUS USERS
Remote users	STATUS USERS
Special form spool files	SPOOL -LIST FORM type
Specific Batch jobs	JOB job-id -DISPLAY
Spool files	SPOOL -LIST
Spool files, deferred	SPOOL -LIST DEFER
Spool files, special form	SPOOL -LIST FORM type
Spool files, user's own	SPOOL -LIST OWN
Spool plot files	SPOOL -LIST PLOT
Spool print files	SPOOL -LIST PRINT
Type of network	STATUS NET
Units, file, in use	STATUS UNITS
User assigned devices	STATUS USERS
User line (asynchronous)	STATUS USERS
User logins to other nodes	STATUS USERS
User number	STATUS USERS
User physical devices	STATUS USERS
User priority	STATUS USERS
User's own spool files	SPOOL -LIST OWN
Users, number of	USERS
Users, phantom	STATUS USERS
Users, remote	STATUS USERS
Volume names	STATUS DISKS
Your user-id	STATUS ME

PRIMOS Processes

The PRIMOS operating system is a time-shared operating system. This means that it allows several users to use the system at once. To do this, PRIMOS manages processes. Each user of the system owns at least one process. Often, the term user is used synonymously with the term process. The exceptions are processes that do not represent specific users, such as phantoms.

A phantom process is not connected to a user terminal. It runs programs automatically, without user intervention. Its sequence of actions comes from a command file or a CPL file. After a user creates a command file or CPL file, he or she may start up a phantom to run the file. The phantom executes the commands in the file just as if the user had invoked the file at the user terminal. When the phantom has executed all the commands in the file, or if it runs into an error requiring user intervention, it logs out. See the Prime User's Guide

for more information on phantoms, command files, and CPL files. See the CPL User's Guide for information on CPL.

In this book, the term user may refer to any user process, including phantoms started by users and people actually sitting at user terminals. Sometimes, a distinction is made between the two by referring to user phantoms (or just phantoms), and interactive users (for users logged in at user terminals).

Interactive Users: On a system with a network, there may be three kinds of interactive users:

- Users logged into the system at terminals which are connected to the same system.
- Users logged into another node on the network, who are at user terminals connected to the system.
- Users logged into the system from user terminals connected to another node on the network.

The first group of users are called local users. The second and third groups are referred to as remote users. For the most part, the distinction is unimportant to the system operator.

Phantoms: Several types of phantoms run under PRIMOS:

- User phantoms
- Batch phantoms
- Subsystem phantoms
- Network server
- Slaves

User Phantoms: The basic phantom is a user phantom. It is started up by an interactive user. In general, the operator need not worry about user phantoms.

Whenever a user phantom runs into an error requiring user input, a message will be sent to the supervisor terminal as follows:

User nnn: Phantom requested terminal input.

For the most part, this message may be ignored, since the user who started up the phantom will probably receive a similar message. The duplicate message is sent to the supervisor terminal so that a record may be kept of the aborted phantom, even if the user who started up the phantom logged out before the phantom aborted.

Batch Phantoms: When a user submits a request to the Batch subsystem, a phantom will be started up (either immediately or sometime later) to satisfy the request. This phantom is called a batch phantom. The operator rarely needs to concern him- or herself with batch phantoms.

Like user phantoms, batch phantoms notify the supervisor terminal when they terminate abnormally. In addition, the Batch subsystem sends two messages to the supervisor terminal for every batch phantom started up. The first message announces the startup of the phantom. The second message announces the termination of the phantom, and hence the satisfaction of the user request. If these messages become annoying, ask your System Administrator to use the -HUSH option on the Batch monitor.

Subsystem Phantoms: Some subsystems, such as Batch, Spool, and the File Transfer Service (FTS) require a program to run constantly. Usually these programs check for new items in a request queue. Your installation may have these subsystems, or may have other subsystems requiring such a program.

A program that must be run at all times to service a subsystem is usually started up as a phantom. These phantoms are referred to as subsystem phantoms or service processes. Although they are similar to user phantoms in many ways, the operator must often treat subsystem phantoms differently. For instance, a user phantom is usually started up to perform a specific task, but a subsystem phantom may perform several tasks.

It may be difficult or impossible to tell whether or not a subsystem phantom is in the middle of performing a task. Therefore, the subsystem usually provides a way for the operator to find out what tasks, if any, the subsystem phantom is performing. In addition, the operator can request that the subsystem phantom log itself out at a convenient time, when it is not performing any task.

Network Server: On systems that support PRIMENET, one phantom is dedicated to servicing the network. This is referred to as the network server or network server process (nsp). It is always logged in, under the name NETMAN, and need not be monitored. The only way to terminate the NETMAN process is to shut down the entire system.

Slaves: Not really phantoms at all, these are processes that perform operations on a system at the behest of a user logged into another node on the network. These processes are called slaves because they serve users on remote nodes.

A slave process represents a remote user using some resource on the local system. Usually, this resource is one of the disks that resides on the local system. For example, when a user on system A accesses a file on system B, a slave process is created on system B to perform the actual operations. This activity is usually transparent to the user on system A, but the slave process is visible to the operator and users on system B.

The fact that slaves are not connected to terminals provides their only resemblance to user phantoms. Unlike user phantoms, they generate error messages at the supervisor terminal only if the error condition indicates something seriously wrong with the system.

Generally, an operator should be concerned with slave processes only when the system is about to be shut down. The presence of slave processes on the system at that time may represent users on other nodes who are unaware of the coming shutdown.

THE PERIPHERALS

As discussed in Volume I of this book, all Prime systems include peripherals. Peripherals include the following devices:

- User terminals (to a maximum of 128)
- Magnetic tape drives (to a maximum of 8)
- Letter Quality Printers
- Parallel printers (to a maximum of 4)
- Serial line printers (to a maximum of 4)
- Plotters
- CAD/CAM work stations
- Synchronous communications boards (to a maximum of 2)
- Paper tape reader/punches
- Card reader/punches (to a maximum of 4)
- A Prime Node Controller (for RINGNET)

Since these peripherals represent scarce resources, they are managed by PRIMOS so that all users of the system can benefit equally from their availability. For example, imagine a system with one printer and two users who both want to print files at the same time. The system cannot honor both print requests immediately.

To solve this problem, PRIMOS uses two basic approaches to manage peripherals. The first and simplest approach is to allow only one user at a time to operate a peripheral device. This is referred to as exclusive assignment. For example, while one user is using a paper tape punch, no other user will be able to use it. This prevents two users from punching one tape and causing data from two separate files to become intermingled on the tape.

The other approach is to allow any number of users to request operations to be performed on a peripheral, and to allow the requested operations to be performed in sequence, as the peripheral device becomes available. This is called request queueing.

These approaches require the system operator to monitor the smooth operation of peripherals, in terms both of the operation of the physical peripheral device itself, and of its use by the user community.

Exclusive Assignment

When a user wishes to use any of the peripherals listed above, he or she must request exclusive access using the PRIMOS command ASSIGN. (The Prime Node Controller cannot be assigned by a user, because it is always assigned to the network manager, NETMAN.) If another user already has exclusive access to the peripheral, the ASSIGN command returns an error message as follows:

```
OK, ASSIGN MT0
The device is in use. MT0 (asrmt$)
ER!
```

In this case, the user desiring exclusive access to the device MT0 must wait until the other user relinquishes control. Once this happens, an attempt to assign the device will work, as follows:

```
OK, ASSIGN MT0
Device MT0 assigned.
OK,
```

The user may then invoke programs that operate the device. For example, after assigning a magnetic tape drive, the user may invoke MAGSAV to save a file or directory, as described in the Prime User's Guide.

The operator should monitor device assignment periodically to ensure that no one user is taking unfair advantage of the ASSIGN command and preventing other users from having time on a peripheral device. At the supervisor terminal, the operator has the power to use the UNASSIGN command to return a peripheral device to the pool of available devices, even when another user has the device assigned.

The operator is primarily concerned with the assignment of two system peripherals — disks and tapes. Disks are not listed as a peripheral in the list above, because they are considered system components rather than peripherals.

As system operator, you will need to perform operations other than normal user operations on disks. In these cases, you obtain exclusive access to a disk so that other users cannot reference it. You then use certain PRIMOS programs to examine and change the disk. PRIMOS itself is not able to access files on an assigned disk, but the program you run will be able to do this.

Only the operator may assign disks. To do so, he or she must first issue a special command — the DISKS command — at the supervisor terminal. The use of the DISKS command during system maintenance procedures is discussed in Chapters 7, 8, and 9. It is fully described in Chapter 13.

Request Queueing

A major drawback of the exclusive assignment approach is that while one user is using a device, another user who wishes to use it must sit and wait until the device becomes free, and then attempt to obtain exclusive access before some other user does this.

To solve this problem, PRIMOS provides request queueing for some peripherals. Users can request use of the device and continue other work. If the device is already in use, the request will be processed automatically by PRIMOS when the device becomes available.

For example, instead of allowing a user to print a file directly on a printer, PRIMOS provides a facility whereby a user makes a request to print a file. This request is placed in a queue. The files are actually printed by a special PRIMOS program. Whenever the program is ready to print a file, it looks in the queue to see if there are any requests waiting. If so, it processes the request by printing the file. When the file is printed, the program removes the request from the queue.

This principle is used to manage printers, plotters, serial printers, and Letter Quality Printers. A similar principle is used to manage various uses of the synchronous communications boards and the Prime Node Controller (PNC). Thus, the File Transfer Service (FTS) allows users to request that files be sent from one system to another. Like the spooler, the File Transfer Service adds file transfer requests to a queue, and a subsystem phantom processes these requests.

Finally, PRIMOS provides a queueing service known as Batch. The Batch subsystem queues requests for the use of phantom processes. Batch is discussed in detail in Chapter 4.

2

The File System

INTRODUCTION

A file system is used to organize data on disks so that it may be easily manipulated by users of the system. The PRIMOS file system is a flexible, easy-to-use hierarchical system.

The PRIMOS file system is central to the operation of the system. Therefore, it is the operator's task to ensure the integrity of the file system. This includes periodic checks of certain system directories.

FILES AND DIRECTORIES

All programs and data are stored in physical records located on disks. On Prime computers these records are formed into files, each of which has its own filename. The file contents may represent a source program, an object program, a runtime memory image, a set of data, a program listing, the text of an online document, or anything else the user can define and express with the available symbols. A file is stored on a peripheral storage medium such as disk or tape.

Directories

Directories, also called User File Directories (UFDs), are a special type of file. A directory contains a list of subdirectories, a list of files, and information regarding each file. Directories are nested, one beneath the other, to form a tree structure. For information on the file system and its use of the tree structure, see the Prime User's Guide.

Some UFDs, and their associated sub-UFDs and files, are delivered on a master disk pack, master disk cartridge, or master magnetic tape; they are loaded as part of your Prime software. Other UFDs are created by the operator for use by the system or by system users.

There are three general categories of directory: the Master File Directory, the User File Directory, and the subdirectory.

The Master File Directory: A Master File Directory is a special directory that contains the names of the top-level UFDs on a particular logical disk. There is one MFD for each logical disk. In most installations, users do not have full access to this level of the tree structure. As an operator, however, much of your work will be done here.

The User File Directory: The UFD is the major subdivision of the MFD, holding files, subdirectories, and information about the location and content of each file or sub-UFD within it. In most cases, users are attached to a UFD when they log in.

The Subdirectory: These directories are subdivisions either of UFDs or of other subdirectories. Separate UFDs for each user, department, project, or software product can be created by the user and will be maintained by PRIMOS. They are generally referred to as sub-UFDs.

For a complete discussion of UFDs and sub-UFDs, see the Prime User's Guide.

Files in the Master File Directory

Each Master File Directory holds the BOOT, BADSPT, and DSKRAT files, plus all top-level UFDs.

The BOOT file: This file contains the bootstrapping procedure for the disk, and is used with every new boot of the partition. For more information, see Volume I of this book.

The BADSPT file: A disk surface may have physical defects such as scratches or areas with little or no coating. The BADSPT file contains a list of all records that fall within these "badspots". It exists only on partitions which have badspots. Whenever a disk is copied, the BADSPT file is searched, in order to be sure that no information is copied onto unusable records. For more information, see Chapter 6, Chapter 7, Chapter 9, and Chapter 13.

The DSKRAT file: This file is the Disk Record Availability Table, a list of available records on the partition. This table is dynamic; that is, it changes constantly as the partition's records are used or freed. A new DSKRAT file is automatically created every time a partition is made. It is used by FIX_DISK, PRIMOS's disk repair command, and by the PRIMOS file system. The DSKRAT's name is the name of the partition. For more information, see Chapter 9 and Chapter 13.

Directories Important to the Operator

Certain top-level directories are of particular interest to the operator. Referred to as system directories, they are the UFDs needed to run PRIMOS, the utilities, and other software. These directories are described below.

The Directory DOS: The UFD DOS contains the single-user operating system, PRIMOS II, in the file *DOS64. This must be a Rev. 19 version to boot Rev. 19.

The Directory SYSTEM: The UFD SYSTEM contains all shared subsystem software, such as FORMS, and compilers for high level languages such as COBOL and FORTRAN. Also in this directory is the DISCS file, described later in this chapter.

The Directory PRIRUN: The UFD PRIRUN contains load maps and the PRIMOS runfiles (the files that are used to start up the PRIMOS operating system).

The Directory BATCHQ: The UFD BATCHQ contains the files that are used whenever Batch jobs are run. Included here are the Batch monitor run file, Batch queue definition files, and job submittal files. (See Chapter 11.)

The Directory SPOOLQ: The UFD SPOOLQ contains the files that control the environments of printer operations. Included here are the spooler monitor, spool definition files, and the spooled listing files. (See Chapter 10.)

The Directory CMDNC0: The UFD CMDNC0 contains external PRIMOS commands. External commands are those that are not a part of the operating system; examples of external commands are ED and FIX_DISK. Frequently, this directory will contain special commands that have been custom-designed for your particular system. The files for the external commands will appear in a format similar to that of the list below.

\$\$SAVE	AVAIL.SAVE	BATCH.SAVE	BATGEN.SAVE
CMPF.SAVE	CONCAT.SAVE	COPY.RUN	COPY_DISK
CPMPC.SAVE	CRMPC.SAVE	DELETE.RUN	ED.SAVE
EDB.SAVE	EDIT_PROFILE.SAVE		EVENT_LOG.SAVE
FILMEM.SAVE	FILVER.SAVE	FIXRAT	FIX_DISK.SAVE
FUTIL	HELP.RUN	HPSD.SAVE	JOB.SAVE
LABEL.SAVE	LATE.SAVE	LD.RUN	LOAD.SAVE
MAGNET.SAVE	MAGRST	MAGSAV	MAKE
MRGF.SAVE	NSED	PHYRST	PHYSAV
PMA.SAVE	PRIMOS	PRMPC.SAVE	PROP.SAVE
PROTECT.RUN	PRSER.SAVE	PRVER.SAVE	PSD.SAVE
PSD20.SAVE	REVERT_PASSWORD.RUN		RUNOFF.SAVE
RWLOCK.RUN	SEG.SAVE	SET_DELETE.RUN	SIZE.SAVE
SLIST.SAVE	SORT.SAVE	SPOOL.SAVE	TERM.SAVE
TRAMLC.SAVE	UPCASE.SAVE	VPSD.SAVE	VPSD16.SAVE

You may consider any commands not appearing in CMDNC0 (such as ATTACH, RDY, and LOGOUT) to be internal commands.

Other Directories: Other directories listed below are under the control of the operator.

<u>Directory</u>	<u>Description</u>
FORMS*	Contains files needed to run the Forms Management System (FORMS). Must be installed to use FORMS. See the <u>FORMS Programmer's Guide</u> .
FTSQ*	Contains File Transfer Service (FTS) run files, the configuration data base, queues of transfer requests, and copies of users' files for transfer. (See Chapter 12.)
LIB	Contains all libraries available on the system. Should be on logical disk 0.
LOGREC*	Contains system event logging files. Should be on logical disk 0.
PRIMENET*	Contains all files needed to run networks, including FAM II and network event logging files.

SYSKOM	Contains parameter insert files.
SYSOVL	Contains files required by COBOL. Also contains data files used by the FORTRAN 77, PASCAL, and PL/I-G compiler default driver programs.
PLIG>TOOLS	Contains the default driver program for the PL/I-G compiler.
PASCAL>TOOLS	Contains the default driver program for the PASCAL compiler.
F77>TOOLS	Contains the default driver program for the FORTRAN 77 compiler.

Additions to these directories should be done only by the operator or System Administrator. Periodically (about once each month), these directories should be checked to see if they are in order. The contents of the directories are obtained with the LD command and may be written into a file using the COMOUTPUT command. The current contents of the system directory should be compared to The proper contents. (This list should be maintained in the system logbook.)

Example of Monitoring System UFDs

An example of checking a directory for integrity follows:

OK, ATTACH SPOOLQ

OK, LD

<SYSTPB>SPOOLQ (Owner)

229 records in this directory, 229 total records out of quota of 0.

20 Files.

CRDSPL.SAVE	E.TPBPR0	E.TP.DBL	E.TP.NPR
L.DEST	L.DFLT	L.FORM	PRT003
NEW_FMT2.CPL	O.TPBPR0	O.TE.NPR	O.TP.DBL
O.TP.NPR	O_TPBPR0	O_TP.DBL	O_TP.NPR
PRT005	PRT008	PRT009	Q.CTRL

1 Segment Directory.

SPPHN.SEG

OK,

To obtain a listing of a directory sorted in reverse chronological order, so that you can see which files and directories have been modified most recently, use the -SORT_DTM option on the LD command line.

PARTITIONS

Most users identify disks and disk partitions by partition names. For example, in the pathname <BEECH>BRANCH>TWIG, the name BEECH identifies a specific disk or disk partition. PRIMOS automatically determines which physical disk drive unit is being referenced, by looking up the partition name in a list of disk partitions. For each disk partition, this list defines the actual disk on which the partition resides and the location of the partition on the disk.

Although most users identify disks by their partition names, operators often perform functions when the relationship between a disk partition and its physical disk drive has not been defined. Such functions require the specification of a disk that is not defined to the system. In these cases, physical device numbers are used. Each disk or disk partition has a physical device number that identifies the type of storage device, the drive unit on which it is mounted, the size of the partition, and its location on the disk. These physical device numbers are used in the following commands to perform special operations:

<u>Command</u>	<u>Function</u>
ADDISK	Defines the relationship between a partition name and a physical disk or disk partition to PRIMOS, so users may access the disk.
ASSIGN DISK	Allows the operator to gain exclusive access to a physical disk or disk partition, so that special operator commands (such as MAKE, COPY_DISK, and PHYSAV) can be performed on that disk.
CONFIG	Describes the physical disks that PRIMOS will use for paging (PAGDEV and ALTDEV directives) and command execution (COMDEV directive) during system coldstart.
COPY_DISK	Copies the entire contents of a physical disk or disk partition to another disk.
DISKS	Allows or disallows use of the ASSIGN DISKS command for a physical disk or disk partition.
FIX_DISK	Determines the integrity of the file system structure of a physical disk or disk partition; can also make repairs.
FIXRAT	Is an obsolete version of FIX_DISK.
MAKE	Performs the initialization on a physical disk or disk partition, so that the disk may be used. This is where the partition name for a disk is first assigned.

PHYRST	Restores the entire contents of a physical disk or disk partition from a tape created by PHYSAV.
PHYSAV	Saves the entire contents of a physical disk or disk partition onto magnetic tape.
SHUTDN	Undoes the effect of an ADDISK command by removing a partition name from the list of disk partitions, preventing further access of the disk by users.
UNASSIGN DISKS	Undoes the effect of an ASSIGN DISKS command, disallowing the operator's exclusive access to the disk. No other users may access the disk before or after an UNASSIGN DISKS command, until an ADDISK command is performed for that disk.

Appendix A describes how to determine a physical device number for a given disk.

Disk Formatting

Before a disk pack can be used on the system, it must be formatted and given a name. You do this with the MAKE utility. First add the new disk or partition number to the table of assignable devices with the DISKS command. Then the new device can be assigned and formatted with the MAKE utility. This process is described in detail in Chapter 6.

Changing the Assignable Disks Table

Before a disk or partition can be assigned, its physical device number must be added to the assignable disks table by the DISKS command. (See Chapter 13.) Once this is done, the disk can be assigned with the ASSIGN DISK command.

Under PRIMOS, devices must always be assigned to the user prior to such operations as MAKE, FIX_DISK, or COPY_DISK. Devices should be unassigned (using UNASSIGN) after completion of the operation. Devices are not assigned under PRIMOS II.

Note

The assignable disks table has space for a maximum of 10 devices. Devices may be removed from the table by the DISKS NOT command. (See Chapter 13.)

Adding Disks to the System

To define the connection between a partition name and the other four aspects of a disk partition, the ADDISK command is used. This command defines the physical characteristics of the disk to PRIMOS. PRIMOS then reads information from the specified disk partition to determine its name, and adds the name and corresponding information to a list of known disk partitions. (This list may be displayed with the STATUS DISKS command.) At this point, the disk partition is accessible by users.

When the disk is to be dismounted, the operator uses the SHUTDN command to terminate file system activity on the disk, and to delete the partition name from the list of known disk partitions. Users should be notified well in advance that the partition is to be shut down.

ONLINE MAINTENANCE

The monitoring of the file system while the system is up and running is called online maintenance. It includes periodically checking the integrity of system directories, responding to user complaints, and so on.

System directories that are important to the operator are discussed above. The directories should be checked to make certain their contents are as expected.

Sometimes, users may complain about error messages that are produced by the file system when they attempt to access files. Most of these error messages indicate user error; however, some of them may indicate that the integrity of the file system is compromised. When this happens, it is up to the operator to attempt to restore that integrity.

There are several areas of the file system that must be monitored by the operator. They are:

- Access Control Lists (ACLs)
- Disk quotas
- Disk space utilization

Messages that indicate problems with these aspects of the file system are:

No information	(ACLs)
Insufficient access rights	(ACLs)
Maximum quota exceeded	(Disk quotas)
The disk is full	(Disk space utilization)

Other messages may indicate problems with the physical integrity of the disk partition involved. These messages are as follows:

- Pointer mismatch found (not the same as "POINTER_FAULT\$")
- The directory is damaged
- Directory too large
- Bad DAM file
- Bad truncate of segment directory
- Segment directory error
- The file is too long
- Too many subdirectory levels
- Disk format does not support this revision of PRIMOS

If any of these messages appear, offline maintenance of the disk partition on which the error occurred should be performed when convenient. This is discussed below in OFFLINE MAINTENANCE.

Access Control Lists (ACLs)

The PRIMOS file system allows a user to specify who can access his or her files. This is done by specifying an access control list (ACL) for a file. Complete information on ACLs is found in the Prime User's Guide.

The System Administrator will have set up ACLs on the special system directories discussed earlier. These ACLs should be periodically checked by the operator to make certain that they have not been corrupted, allowing access to unauthorized users.

Three commands are available for the purpose of monitoring Access Control Lists (ACLs). These commands are:

<u>Command</u>	<u>Function</u>
LIST_GROUP	Lists the ACL groups to which you belong. Such groups may determine access rights to certain files and directories. Abbreviation: LG
LIST_ACCESS [objectname]	Lists the access rights for any object. Abbreviation: LAC
LIST_PRIORITY_ACCESS disk-name	Reads the contents of a priority ACL on a disk partition. Abbreviation: LPAC

The use of these commands is illustrated below.

ACL access rights are indicated by symbols:

<u>Symbol</u>	<u>Right</u>	<u>Applies To</u>	<u>Meaning</u>
R	Read	Files	File may be read.
W	Write	Files	File may be modified.
U	Use	Directories	User may attach to directory.
L	List	Directories	Directory contents may be listed.
A	Add	Directories	Directory entry may be added.
D	Delete	Directories	Directory entry may be deleted.
P	Protect	Directories	Access may be changed.
ALL		Files and directories	All of the above rights.
NONE		Files and directories	No access allowed.

These ACL symbols may be combined to specify a variety of rights. For example, the combination ALUR allows a user to attach to a directory, list and add to its contents, and read any file within it that is not otherwise protected.

The LIST_GROUP Command: LIST_GROUP lists the ACL groups to which you belong. Such groups may determine your access rights to certain files. As an operator, you may expect to be a member of a group that has special operator's rights. For example:

```
OK, LIST_GROUP
Groups are: .OPERATIONS
```

Group membership is defined by the System Administrator.

Note

User 1 is never a member of any group.

The LIST_ACCESS Command: LIST_ACCESS lists your access rights to a file or directory. The format is:

LIST_ACCESS [objectname]

where objectname may be a pathname. If objectname is omitted, access rights are given for the current directory. For example:

OK, LIST_ACCESS

ACL protecting "<Current directory>":

FLOPSY:	ALUR
MOPSY:	ALL
PETER:	ALL
SYSTEM:	ALUR
.ADMINISTRATORS:	ALL
\$REST:	NONE

OK, LIST_ACCESS CONTROL>FLOW

ACL protecting "CONTROL>FLOW":

MOPSY:	ALL
SYSTEM:	ALL
.ADMINISTRATORS:	ALL
\$REST:	LUR

In the first example, the .ADMINISTRATORS group, along with users MOPSY and PETER, have full access rights to the directory. Users FLOPSY and SYSTEM may read files (R), attach to and list the contents of the directory (LJ), and create new files or subdirectories (A). Other users of the system have no access rights.

In the second example, users SYSTEM and MOPSY, along with the group .ADMINISTRATORS, have all access rights. Other users of the system may list and use directories, and may read files.

See the Prime User's Guide for more information.

The LIST_PRIORITY_ACCESS Command: System Administrators and operators may override any user-defined ACL by creating a priority ACL. The priority ACL defines access for the entire disk.

Since it is possible to prevent users from accessing even the MFD with a priority ACL, the LIST_PRIORITY_ACCESS command allows the operator and users to read the contents of the priority ACL on any disk partition. The name of the partition must always be given.

For example:

```
OK, LIST_PRIORITY_ACCESS
Partition name must be supplied. (list_priority_access)
ER! LIST_PRIORITY_ACCESS PATCH
Priority ACL on partition "<PATCH>":
    SYSTEM:    ALL
    $REST:    NONE
```

If the partition PATCH were not protected by a priority ACL, the following would occur:

```
OK, LIST_PRIORITY_ACCESS PATCH
Priority ACL not found. <PATCH> (list_priority_access)
ER!
```

Note

When a priority ACL is active on a disk, its contents are always displayed when the LIST_ACCESS command is issued. For example:

```
OK, LIST_ACCESS

ACL protecting "<Current directory>":
    FLOPSY:    ALUR
    MOPSY:    ALL
    PETER:    ALL
    SYSTEM:    ALUR
    $REST:    NONE
Priority ACL in effect for "<Current directory>":
    .ADMINISTRATORS: ALL

OK,
```

For information on setting priority ACLs, see the description of SET_PRIORITY_ACCESS in Chapter 13.

Disk Quotas

To ensure equitable sharing of disk storage, administrators can set limits (called quotas) on the amount of storage space that top-level directories can occupy on a disk. In some facilities, operators also may control quotas. The commands for using quotas allow the operator to:

- Set a maximum storage quota on a directory (SET_QUOTA).
- Change an existing quota (SET_QUOTA).
- Examine existing quotas and current storage use (LIST_QUOTA, LD, SIZE).

SET_QUOTA is discussed in Chapter 13. The other commands are discussed below. For a more complete discussion, see the Prime User's Guide.

Measuring and Allocating Storage Space: Storage space is measured in disk records. A record can contain up to 2048 user data bytes. Thus, the number of records in a file system object equals the total number of data bytes in the object divided by 2048 and rounded up to the next whole number. However, a zero-length object (such as an empty directory or file) always contains one record. All numbers are decimal.

You may wish to examine the quota on a directory and the current storage space used by directories, files, and segment directories. The LIST_QUOTA, LD, and SIZE commands provide this information.

Using LIST_QUOTA: The LIST_QUOTA command provides the following information:

- The maximum quota on a directory
- The total number of records used by the entire subtree beginning with and including the designated directory
- The number of records used by this particular directory

The format of the command is:

```
{ LIST_QUOTA } [pathname] [-BRIEF]
{ LQ }
```

pathname gives the name of the directory on which quota information is requested. If pathname is omitted, the quota information on the current directory is listed. The -BRIEF option prints a one-line summary of the directory's quota status.

For example, to list the quota information on all top-level directories on partition SYS.B, type:

```
OK, LIST_QUOTA <SYS.B>@@
Operation illegal on MFD. <SYS.B>MFD>MFD

"<SYS.B>MFD>LOGREC*" is not a quota directory.
Total records used = 28.
Records used in this directory = 28.
```

```
Maximum records allowed on "<SYS.B>MFD>CMDNC0" = 5000.
Total records used = 3500.
Records used in this directory = 3500.
```

```
.
.
.
```

Use of the -BRIEF option outputs a one-line summary of the directory's quota status. For example:

```
OK, LIST_QUOTA UFD.1 -BRIEF
Max:      200, Used:      178, Records:      65, UFD.1
OK,
```

In this example, the maximum number of records allowed is 200. The total number of records used for this directory and its subtree is 178. The number of records used by this directory alone is 65. If you omit the pathname from the command line, the pathname is omitted from the one-line summary.

Obtaining Quota and Storage Information with LD: The LD command provides quota and storage information on the first line of its display. For example:

```
OK, LD

<SYS.B>CURTAINS (ALL access)
1150 records in this directory, 1165 total records out of quota of 0.
```

```
.
.
.
```

The number of records used by this directory, the total number of records used by the directory and its entire subtree, and the maximum number of records permitted for use by the directory and its subtree, are indicated. If the third number is 0, there is no maximum limit other than the limit of the disk.

You may wish to learn the number of records in a file or segment directory within a directory. The size of these objects, as well as of directories, is provided by the `-SIZE` option to the `LD` command. Use the format:

```
LD [pathname] -SIZE
```

Wildcards may be used to get size information for an entire directory. See the *Prime User's Guide* and the *PRIMOS Commands Reference Guide* for complete information on wildcards. For example, to display information for a partition, type:

```
OK, LD <TDISK>@@ -SIZE -NO_WAIT
```

```
<TDISK>MFD (LJR access)
```

```
2 Files.
```

```
name
  type      size
```

```
BADSPT
  sam        1
BOOT
  sam        2
TDISK
  sam        4
```

```
60 Directories.
```

```
name
  type      size  quota
```

```
AARON
  dir         9      0
AESOP
  dir        29    1000
ANDY
  dir       787      0
ANTELOPE
  dir        58      0
APPLICATIONS
  dir       173    1000
AQUA
  dir       273      0
```

```
.
.
.
```

```
OK,
```

In this example, the files `BADSPT` and `BOOT` contain, respectively, 1 and 2 records. The file `TDISK` (the `DSKRAT`) contains 4 records. The directories `AESOP` and `APPLICATIONS`, with quotas of 1000 records each, contain 29 and 173 records. The nonquota directory `ANDY` contains 787 records.

By using a specific pathname or wildcard pathname, you can request size information on a single object or on a specific group of objects.

Even greater detail may be obtained using the LD -DETAIL command. Use the format:

LD [pathname] -DETAIL

Additional information on LD appears in the PRIMOS Commands Reference Guide.

Using SIZE: The SIZE command, like the LD -SIZE command, provides the number of records in an existing file, though in a different display. The operator should be logged in with complete access rights. The command format is:

SIZE pathname [-NORM]

pathname is the name of the object whose size you wish to know. It may be a wildcard name. -NORM presents records in normalized (1 record = 880 bytes) format. SIZE can report on other file system objects as well. However, for directories, segment directories, and access categories, SIZE returns the number of entries in the object. Hence, the report returned by SIZE depends upon the type of object specified by pathname, as follows:

<u>Object</u>	<u>Report</u>
file	The size of the file in 2048-byte records (880-byte records if -NORM is specified). The number of words in the file (1 word contains 2 bytes) and the file type ("sam file" or "dam file") are also printed.
directory	The number of top-level entries in the directory and the directory type ("pwd UFD" or "acl UFD"). "pwd" = password. The size of the directory listing in words is also reported.
segment directory	The number of entries in the segment directory and the directory type ("sam SEGDIR" or "dam SEGDIR"). The maximum number of entries the segment directory can hold is also reported ("n total"). Multiplying this number by 2 yields the size of the segment directory in words. (For example, "65 total" equals a size of 130 words.)
access category	The number of access pairs (identifier: rights) in the access category.

In all cases, SIZE prints the current pathname, so that you know which object SIZE is looking at when you use wildcards.

For example, to obtain the size of all objects on the partition SYS.A, type:

```
OK, SIZE <SYS.A>@@
    9 records in sam file    "<SYS.A>MFD>SYS.A" (8804 words)
   153 entries in acl UFD    "<SYS.A>MFD>MFD" (4852 words)
    2 records in sam file    "<SYS.A>MFD>BOOT" (1092 words)
    8 entries in pwd UFD     "<SYS.A>MFD>USR.1" (153 words)
        .
        .
        .
OK,
```

If the "Disk Full" Condition is Encountered: The System Administrator can assign UFD quotas whose sum exceeds the capacity of the disk. This capability assumes that not all users will be using their full storage allotment at the same time. In effect, users "share" part of their space. This technique provides efficient use of disk space. However, it also makes it possible for users to completely fill the disk, even though none of them has exceeded (or even reached) their individual quotas.

If the disk is full when users attempt to store an object, they will get the message "The disk is full." If such a situation is reported, do the following:

1. Use the MESSAGE command to ask system users to delete unneeded storage from their directories.
2. Report the situation to the System Administrator.

Monitoring Disk Space Utilization

The AVAIL command prints, for a specified disk, the number of records used, the number of records available, and the percentage of records used. Information is given as physical records (1 record = 2048 bytes), but is also available in "normalized" form (1 record = 880 bytes).

The correct format for checking disk space utilization is:

```
AVAIL [disk] [-NORM]
```

The argument disk may be specified in one of the following manners:

<u>Argument</u>	<u>Definition</u>
partition	The name of the disk
*	"all started partitions"
-LDEV nn	The logical device number, where <u>nn</u> is represented numerically, in decimal (e.g., 2, 6, 18)

If AVAIL is given without arguments, information will be printed for the device currently attached to. For example:

```
OK, AVAIL
Volume OLIO
  44442 total records
   1070 records available
   97.6% full
```

The option -NORM may be used if records given in normalized format are desired. For example:

```
OK, AVAIL -NORM
Volume OLIO
 103428 total records (normalized)
   2490 records available (normalized)
   97.6% full
```

If the command AVAIL * is given, PRIMOS reads the file SYSTEM>DISCS and prints a table of record utilization for all partitions listed there. For example:

```
OK, AVAIL *
```

VOLUME ID	TOTAL RECS	FREE RECS	% FULL	COMMENTS	
PITHOS	140733	1984	98.6	0	4463
LKYTHS	14814	3894	73.7	1	460
POTS	44442	1069	97.6	3	31460

The two columns listed under COMMENTS give information held in the DISCS file. In this example, the comments concern each device's logical device number and physical device number.

If normalized format is requested, the table appears as:

OK, AVAIL * -NORM

VOLUME ID	TOTAL RECS	FREE RECS	% FULL	COMMENTS NORMALIZED
PITHOS	327524	4351	98.6	0 4463
LKYTHS	34476	9062	73.7	1 460
POTS	103426	2487	97.6	3 31460

Notes

- For non-ACL partitions, AVAIL requires that either the owner or the nonowner MFD password be XXXXXX, and that the DSKRAT (partition name) protection be set so that a user has read access when attached to the MFD. In most cases it is the nonowner password that is set in this fashion.
- For an ACL partition, user rights must be set to Use on the MFD and Read on the DSKRAT file.

The DISCS File: AVAIL * will not work unless the file DISCS has been built in the UFD SYSTEM. The DISCS file is a list of partition names, in column form, that has been created with the editor. In addition to the column listing partition names, other information may be included in separate columns. For example:

- The disk's logical device number
- The disk's physical device number
- Miscellaneous information, such as the backup schedule for the disk

The AVAIL command takes this information from the DISCS file and adds to it information on record utilization (determined from the system) to create its display.

Here is an example of a DISCS file:

```
OK, SLIST SYSTEM>DISCS
PITHOS 0 4463 M, W, F
LKYTHS 1 460 M, W, F
POTS 3 31460 T, TH
```

OFFLINE MAINTENANCE

As often as is reasonable, disk repair operations to ensure file system integrity should be performed on the system's partitions. The program that performs these operations is `FIX_DISK`. Running `FIX_DISK` on a partition will check it for integrity. If `FIX_DISK` finds any inconsistencies on the partition, it will report them. Additionally, if the `-FIX` option is specified on the `FIX_DISK` command line, `FIX_DISK` will attempt to repair any inconsistencies.

`FIX_DISK` can also perform other useful activities, such as reducing the number of records in use by a directory when possible (UFD compression).

If a partition should have `FIX_DISK` run on it, you will be told this when the `ADDISK` command is used for the partition. For example:

```
OK, ADDISK 1060
Starting up revision 19 partition "IONIA".
(Quota system may be incorrect; please run FIX_DISK.)
OK,
```

As `ADDISK` commands are frequently included in the system startup command file `QMDNC0>C_PRMO` or `QMDNC0>PRIMOS.COMI`, you should check the output generated during system coldstart to see if the parenthetical message was printed after the message indicating that the disk partition was started up.

If you wish to run `FIX_DISK`, see Chapter 7 for information on preparing a system for the shutdown of a partition, and then see Chapter 9 for information on running `FIX_DISK`. `FIX_DISK` command line options are also discussed in Chapter 13.

3

The User Community

INTRODUCTION

Many of your tasks as operator involve dealing with members of your system's user community. You may meet the users in person, receive phone calls from them, or send and receive interuser messages via the system software. You must understand the needs of users on your system, and be able to communicate effectively with them. For example, effective use of the PRIMOS interuser message facility could minimize user complaints of unexpected system behavior.

This chapter:

- Defines users, groups, projects, and user profiles.
- Tells how to respond to user requests.
- Tells how to monitor user status.

Users

Under PRIMOS, a user is defined as the "owner" of a user name, such as FRED. A user name, also referred to as a login name, can be up to 32 characters in length. Most user names, however, are about 8 characters long. When users log in, they specify their user names and the passwords for those names.

While the users are logged in, they are identified by their user names. For instance, if a user sends a message to the supervisor terminal, the message includes the user name of the user who sent the message.

To determine the user names of all users who are currently logged into the system, use the STATUS USERS command. For example:

OK, STATUS USERS

User	No	Line	Devices
SYSTEM	1	asr	<SYSTPB> AL077
FRED	7	5	<TDISK2>
SUSANK	8	6	<TDISK> <SYSTPB>
GENED	15	15	<TDISK2> <TDISK>
SABRAM	18	20	<TDISK2> <SYSTPB>
CORKY	19	21	<TDISK> <SYSTPB>
ANERT_D	20	22	<TDISK> <SYSTPB>
KAREEM	21	23	<TDISK> <SYSTPB>
ABDUL	24	26	<TDISK2> <SYSTPB>
CILIA	25	27	<TDISK2> <SYSTPB>
SHEILA	29	33	<SYSTPB> <TDISK>
CHILD	31	35	<TDISK2>
TRAIN	32	36	<TDISK>
WONDERLAND	65	rem	<TDISK> (from SYSTPQ)
MEMORY_MGR	66	rem	<TDISK2> (from SYSTRJ)
NETMAN	85	nsp	<SYSTPB>
BATCH_SERVICE	94	phant	<SYSTPB> (2)
METER_SERVICE	95	phant	<TDISK3> (3)
SYSTEM	96	phant	<SYSTPB> PR0
TPBMAN	98	phant	<SYSTPB> AL011
TPBMAN	105	phant	<SYSTPB> AL031
ABDUL	107	phant	<TDISK2>

OK,

The names under the "User" column are the login, or user, names. The "No" column shows the user number for each user. This number is often needed when issuing commands that do not accept user names, or when more than one user is logged in with the same name. Each user on a system always has a unique user number.

The "Line" column shows the asynchronous line number of the user terminal being used by the user. However, phantoms and remote users do not have user terminals on the system. In these cases, the "Line" column shows the type of user. "rem" means remote user. "nsp" identifies the network server process, NETMAN. "slave" means a slave user. "phant" indicates a user or subsystem phantom. "batch" indicates a running batch job.

19.2

The "Devices" column shows the disks and other peripheral devices being used by each user. Names such as <TDISK> show which disks are in use by each user. Other names such as AL011 indicate exclusive assignment of the device by the user. In this example, AL011 means that user number 98 has exclusive access to asynchronous line number 11.

The "Devices" column also indicates the other node involved for remote users, as in "(from SYSTRJ)".

Groups

Your System Administrator may have organized the user community on your system into groups. Groups serve two purposes: they make it easier for PRIMOS to check users' access, and they allow the System Administrator to merge new users into existing ACLs simply by adding them to existing groups.

The LIST_GROUP command lists the groups to which the user belongs. For example:

```
OK, LIST_GROUP
Groups are: .TPEOPLE
OK,
```

In this example, the user who entered the LIST_GROUP command belongs to one group, .TPEOPLE. Group names always begin with a dot (.), to distinguish them from user names. User names cannot begin with (.).

If a user has a complaint about insufficient access, ask the user to issue the LIST_GROUP command. This will show the groups to which the user belongs. You may discover that the user has been omitted from a group to which he or she should belong, or you may find that the user belongs to a group that is explicitly denied access to the file or directory the user is attempting to reference.

Projects

The System Administrator may have chosen to define projects for the system. A project is a collection of users working in the same area. A user may belong to more than one project, but may only be logged in under the auspices of one project at a time. This feature is often used to enhance system accounting.

When a user logs in, he or she may be asked for a project name. (The System Administrator determines whether or not users need to specify a project name at login time.) The groups to which the user belongs and the user's origin directory depend upon the project specified by the user. A user who complains about problems when logging in, or about insufficient access, may be trying to log in to the wrong project.

To list the projects for all logged-in users, issue the STATUS PROJECTS command, as follows:

OK, STATUS PROJECTS

User name	Project id	no
SYSTEM	DEFAULT	1
FRED	TURING	7
SUSANM	ACCOUNTING	14
MAXWELL	LAGRANGE	15
TURILE	TURING	18
NIX	TURING	19
OTHELLO	ACCOUNTING	20
PETER	LAGRANGE	21
FRIEDA	PEANUTS	24
CILIA	SHAMPOO	25
ORON	VICTORY	29
TRAIN	ACCOUNTING	32
WONDERLAND	PEANUTS	65
MEMORY	TURING	66
NETMAN	DEFAULT	85
BATCH_SERVICE	DEFAULT	94
METER_SERVICE	DEFAULT	95
SYSTEM	DEFAULT	96
FRIEDA	VICTORY	97
TEKMAN	DEFAULT	98
TEKMAN	DEFAULT	105

OK,

The first column shows the username, the second column shows the project that the user is logged into, and the third column is the user number (as in STATUS USERS).

In the above example, note that there are two users named FRIEDA. The first is in project PEANUTS, the second in project VICTORY. This indicates that user FRIEDA is working on two projects at the same time. This can happen when a user logs into one project, starts up a phantom, logs out, and logs in again, beginning a new project.

User Profiles

Each user who can log into the system has a user (login) name. This user name is associated with a user profile. The user profile describes the projects to which the user may log in, the groups to which the user belongs, and the origin directory or initial attach point (IAP) of the user. Both the list of groups and the origin directory for a user depend on which project the user logs into.

When a user logs in, the user profile is checked to see what directory the user should be attached to. This origin directory must exist, and the user must have the right to attach to it; otherwise, the user will not be logged in, and an error message will be displayed.

If a user complains about being unable to log in, it may be because his or her origin directory does not exist, because it resides on a disk that is currently shut down, or because it has an ACL preventing the user from attaching to the origin directory.

RESPONDING TO USER REQUESTS

To allow the operator to communicate with users, PRIMOS includes an inter-user message facility. Users may use this facility to request actions that can only be performed by the operator. Such requests may be for the following actions:

- Sending broadcast messages to all users
- Adding new UFDs to the system
- Setting quotas on directories
- Adding or replacing software in `OMDNC0` and `LIB`
- Incorporating shared segments
- Changing user priority or time slice

When such a request is received, you should log the request in the system logbook before you honor it.

Sending Broadcast Messages

If a user discovers something about which all users should be told, such as a full disk, he or she may ask you to send all users a message.

From the supervisor terminal the operator can send messages to:

- All users on the local node of the network
- A specified user on any node of the network
- The supervisor terminal of a different network node (for operator-to-operator messages)

The `MESSAGE` command is useful for giving users general information (such as system being shut down, disk full), communicating with a single user (answering questions, requesting action), or for passing information between nodes (such as remote disk available).

To send a message to all users stating that the BEEBLE partition is nearly full, the operator might type:

```
OK, MESSAGE ALL -NOW
BEEBLE PARTITION 99% FULL -- PLEASE DELETE UNNECESSARY FILES
OK,
```

For more information on the MESSAGE command see Chapter 13.

Adding New UFDs to the System

Because access to MFDs is not usually granted to users, new UFDs can only be added to MFDs by the operator or System Administrator. When a request is made to add a new UFD to the system, the operator should first determine from the user the new directory's name and the partition on which it is to reside. The new UFD name is automatically checked by PRIMOS to ensure that it does not duplicate an existing UFD name on that partition.

If you create a new directory, its quota will initially be set to zero; that is, it has no maximum quota. If you set a quota on a UFD, you limit the storage allowance on any subdirectory within the directory. If you set no limit on the UFD, its storage capacity is limited only by the physical capacity of the disk with which it is associated. (Note that a quota of zero does not signify that the directory is allowed no storage at all; rather it signifies the reverse.) Information on setting quotas is given below.

The access for a newly created UFD will default to the access at the MFD level. Therefore, you should set its access to allow ALL access for the owner of the UFD.

The operator should log in as a user (usually SYSTEM), attach to the MFD on the appropriate partition, and generate the new UFD with the CREATE command (explained in the PRIMOS Commands Reference Guide).

Note

Only the System Administrator or a Project Administrator may specify a UFD to be a user's initial attach point.

Because access to the MFD is not usually available to users, the operator must also perform all requests for UFD name changes (by using the CNAME command).

Setting Quotas on Directories

To set maximum storage quotas on UFDs, use the SET_QUOTA command. Because you must have protect access rights (if your system uses ACLs) or owner rights (if your system uses passwords), you should be logged in as the System Administrator or as an operator. When such a requirement exists, it is frequently simplest to issue commands from the supervisor terminal (if SYSTEM has ALL rights). See Chapter 13 for a description of the SET_QUOTA command.

Adding/Replacing Software in CMDNC0 and LIB

CMDNC0 and LIB are ACL or password-protected directories under operator control. They contain essential system software. New software is copied into these directories with the COPY command. All new or changed software should be debugged before installation, insofar as is feasible.

All changes to CMDNC0 and LIB should be noted in the system logbook. No new or changed software should be installed without first obtaining complete details of operation. For commands, this should include command line options and keywords as well as answers to any queries asked by the program. The proper position in loading sequences should be indicated for each library. This information should be entered in the system logbook and distributed to interested users.

Caution

When installing a new version of a command or program, it is recommended that the operator save a copy of the old version in a convenient directory until such time as the new version is thoroughly checked out and it is determined that the old version is no longer needed.

Incorporating Shared Segments

Normally, shared subsystems will be incorporated into PRIMOS at system startup time. At times, experimental subsystems may need to be incorporated for test purposes. The command sequence for this (from the supervisor terminal) is as follows:

```
OPRPRI 1
SHARE pathname segment-number [access-rights]
OPRPRI 0
```

The System Administrator will assign and coordinate shared segment usage. See the discussions of the OPRPRI and SHARE commands in Chapter 13 for complete details.

Changing Priority or Time Slice

To increase efficiency and/or system performance, priorities or time slices can be changed. Important jobs may be given special, higher priorities. Priority and time slice are changed by the CHAP command, which is described in Chapter 13.

For example, to set the priority of user 12 to 2, and the timeslice to 2.3 seconds, you would type:

```
CHAP -12 2 27
```

(The timeslice is specified in octal, as tenths of a second.)

MONITORING USER STATUS

The STATUS command, described in Chapter 13, allows the operator to monitor the status of system users. Information is output indicating active users, active devices, active disks, network status, system status, open file units, etc.

When to Use STATUS

Some typical instances in which the STATUS command might be used are:

- Prior to mounting a new disk pack to determine what physical disk assignments are available.
- After a request that all users release a given disk or disks, to determine that they have done so before shutting down that disk or disks.
- As a check that all users have logged out before shutting down PRIMOS. (No harm to the system results if the users of a particular disk are still logged in when the disk or the system is shut down. However, the user's files are closed. Information held in a buffer will be lost.)

An Example of the STATUS Command: When given from the supervisor terminal, the STATUS ALL command prints all the system information shown in the following example. (At the supervisor terminal the command STATUS will print out the same information as STATUS ALL.) A detailed description of the information follows the display.

OK, STATUS ALL

System is currently running PRIMOS rev. 19.2

4096K bytes memory in use

User SYSTEM

TEK1

File Unit	File Position	Open Mode	File Type	RWlock	Treename
126	000000110	W	DAM	NR-1W	<PRECAM>LOGREC*>LOG.11/02/82

Device	User name
MT0	BAOBAB

Usrnum	Ldevice
22	MT0

Disk	Ldev	Pdev	System
PLEIST	0	4463	
OLIGOC	1	460	
PRECAM	2	31460	
ORDOVI	5		TEK2
SILUR	6		MKTG
DEVON	7		MKTG
PERMI	11		MNFG.A
JURI	12		MNFG.B

Sem.	Value	Users
- 32	1	
- 16	177777	
- 15	1	

Full duplex network

Node	State
TEK1	****
MX.B	Up
TELENET	Up

Ring network

Node	State
TEK1	****
TEK2	Up
MKTG	Up
MNFG.A	Up
MNFG.B	Up
RES1	Up
RES2	Down

Public data network

Node
ATHNS
CNBER
RIO.A
SNGPR

VII

Pagdev = 11060 Comdev = 4463

VIII

User	No	Line	Devices
SYSTEM	1	asr	<PLEIST> SMLC00 SMLC01 AL077
SYSTEM	5	3	<PLEIST>
ASH	7	5	<PRECAM>
YGDRL	9	7	<PLEIST> (to MKTG)
VITAE	10	10	<PRECAM>
BANYON	12	12	<PLEIST>
BAOBAB	22	24	<PRECAM> <PLEIST>
TNBAUM	63	rem	<PRECAM> <PLEIST> (from RES1)
FMLY	64	rem	<PLEIST> (from TELENET)
ELEMEN	65	rem	<PLEIST> TEK2<ORDOVI> (from TEK2)
NETMAN	84	nsp	<PLEIST>
SLAVE\$	85	slave	<PLEIST>
SYSTEM	98	slave	<OLIGOC>
BATCH_SERVICE	101	phant	<OLIGOC> (2)
YTSMAN	104	phant	<PLEIST>
FTP	105	phant	<PLEIST>
SYSTEM	114	phant	<OLIGOC> <PLEIST> PRO

IX

Description of STATUS Information: The following list describes the information presented in the example of printout from the STATUS command.

<u>Section</u>	<u>Information</u>
I	The version of PRIMOS currently in use. The size of main memory, in kilobytes.
II	The user will always be SYSTEM followed by the local system name, if any. (Here, the system name is TEK1.)
III	List of all PRIMOS file units currently open. (In this example the file LOGREC*>LOG.11/02/82 is open on file unit 126.)
IV	List of magnetic tape devices currently assigned. Column 1 (Device) gives the physical device number. Column 2 (User name) gives the user-id of the user to whom the device is assigned. Column 3 (Usrnum) gives the usernumber of the user. Column 4 (Ldevice) gives the logical device number that the user has assigned

to the physical device (using the -ALIAS option of the ASSIGN command). If the user has assigned no logical number, then Ldevice is the same as Device.

- V Column 1 (Disk) is the packname of the disk partition, which is also the name of the DSKRAT file. Column 2 (Ldev) is the logical device number associated with the physical device by the ADDISK command. Logical device 0 must be the command device; the paging device or partition is not included in this list (see VIII) as it is not directly accessible by the user or the operator. Column 3 (Pdev) is the physical device number, indicating the type of device, drive unit, partition size, and offset (see Appendix A). Column 4 (System) tells the network node on which the disk is physically mounted. A blank in this column means the disk is a local one; a nodename in this column shows that this is a remote disk mounted at that node of the network.
- VI Semaphore information.
- VII Under PRIMOS, multiple network types may be in operation simultaneously. This section indicates those types currently in use. The nodename is given along with the state of that node, either Up (in operation) or Down (not in operation). The local node is indicated by ****.
- VIII The physical device numbers of the paging devices (PAGDEV and ALTDEV) and the command device (COMDEV). COMDEV is the partition at logical device 0 at the time of system startup. This information is available only if the STATUS command was issued at the supervisor terminal.

Note

This information is printed only by the STATUS ALL (or STATUS) command.

- IX List of users currently logged into the system. Column 1 (User) is the user-id of the user. Column 2 (No) is the usernumber; this is a decimal number and is usually the line number plus 2. Column 3 (Line) is the asynchronous line number of the user terminal (octal). Specially assigned (non-asynchronous) line designations follow.

19.2 |

<u>Line</u>	<u>Meaning</u>
asr	User is user 1 or the supervisor terminal using the USRASR command
batch	Batch job
nsp	Network server process (NETMAN)
phant	Phantom user
rem	User logged in remotely from another node in the network
slave	NPX slave

Column 4 (Devices) lists all partitions and assigned devices in use by a particular terminal. A disk is considered to be in use (under PRIMOS) if it contains the user's origin, home, or current directory, or if the user has any files open on that disk. Currently assigned devices are then indicated with the same device abbreviations as are used by the ASSIGN command (e.g., PR0, CR1, MT2, etc.) except that assigned asynchronous lines are shown by AL line number, and assigned disks by DK number.

Other information that may appear in this column is:

- Remote login to another system on the network (see user 9).
- Remote login from another system on the network (see users 63, 64).
- User priority (user 101 is running at priority 2). Normal user priority is 1; a priority of 1 is not printed.
- Use of a remote disk (see user 65).

Monitoring the Number of Users

The total number of system users can be obtained with the internal command USERS. This is the number of terminal, phantom, and remote users, not including the supervisor terminal user or users logged in remotely from or through the system. The dialog is:

```
OK, USERS
Users= 23
```

4

System Resources

INTRODUCTION

This chapter discusses the operator interface to the following system resources:

- The Spooler subsystem
- The Batch subsystem
- Magnetic tapes
- The File Transfer Service (FTS)
- PRIMENET

THE SPOOLER SUBSYSTEM

To allow users to print files to one or more printers in an orderly fashion, PRIMOS provides a spooler subsystem. Introduced in Chapter 1, this subsystem allows users to request that files be printed whenever a printer becomes available.

You should read the Prime User's Guide to understand how users use the SPOOL command to issue print requests. The SPOOL command stores requests in the spool queue, where they wait until a printer has printed the files, satisfying the requests. When this happens, the requests are removed from the spool queue.

Examining the Spool Queue

The contents of the spool queue are monitored with the command:

SPOOL -LIST

Each listing contains the following information: user-id, PRT number, time of spool request, filename, file size options, form request (if any), defer time (if any), delivery location.

Here is an example of a spool queue listing:

OK, SPOOL -LIST
[SPOOL rev 19.2]

System MEGA									
user	prt	time	name	size	opts/#	form	defer	at: BLD	
FRED	001	10:04	RT983	231		WHITE		1	
BARNEY	002	10:11	LOGOS.3	11	2	WIDE		1	

Spooler Phantoms

Printers are controlled by phantoms. These phantoms use the ASSIGN command to acquire exclusive access to printers. For example, a phantom that prints on printer PR0 will assign device PR0. A phantom that controls a printer is called a spooler phantom, because it runs a program called the spooler (or sometimes the despooler).

The spooler program periodically checks the spool queue on the system to see if there are any files to be printed. If there are, it starts printing one of them. When it finishes, it checks the spool queue again.

Printer Environments

Because many installations have more than one printer, the spooler subsystem allows the System Administrator and operator to control which files are printed on which printers. This is done by defining printer environments. An environment defines the way in which any phantom using it will handle the files it prints: what printer it will use, what disks it will search for queued jobs, what size jobs it may accept, what page margins it will use, and so on. By doing this, it also defines the types of files the printer can handle. Thus, one printer may accept only short files; another may print any file requiring extra-wide paper; and so on.

Environments are usually set up by the System Administrator, but they may be set up by the operator or the user. The operator has the power to modify an environment, no matter who created it.

Whenever the operator starts a spooler phantom, he or she selects the environment it will use. There may be more than one environment available for each printer. However, a printer may use only one environment at any given time.

Each environment on a system has a unique name. This name is up to 30 characters long, and may contain only the letters A-Z, the digits 0-9, and the symbols `_ . * # $ & - /`.

Remote Spool Queues

Many installations have more than one system in the network. Therefore, in addition to reading the spool queue on the local system, the spooler subsystem is capable of searching all of the spool queues on the network.

Requests found on remote spool queues will be printed just as if they were found on the local spool queue. However, a phantom will always check the local spool queue before it checks remote spool queues for files. Thus, local requests have absolute priority over remote requests.

The PROP Command

Spooler phantoms are manipulated by using the PROP command. The PROP command is used to start up or shut down a spooler phantom. In addition, it may be used to affect the operation of the spooler phantom, even when the spooler phantom is not running.

When the operator uses the PROP command to operate on a printer environment or spooler phantom, he or she specifies the environment name on the command line. This tells PROP which environment, and hence which spooler phantom, is being referred to by the operator.

When a spooler phantom is running, it is said to be servicing a particular print environment. To see which of the environments are defined on your system, enter the PROP -STATUS command. This command will list the defined environments, and also display an indicator of whether each environment is being serviced by a phantom. For example:

```
OK, PROP -STATUS
[PROP rev 19.2]
```

```
TPBPR0      started
TP.NPR      started
TP.DBL      started
WHITE       stopped
OK,
```

In this example, the environments TPBPR0, TP.NPR, and TP.DBL are all being serviced by phantoms. The environment WHITE is defined, but it is not being serviced. (It may require the same printer as does one of the other environments. If this is so, then the operator cannot start up WHITE without first stopping the phantom servicing that other environment.)

Whenever the operator changes the type of paper in a printer, he or she also starts up a new printer environment. This notifies the spooler subsystem that a form change has been made and prevents user files from accidentally being printed on the wrong paper.

See Chapter 10 for more information on using PROP.

User Print Requests

When users issue the SPOOL command to request that files be printed, they sometimes use the -FORM and -AT options to specify the type of paper and the printer to be used. The spooler subsystem matches the form name specified by -FORM and the destination name specified by -AT to the list of accepted form and destination names for each environment being serviced. Only if the form and destination names supplied by the user match names in the environment will the file be considered for printing by the phantom servicing that environment.

It is possible for a user to spool a file that can not be printed by any environment. This can happen if the user specifies an illegal combination of form and destination name, or if the file is larger than the limit on file size specified in the environment.

If a user complains that a file is not being printed, check the print request using SPOOL -LIST, and use PROP to display the parameters for all environments on the network. This will tell you which environments are able to print the user's file. It is possible that there are no such environments, in which case you might ask the user to re-spool the file using a legitimate combination of options and file size. It is also possible that environments do exist that consider the request to be eligible. However, these environments may have no phantoms servicing them, or the phantoms may already be busy printing other user's files. In this case, tell the user that the file will be printed later, when a printer becomes available.

Operating Line Printers

In addition to overseeing the line printers in general, the operator is usually responsible for:

- Removing listings from the printer, separating them by user (banner name before each file), and placing them in a specified distribution area. (This may not be necessary for small systems.)
- Monitoring the spool queue with the SPOOL -LIST command. (See Chapter 10.)
- Reloading paper and ribbons in the line printer as required.
- Changing paper to print special forms requests. (Use the SPOOL -LIST command to see if any such requests are outstanding.) It is good procedure to schedule the printing of special forms for a specific time of day: for example, directly after performing backups at the start of a shift.
- Vacuuming the printers twice a day, or more often when necessary.

Plotters

In addition to queueing requests for printers, the spooler subsystem is able to queue requests for plotters. Plotter environments are similar to printer environments, except that the device specified by a plot environment is always named PLOT. The SPOOL command also accepts a -PLOT option, indicating that the file is to be plotted, not printed. This prevents a plot file from being sent to a printer.

In a plotter environment, the System Administrator or operator can specify whether the phantom servicing the environment should accept both print and plot requests, or only plot requests. In the listing generated by a SPOOL -LIST command, plot requests are shown by a "P" in the "opts/#" column.

THE BATCH SUBSYSTEM

There are times when a user needs to do a large amount of computational processing. If the user performed this processing interactively at the terminal, the user could not begin any other tasks at that terminal until the processing was completed. Other users of the system would notice a performance degradation, particularly if several users began large computational jobs at once. In an extreme case, the system could spend a significant amount of time juggling these large jobs, and less time actually performing them. In addition, the users would spend a significant amount of their time waiting for the jobs to complete.

To solve this problem, the Batch subsystem allows a user to submit a job into a queue. This job is a request to execute a command or CPL file. Frequently, the command or CPL file submitted performs a large amount of processing. The Batch subsystem waits until other batch jobs are finished to execute these jobs. The System Administrator sets up one or more batch queues into which jobs may be submitted, and defines the relative priorities of the queues. The system operator monitors the Batch subsystem, and may also be asked to prevent batch jobs from executing during the peak hours of system usage by interactive users.

Operator Responsibilities for Batch

Generally, the System Administrator is responsible for configuring the Batch subsystem and maintaining its database. These operations are explained in the System Administrator's Guide. The operator is responsible for starting and stopping the Batch monitor when the system (or the Batch subsystem) is brought up or down, and for helping with users' jobs when necessary.

There are two main reasons for operator intervention in user jobs. If a job is holding up the queue (for example, because of an infinite loop, or because the job is waiting for some unavailable resource), the operator may abort the job. If a user knows that a job will need a particular resource, the user may ask the operator to hold that job in the queue until the resource is available. When the resource is available, the operator can release the job.

Batch Elements

The Batch subsystem consists of the following elements:

- Jobs
- Queues
- The Batch monitor
- Batch phantoms

A user request to the Batch subsystem to execute a command or CPL file is called a job. This request is placed in a queue until it is honored by the batch monitor. The batch monitor honors the request by starting up a batch phantom to process the job.

Every batch job has a job id. This consists of an external name, which is the name of the command or CPL file submitted by the user, and an internal name, which is assigned by the Batch subsystem to identify uniquely every job in the system. Job ids are used by the operator when users' jobs are manipulated.

The System Administrator may define from one to sixteen batch queues, each with its own name and characteristics. Users who submit batch jobs may specify the queues in which their jobs should be placed. If they do not specify queues, the job submission program automatically determines an appropriate queue for each job.

The program that actually starts up batch jobs is called the batch monitor. This program runs as a subsystem phantom. It is started up during system coldstart, as part of the CMDNC0>C_PRMD or CMDNC0>PRIMOS.COMI file. It not only starts up batch jobs, but updates the queue when batch phantoms log out, indicating that the corresponding jobs have completed. The batch monitor is always logged in as user BATCH_SERVICE.

To execute the requested command or CPL file, the batch monitor starts up a batch phantom. This phantom is logged in with the same login name as the user who submitted the job, including the same project name and group names. Therefore, the batch phantom has the same privileges enjoyed by the submitting user.

Batch Commands

There are three essential commands in the Batch subsystem. They are:

<u>Command</u>	<u>Function</u>
JOB	Submit, display, and control user jobs
BATCH	Start up, shut down, and display the state of the batch monitor and subsystem
BATGEN	Define, modify, and delete batch queues

In addition, there are two programs used by both the System Administrator and the operator. They are:

<u>Program</u>	<u>Function</u>
INIT	Initialize the batch database
FIXBAT	Repair and compress the batch database

The use of these commands and programs by the operator is fully discussed in Chapter 11. The use of the JOB, BATCH, and BATGEN commands by the user is discussed in the Prime User's Guide and the PRIMOS Commands Reference Guide. The use of the BATGEN command for the System Administrator is discussed in the System Administrator's Guide.

MAGNETIC TAPES

Magnetic tape provides a relatively inexpensive storage medium and is used extensively. Operators routinely perform several tasks connected with magnetic tape and magnetic tape drives, including:

- Setting tape drive assignment mode.
- Assigning and unassigning tape drives.
- Turning on tape drives.
- Mounting and unmounting tapes.
- Responding to special user options.
- Performing general maintenance of the tapes and drive unit.
- Providing backup copies of files and programs. (See Chapters 7 and 8.)

The most important commands associated with the operator's magnetic tape responsibilities are:

<u>Command</u>	<u>Function</u>
SETMOD	Sets drive assignment mode
ASSIGN	Allocates tape drives
REPLY	Replies to a user request for a tape drive
STATUS	Gives status of system use
UNASSIGN	Releases tape drive from user

The SETMOD, REPLY, and UNASSIGN commands for the operator can only be issued from the supervisor terminal. (The UNASSIGN command can be issued at any user terminal, but it only affects that user. From the supervisor terminal, it affects any user on the system.) For more information regarding magnetic tape and magnetic tape drives, refer to the Magnetic Tape User's Guide.

Setting the Mode of Assignment

The operator can choose from three modes of tape drive assignment:

- Users can assign tape drives without operator intervention, unless special assistance is needed. This is the default mode.

- Users must channel all assignment requests through the operator.
- Users are not permitted to assign tape drives at all.

The SETMOD command establishes the assignment mode and can only be issued from the supervisor terminal. If SETMOD is not issued, the default mode (user assignment) prevails. The SETMOD command can only be issued at the supervisor terminal. There are three assignment modes:

<u>Mode</u>	<u>Description</u>
User mode	Users can assign tape drives without operator assistance unless options require special intervention. This is the default mode. The command to enter user mode is: SETMOD -USER
Operator intervention mode	Users must channel all assignment requests through the system operator. The command to enter operator intervention mode is: SETMOD -OPERATOR
No-assignment mode	Users are not permitted to assign tape drives at all. The command to enter no-assignment mode is: SETMOD -NOASSIGN

The SETMOD command is described in detail in Chapter 13.

Allocating Tape Drives

Each magnetic tape operation requires the use of at least one tape drive. Tape drives are reserved with the ASSIGN command. ASSIGN associates the drive's physical device number with the number of the user who issued the ASSIGN command. As long as the usernumber and device number correlation exists internally, the user has exclusive access to the drive. Access privileges are relinquished with the UNASSIGN command.

The ASSIGN command reserves magnetic tape drives for users, COMINPUT files, and CPL programs. It is frequently used to request operator assistance in assigning a drive or mounting a tape.

In addition to the default assignment, which simply designates a particular tape drive, the user can ask the operator to:

- Assign any available tape drive.
- Assign a tape drive with certain features, such as special density settings.
- Assign a particular tape drive when it becomes available.
- Mount a particular tape on an indicated or available drive.
- Assign a particular or a random tape drive, and give it a user-chosen logical device number, or "alias," with which the user will subsequently reference the assigned tape drive.

Information on the ASSIGN command for operators is found in Chapter 13. Information on the ASSIGN command for users is in the PRIMOS Commands Reference Guide.

Replying to User Requests

Users will request magnetic tape assignments (via the ASSIGN command, described above), and must be informed of the status of their requests. The REPLY command is the operator's method of communicating with the requesting user. It is preferred over the MESSAGE command when the user is assigning magnetic tape, since REPLY communicates successfully with batch jobs, user phantoms, and command or CPL files. The message sent by a MESSAGE command can only be read by an interactive user.

The REPLY command can only be issued at the supervisor terminal. It allows the operator to:

- Approve a simple request (in operator intervention mode).
- Inform the user which tape drive has been assigned when the user has requested "any drive" (MTX).
- Request repetition of an ASSIGN message.
- Inform a user that a special request has been fulfilled.
- Deny a request.

Whenever a user's ASSIGN command line necessitates operator intervention, the request appears at the supervisor terminal.

Users may request assignment of magnetic tape drives in either of two ways:

- By physical device number (pdn), appearing at the supervisor terminal as:

```
***** MAGTAPE REQUEST *****
From user-id (username) : ASSIGN MTpdn [options]
```

- By logical device number (ldn), appearing at the supervisor terminal as:

```
***** MAGTAPE REQUEST *****
From user-id (username) : ASSIGN MTX -ALIAS ldn [options]
```

The user-id and username identify the originator of the request.

The message is repeated at the supervisor terminal until acknowledged by the operator with REPLY.

Responding to User Requests

Operators respond to user requests by:

1. Determining the appropriate tape drive to be used, based upon the specified options.
2. Performing all requested tasks.
3. Mounting the correct tape.
4. Issuing the appropriate REPLY command.

If user mode is selected, you will have to respond only when options are requested, or when drive assignments are requested by logical device number. If operator intervention mode is selected, you must respond to all requests. If no-assignment mode is selected, you will not receive requests from users, and they will be unable to assign tapes.

Some Examples of How to Fulfill ASSIGN Requests

Responding to Simple ASSIGN MTpdn Requests: Requests appear at the operator's terminal containing username, user number, and a command line. For example:

```
***** MAGTAPE REQUEST *****
from SHANIN (user 7) : MT1
```

This indicates that user SHANIN (designated by PRIMOS as user number 7) requests physical device MT1.

Since no options are requested, in user mode this request would be approved or rejected (depending on MT1's availability) without need for operator intervention.

In operator intervention mode, the user's request would appear at the supervisor terminal. The operator must always respond. If the request cannot be approved, inform the user (in this case, number 7) of this by issuing the command:

REPLY -7 -TAPE ABORT

If the request can be approved, type:

REPLY -7 -TAPE GO

This indicates that SHANIN now has exclusive access to physical device MT1.

Responding to ASSIGN MTpdr Options: More complicated requests will require additional actions by the operator. The message:

***** MAGTAPE REQUEST *****
from HARRIET (user 11): MT3 -TPID EXEC -RINGON -WAIT

indicates that user HARRIET (user number 11) has requested assignment of tape drive number 3, with the tape EXEC loaded, and with the write ring on. Additionally, the user is willing to wait until drive MT3 is available.

Here, the operator's procedure would be:

1. Determine the availability of drive MT3.
2. Locate the tape identified as "EXEC".
3. Mount the tape with the ring on.
4. Issue the command: REPLY -11 -TAPE GO.

Responding to MTX Requests: With requests specifying the MTX -ALIAS option, operators get a message in the format:

***** MAGTAPE REQUEST *****

From BOB (user 34): MTX -ALIAS MT0 -TPID JEN -DENSITY 1600

The operator's response would be:

1. Locate the tape marked "JEN".
2. Mount the tape on an available drive (assume here, MT1).
3. Set the density switch to 1600 bpi.
4. Issue the command: REPLY -34 -TAPE 1.

Information Gathering

Some typical areas of concern for the operator are listed below. Appropriate actions are indicated:

- To clarify an unintelligible request from a user, type:

REPLY -usernum -TAPE RESEND

The ASSIGN command request from usernum will reappear.

- To repeat the most recent magnetic tape request, type:

REPLY -TAPE RESEND

- To display all unanswered tape requests, type:

REPLY -ALL RESEND

- To determine the availability of a requested drive, use the STATUS DEVICES command, described below.

Determining the Current Status of Users and Magnetic Tape Drives

The system operator can obtain a quick list of the magnetic tape drives currently in use by typing STATUS DEVICES. The information returned might look something like this:

OK, STATUS DEVICES

Device	User name	Usrnum	Ldevice
MT1	SHANIN	7	MT0
MT2	KARP	12	MT0
MT3	KARP	12	MT1

Only currently assigned magnetic tape devices are listed.

The operator can also tell who owns what peripheral devices by using the STATUS USERS command. Peripheral devices include magnetic tape drives, card readers, and punches:

OK, STATUS USERS

User	No	Line	Devices
SYSTEM	1	asr	<SZY> SMLC00 SMLC01
BOBBY	5	3	<POE> MT0
HARRIET	8	6	<SZY> MT1
JEN	12	11	<POE> MT2

See Chapter 3 for a thorough discussion of the STATUS command.

Releasing Tape Drives

There are times when the operator must revoke exclusive access to a magnetic tape drive from a user. This can be done with the UNASSIGN command from the supervisor terminal. For example:

OK, UNASSIGN MT0

This command can be issued in all tape assignment modes. (Only the user who assigned an alias can use the alias number when unassigning a drive. The actual device number must be specified in the UNASSIGN command at the supervisor terminal.)

For example, suppose that user 17 assigns MT1 -ALIAS MT2 and also assigns MTX -ALIAS MT0. If the operator chooses physical drive MT2 as MTX, the effective internal relationship can be represented as:

<u>Usernum</u>	<u>Physical Device Number</u>	<u>Logical Device Number</u>
17	MT1	MT2
17	MT2 (formerly MTX)	MT0

Note that this representation is similar to the table displayed by STATUS DEVICES. It should be noted that every magnetic tape drive has a default logical device number. This number is the same as the drive's physical device number, unless changed with the -ALIAS option.

The operator could release these drives with the commands: UNASSIGN MT1 and UNASSIGN MT2.

In operator intervention mode, when a device is successfully unassigned at the supervisor terminal the message "Device released" is printed at the supervisor terminal. The message "Device MTPdn unassigned" is printed if the UNASSIGN command was successfully given by a user.

In user mode, the message "Device released" signals a successful UNASSIGN command issued from the supervisor terminal.

Operator Use of Tapes

The operator will need to use tapes when performing system backup and restore operations involving tape. The commands to back up and restore partitions to tape are MAGSAV, MAGRST, PHYSAV, and PHYRST. These commands, and the procedures to use them, are discussed in Chapters 7 and 8.

THE FILE TRANSFER SERVICE (FTS)

The File Transfer Service (FTS) allows users to request that files be transferred from one node in the network to another. Like Batch and Spool, FTS accepts requests from users and queues them for execution. If one of the nodes involved in the transfer is not up and running, FTS attempts the transfer later.

FTS Elements

The File Transfer Service consists of the following elements:

- Requests
- Queues
- Sites
- FTS Manager
- FTS Servers

Users submit file transfer requests into request queues. These requests identify sites known to FTS as acceptable nodes to which communication is possible. When communication lines are open and not busy, the FTS Manager chooses a request and passes it to an FTS Server. The FTS server performs the transfer, then reports back to the FTS manager that it has completed the transfer. The request is then removed from the queue.

The FTS server performs the transfer by communicating with another FTS server on a remote node. The two servers pass the file data over the communications path; they also pass status information to ensure the correct transmittal of the file.

FTS Commands

There are three FTS commands available to the operator:

<u>Command</u>	<u>Function</u>
FTR	Submit, display, and control file transfer requests
FTOP	Start up, shut down, display, and control FTS servers and the FTS manager
FTGEN	Configure FTS queues and sites

Users can use only the FTR command. The operator and System Administrator may use all three commands.

Operator Responsibilities for FTS

The operator is responsible for ensuring that the FTS server phantoms are running, monitoring the progress of users' file transfers, and generally ensuring the smooth running of the FTS service.

It is the operator's task to monitor all aspects of the File Transfer Service. Specific tasks are:

- Ensuring that file transfer servers and the file transfer manager are in operation.
- Monitoring user requests.
- Monitoring and archiving FTS system log files.
- Making sure the FTSQ* UFD has enough room to accommodate users' files.

The operator's responsibilities when working with the File Transfer Service are discussed in detail in Chapter 12.

The FTS server performs the transfer by communicating with another FTS server on a remote node. The two servers pass the file data over the communications path; they also pass status information to ensure the correct transmittal of the file.

FTS Commands

There are three FTS commands available to the operator:

<u>Command</u>	<u>Function</u>
FTTR	Submit, display, and control file transfer requests
FTOP	Start up, shut down, display, and control FTS servers and the FTS manager
FTGEN	Configure FTS queues and sites

Users can use only the FTTR command. The operator and System Administrator may use all three commands.

Operator Responsibilities for FTS

The operator is responsible for ensuring that the FTS server phantoms are running, monitoring the progress of users' file transfers, and generally ensuring the smooth running of the FTS service.

It is the operator's task to monitor all aspects of the File Transfer Service. Specific tasks are:

- Ensuring that file transfer servers and the file transfer manager are in operation.
- Monitoring user requests.
- Monitoring and archiving FTS system log files.
- Making sure the FTSQ* UFD has enough room to accommodate users' files.

The operator's responsibilities when working with the File Transfer Service are discussed in detail in Chapter 12.

Periodically, the operator should inspect the FTSQ* UFD. The exact contents of this directory depend upon how your System Administrator has configured the FTS system. The directory contains log files that record the activity of each FTS server. As these files have no limit as to their size, they may grow to such size that the FTSQ* directory no longer has room to hold copies of files being transferred for users. Therefore, log files should be periodically reviewed and archived to offline storage. The names of the log files are set in the queue and site configurations using the FTGEN command.

PRIMENET

This section outlines the operational tasks involved in maintaining your system as a node in a PRIMENET network. Four categories of tasks will be discussed:

- Using the ADDISK and SHUTDN commands.
- Communicating with operators on other systems.
- Monitoring the Network Event Log.

19.3 |

At some installations, system operators also perform network-related tasks that are normally described as System Administrators' duties. (For example, operators may configure the network or maintain network security.) For information on these tasks, as well as descriptions of the various kinds of communications lines that PRIMENET supports, see the chapter on PRIMENET in the System Administrator's Guide. Additional background information on PRIMENET can be found in the Prime User's Guide and the PRIMENET Guide.

Adding and Shutting Down Remote Disks

As a system operator, you are probably responsible for the following tasks:

- Starting up remote disks on your system, using the ADDISK command.
- Shutting down remote disks on your system, using the SHUTDN command.

The ADDISK and SHUTDN command are used for starting up and shutting down both local and remote disks. These two commands can only be used at the supervisor terminal. They are described in Chapter 13. The information on ADDISK and SHUTDN in this chapter is of particular importance to operators of networked systems.

In order for a process on your system to access a remote disk, the following must occur:

- The operator on the remote system must use ADDISK to start up the disk on that system.
- You must start the disk on your system, using the ADDISK command.

The order in which these two operations are performed is unimportant. However, a disk that is not started up on its own system cannot be accessed by any other network node.

Understanding RFA: RFA (formerly FAM II) stands for Remote File Access. It allows users to access files and directories residing on disks connected to other nodes in the network, instead of being limited to only the local system disks.

19.3

At Rev. 19.3, PRIMOS supports only RFA. If you are networked to a system running an older version of PRIMOS that uses FAM I, you must upgrade that system to use RFA in order for the two systems to continue communicating.

Adding Remote Disks (ADDISK): ADDISK simply adds a specified partition name residing on a specified network node to the local logical device list. There is no check on the up/down status of the remote node or on the existence of the disk. Thus, you need not wait until a remote node comes up in order to add one of that node's disks.

When you add a disk with ADDISK, the partition name appears in the STATUS DISKS list on your system. Because there is no check on the up/down status of the remote system or on the existence of the disk, you cannot assume that every disk on the STATUS DISKS list has been started up and is accessible at this time. You must attempt to attach to a disk if you wish to confirm that the remote link and system are up and that the disk exists.

When a remote disk is added its physical device number does not appear on the STATUS DISKS list.

Remotely added disks acquire the write-protection status assigned them on their local systems.

Shutting Down Remote Disks (SHUTDN): The SHUTDN command is used to remove a remote disk from the local list of logical disks. Use of the SHUTDN command on a remote disk only affects local users of that disk. It does not affect any other users of that disk on the network.

Communicating with Operators on Other Nodes

As a system operator, you may occasionally have to communicate with an operator on another node of your network. For example, you may need to confer on the accessibility of certain disks between your systems. One way to contact other operators is by means of the MESSAGE command. For example, to send a message to the operator on system SYSA, you would issue the following command:

```
OK, MESSAGE -1 NOW -ON SYSA
Hi Marty, what's wrong with your PAYROL disk - can't attach to it!
OK,
```

For an explanation of the MESSAGE command, refer to Chapter 3.

Monitoring the Network Event Log

The system operator may sometimes need to examine the network event log file, NET_LOG.mm/dd/yy. This file contains records of events such as circuit resets, ring errors, and packet sequence errors. To display the contents of NET_LOG.mm/dd/yy, issue the PRINT_NETLOG command. For example:

```
OK, PRINT_NETLOG TTY
PRINT_NETLOG REV 19.2
```

```
PRINT_NETLOG EVENT LOG FOR INPUT FILE <0>PRIMENET*>NET_LOG.04/22/83
23:32:12 MONDAY APRIL 25, 1983
```

```
07:54:24 FRIDAY APRIL 22, 1983
```

```
COLD START - PRIMOS REV REV.19.2
```

```
09:38:24 FRIDAY APRIL 22, 1983
```

```
RING NODE: 134 NOT ACCEPTING XMITTS.
PACKET WACKED
XMIT STAT IS: 020100 (OCT)
```

```
LEVEL III PROTOCOL DOWN - RING NODE: 134
```

```
09:52:52 FRIDAY APRIL 22, 1983
```

```
RING NODE: 2 NOT ACCEPTING XMITTS.
PACKET LOST, RING DOWN
XMIT STAT IS: 100100 (OCT)
```

The STATUS NET Command: The STATUS NET command displays the up/down status of all defined remote sites in the network. HDX sites may be listed as offline as well as up or down. For more information on the STATUS command, see Chapter 3 of this book, the Prime User's Guide, and the PRIMOS Commands Reference Guide.

Communicating with Operators on Other Nodes

As a system operator, you may occasionally have to communicate with an operator on another node of your network. For example, you may need to confer on the accessibility of certain disks between your systems. One way to contact other operators is by means of the MESSAGE command. For example, to send a message to the operator on system SYSA, you would issue the following command:

```
OK, MESSAGE -l NOW -ON SYSA
Hi Marty, what's wrong with your PAYROL disk - can't attach to it!
OK,
```

For an explanation of the MESSAGE command, refer to Chapter 3.

Monitoring the Network Event Log

The system operator may sometimes need to examine the network event log file, NET_LOG.mm/dd/yy. This file contains records of events such as circuit resets, ring errors, and packet sequence errors. To display the contents of NET_LOG.mm/dd/yy, issue the PRINT_NETLOG command. For example:

```
OK, PRINT_NETLOG TTY
PRINT_NETLOG REV 19.2
```

```
PRINT_NETLOG EVENT LOG FOR INPUT FILE <0>PRIMENET*>NET_LOG.04/22/83
23:32:12 MONDAY APRIL 25, 1983
```

```
07:54:24 FRIDAY APRIL 22, 1983
```

```
-----
COLD START - PRIMOS REV REV.19.2
```

```
09:38:24 FRIDAY APRIL 22, 1983
```

```
-----
RING NODE: 134 NOT ACCEPTING XMTS.
PACKET WACKED
XMIT STAT IS: 020100 (OCT)
```

```
LEVEL III PROTOCOL DOWN - RING NODE: 134
```

19.2

09:52:52 FRIDAY APRIL 22, 1983

RING NODE: 2 NOT ACCEPTING XMITs.
PACKET LOST, RING DOWN
XMIT STAT IS: 100100 (OCT)

If you want to send the output to a printer instead of displaying it on your terminal, issue the following command:

19.2

`PRINT_NETLOG -SPOOL -DELETE`

The output file will be created, spooled, and then deleted. To choose an input file other than the most recent event log file, include the `-INPUT` option followed by the pathname of the input file on the command line, as follows:

`PRINT_NETLOG TTY -INPUT PRIMENET*>NET_LOG.04/20/83`

Network event log files are located in the directory `PRIMENET*`.

`PRINT_NETLOG` is fully described in Chapter 13. The event log messages produced by `PRINT_NETLOG` are fully described in Appendix I.

5

Monitoring the System

INTRODUCTION

This chapter focuses on general system monitoring. PRIMOS provides three methods by which systems may be monitored. These methods are:

- Performance measurement
- Event logging
- Supervisor terminal messages

MONITORING SYSTEM PERFORMANCE

The USAGE command prints a meter display of system performance. This tool is especially useful for determining the degree to which individual users and processes are using system resources, and thus affecting system performance.

The USAGE command allows operators and users to monitor several performance factors of PRIMOS's operation. Both manual and automatic sampling modes are available, as are brief and long forms of display. At Revision 19.3, the USAGE command will monitor remote systems, as long as the remote system is also running Revision 19.3.

| 19.3

Automatic Sampling

Automatic sampling allows the user or operator to observe system performance at equal sampling intervals. The USAGE program automatically times each interval, as specified by the user, and displays new information at the end of each interval.

When you use automatic sampling, you must use the -FREQ option to specify the sampling interval. You may also use the -TIMES option to specify how many samples are to be taken. An example of the command to invoke automatic sampling follows:

OK, USAGE -FREQ 1800 -TIMES 10

In this example, the operator asks PRIMOS to monitor the system 10 times, with an interval of 1800 seconds (30 minutes) between each sampling. USAGE will thus run for a period of 5 hours (1800 seconds x 10 samples / 3600 seconds per hour = 5 hours).

Manual Sampling

Manual sampling allows the user or operator to choose each sample interval individually. The intervals do not have to be identical. If you want to use manual sampling, you must not specify either the -FREQ option or the -TIMES option. (You may specify the -BRIEF option, to request a brief form of USAGE's display.)

With manual sampling, USAGE prints a prompt message when you first invoke it. From then on, USAGE takes and displays a sample each time you enter a START command. (There are no options that may be used with the START command.)

After each sample is taken, USAGE pauses and return you to command level, allowing you to enter other commands.

It is recommended that manual sampling times be not less than 30 real seconds.

An example of the command to invoke manual sampling follows:

OK, USAGE -BRIEF
[USAGE 19.3]
Type "START" to continue.

OK, START

19.3

Remote Sampling

Remote sampling allows you to sample a remote node. You can request either the long or brief displays and choose either automatic or manual sampling. The remote system must be configured for RFA and must be a Rev. 19.3 system. An example of the command to invoke remote sampling follows:

19.3

```
OK, USAGE -FREQ 30 -ON GOLD  
[USAGE 19.3]  
Type "Start" to continue.
```

```
OK, START
```

USAGE Data Display

An example of the long form of data display is shown in Figure 5-1. Explanations of symbols follow in the next section.

Note

Processes may accumulate CPU time without actually being logged in. Such processes will be displayed in the USAGE output with a blank space in the UserId column.

USAGE Symbols

You will find the definitions for USAGE's screen displays listed after the sample USAGE session in Figure 5-1, appearing on the following page. The USAGE screen display definitions are divided into three categories:

- system meter displays (appearing in the top portion of the USAGE display)
- user meter displays (appearing after the system meter displays)
- disk meter displays (optionally appearing at the bottom of the USAGE display)

All percentages that appear in the USAGE display are based on elapsed or CPU time in the last sampling interval, unless otherwise stated.

OK, USAGE -ALL

[USAGE 19.3]

Type "START" to continue.

OK, START

Oct 83 12:39:34.72 dTIME= 30.78 CPU= 5.14 I/O= 0.00
 Up since 05 Oct 83 06:11:40 Wednesday CPUTot= 5959.09 I/Otot= 2142.96

%CPU	%Idl1	%Idl2	%Error	%I/O	%Ovlp	IO/S	PF/S
16.68	77.98	0.00	1.68	0.00	0.00	0.00	0.29

%Clock	%FNT	%MPC	%PNC	%SLC	%GPPI	%DSK
0.77	0.00	0.00	0.25	0.00	0.00	0.01

%AMLC	%Async	%Sync	%ICS	Segs	Used	Pages	Used	Wired
0.56	2.06	0.00	0.28	8192	809	4096	4093	343

Locate	%Miss	%Found	%Same	%Share	Loc/S	IM/S
72	0.00	51.39	48.61	0.00	2.34	0.00

Disk	Qwaits	%Qwait	DMAovr	%DMAovr	Hangs	%Hang
0	0	0.00	0	0.00	0	0.00

Usr	UserID	Mem	Wire	Segs	CPUtime	dCPU	%CPU	I/Otime	dI/O	%I/O
1	SYSTEM	1661	326	153	47.547	0.005	0.017	224.668	0.000	0.000
13	SHELOB	6	1	15	35.908	0.041	0.133	12.956	0.000	0.000
15	MARY	365	1	30	206.096	1.117	3.629	43.136	0.000	0.000
23	GOLLUM	181	1	19	324.228	0.096	0.313	74.728	0.000	0.000
34	BILBO_BAGGINS	437	1	35	272.127	3.109	10.099	79.468	0.000	0.000
85	MOS	70	1	8	1.021	0.157	0.509	1.008	0.000	0.000
118	SYSTEM	8	0	3	10.295	0.053	0.173	2.688	0.000	0.000
148	NETMAN	3	1	5	242.474	0.257	0.835	4.128	0.000	0.000
149	RT_SERVER	8	1	8	6.234	0.007	0.023	1.824	0.000	0.000
151	SYSTEM	23	1	10	28.687	0.045	0.146	20.112	0.000	0.000

Disk	Count	%Count	Time	%Util	Total %Count	Total %Util	Avg time (msec)
'26	0	0.00	0.00	0.00	45.34		
0	0	0.00	0.00	0.00	45.34	1.94	16.53
'22	0	0.00	0.00	0.00	7.61		
0	0	0.00	0.00	0.00	7.61	0.32	16.38
'27	0	0.00	0.00	0.00	2.17		
0	0	0.00	0.00	0.00	2.17	0.09	16.56
'23	0	0.00	0.00	0.00	44.88		
0	0	0.00	0.00	0.00	44.88	2.35	20.28

OK,

USAGE Example
Figure 5-1

Definitions of System Meter Displays

Following is a list of system meter screen displays invoked by the USAGE command:

<u>Display</u>	<u>Definition</u>
dTIME	The number of real seconds elapsed between the previous sample time and the current sample time.
Up Since	The time at which the system was last cold started.
CPU	The number of CPU seconds charged to all <u>user</u> processes in the current sampling interval.
I/O	The number of I/O (disk) seconds charged to all <u>user</u> processes in the last sampling interval. ²
CPtot	The number of CPU seconds charged to all user processes since cold start.
I/Otot	The number of I/O (disk) seconds charged to all user processes since cold start.
%CPU	The percentage of real time during which CPU time was charged to <u>user</u> processes. This can be loosely interpreted as the percent of useful utilization of the CPU.
%Idl1	The percentage of idle CPU time. (On a P850, this figure is the percentage of master ISU idle time.) This value can be roughly interpreted as the percent of total CPU time not involved in user processes.
%Idl2	The percentage of idle CPU time for the P850 slave ISU. This number is always zero on non-P850 configurations.
%Error	The percentage of CPU utilization not otherwise accounted for, and presumed taken by interrupts, scheduler overhead, process exchange, and similar operations. This value can be negative if one or more processes have been overcharged with respect to CPU time.

19.3

19.3	%I/O	The average percentage of dTIME of all controllers during which I/O (disk) was charged to user processes. This can be loosely interpreted as the percentage of time disk I/O was in progress.
	%Ovlp	The estimate of the amount of I/O (disk) traffic that has been overlapped with nonidle CPU time during the last sampling interval.
19.3	IO/S	The average I/O (disk) request rate in operations per second, over the last sampling interval.
19.3	PF/S	The average page fault frequency in faults per second, over the last sampling interval.
	%Clock	The percentage of CPU time used by the realtime clock service process.
	%FNT	The percentage of CPU time used by the P850 slave ISU realtime "frontstop" process.
	%MPC	The percentage of CPU time used by the MPC (printer, punch, reader) processes.
	%PNC	The percentage of CPU time used by the PRIMENET Node Controller process.
	%SLC	The percentage of CPU time used by the SMLC process.
	%GPPI	The percentage of CPU time used by the GPPI (general purpose controller) processes.
	%DSK	The percentage of CPU time used by disk driver processes.
	%AML	The percentage of CPU time used by the AMLC process.
	%Async	The percentage of CPU time used to service asynchronous lines connected to an ICS controller.
	%Sync	The percentage of CPU time used to service synchronous lines connected to an ICS controller.
	%ICS	The total percentage of CPU time used to service the ICS controllers on the system.
	Segs	The number of segments present at the end of the last sampling interval.

Used	The total number of segments present at the end of the last sampling interval.
Pages	The number of pages present at the end of the last sampling interval.
Used	The total number of pages used at the end of the last sampling interval.
Wired	The number of wired pages during the last sampling interval.
Locate	The total number of calls made in the last sampling interval to the file system associative buffer manager, LOCATE.
%Miss	The percentage of calls to LOCATE in the last sampling interval that resulted in a disk read being performed (this is the percentage of LOCATE misses).
%Found	During the last sampling interval, the percentage of calls to LOCATE that found the desired record already in the associative buffers.
%Same	The percentage of calls to LOCATE in the last sampling interval to access the same record the process had just located.
%Share	The percentage of calls to LOCATE in the last sampling interval for a record that was already in use by another process.
Loc/S	The LOCATE use rate in calls per second, over the last sampling interval.
LM/S	The LOCATE miss rate, in misses (disk reads) per second, over the last sampling interval.
Disk	The total number of disk I/O operations performed in the last sampling interval.
Qwaits	During the last sampling interval, the number of times that a process had to wait to get a disk request block allocated.
%Qwait	The percentage of disk I/O requests during the last sampling interval that required waiting for a disk request block.
DMAovr	The number of disk operations during the last sampling interval that resulted in DMA overrun errors.

19.3

%DMAovr	The percentage of disk operations during the last sampling interval that resulted in DMA overruns.
Hangs	The number of disk operations during the last sampling interval that caused the disk controller to hang and time out.
%Hang	The percentage of disk operations in the last sampling interval that caused controller hangs.

Definitions of User Meter Displays

The following list indicates USAGE'S user meter displays.

<u>Display</u>	<u>Definition</u>
Usr	The user number.
UserID	The login name of the user. If this name is too long to fit in the column, the name will be printed, and the remaining information will be displayed on the following line.
Mem	The total number of physical pages resident in memory (at the time the page control databases were examined) that belong to the user's segments (segment numbers 0 through '3777 are charged to user 1). This value can be taken as a rough estimate of the demand the user is placing on virtual memory management. If the system is paging at a reasonably high rate, this value can also approximate the size of the user's average working set over reasonably short intervals.
Wire	Number of wired pages used by this user at the end of the last sampling interval.
Segs	Number of segments used by this user at the end of the last sampling interval.
CPUtime	The CPU time, in seconds, used by this user since login.
dCPU	The CPU time, in seconds, used by this user during the last sampling interval.
%CPU	The percent of total CPU time used by this user during the last sampling interval.

19.3

I/Otime	The I/O (disk) time, in seconds, used by this user since login.	19.3
dI/O	The I/O (disk) time, in seconds, used by this user during the last sampling interval.	19.3
%I/O	The percent of realtime (over the last sampling interval) during which I/O (disk) was in progress for this user.	19.3

Note

If a user logs in or logs out during a sampling interval, incorrect or even negative meter values may result. Some caution must therefore be used in interpreting the per-user metering data. It is suggested that multiple samples be taken, with at least a 30-second interval between samplings.

Definitions of Disk I/O Displays

The disk I/O displays invoked by USAGE are as follows:

<u>Display</u>	<u>Definition</u>	
Disk	The octal controller I/O address and the disk drive unit number.	
Count	The number of disk I/O operations for that controller or unit in the last sampling interval.	
%Count	The percentage of total disk I/O operations in the last sampling interval performed by that controller or unit.	
Time	The time, in seconds, spent performing I/O operations on the specified controller or unit during the last sampling interval.	
%Util	The percentage of disk I/O time spent performing I/O on that controller or unit during the last sampling interval.	
Total %Count	Percentage of the total count of I/O operations done on the drive since system boot. For the controller, it is the sum of the drives attached.	19.3
Total %Util	Percent of the time each drive has been busy since system cold start.	

19.3

Avg Time (msec) The average time in milliseconds for each I/O operation on the drive since the system was booted.

Interpreting USAGE Reports

USAGE is a diagnostic tool for determining where the problem is when system performance seems lower than normal. There are three major resources in a PRIME system: the CPU, the disks, and memory. Any of these can become a bottleneck. Systems experiencing these bottlenecks are described as CPU-bound, I/O-bound, and memory-bound, respectively. The following paragraphs describe what to look for when your system is having performance problems.

CPU-bound Systems: A system that is overutilizing the CPU may exhibit the following symptoms:

Symptom: %Idl1 (and %Idl2 on an 850) is less than 1.

Solutions:

- Inform your System Administrator, who may wish to upgrade the CPU.
- Use the CHAP command to give interactive users higher priorities and longer time slices. Decreasing the eligibility slice with the ELIGTS command would favor shorter transactions.
- Identify user processes which are using most of the CPU by looking at the %CPU column in the per-user report. Your System Administrator may be able to optimize particular programs and make them more efficient.

Symptom: %Idl1 (and %Idl2 on an 850) is less than 10% and the sum of the system process percentages is high (greater than 20 to 30%)

Solutions:

- Check the %Aml number. If it is consistently greater than 20% then check the following:
 - Make sure the baud rate on the last line of the last AMLC board is not greater than 300 baud.
 - Make sure there are no unterminated lines.
 - Make sure cables, modems, or terminals are not generating spurious characters.

- Have your Customer Service Representative check the controllers. It may be that a controller is causing spurious interrupts or that it is improperly configured.

Symptom: %Idl1 (and %Idl2 on an 850) is greater than 10% and %I/O is less than 60%, yet performance is slow

Solutions:

- Raise the value of MAXSCH (See Chapter 13) to allow more users to execute concurrently.
- Have your Customer Service Representative check to see if system data bases are creating a bottleneck.

I/O-bound and Memory-bound Systems: When a system is memory-bound, it will tend to become I/O-bound.

Symptom: %I/O is over 60% to 70% and %Idl1 (and %Idl2 on an 850) is greater than 10%

If both these things are encountered, your system is I/O-bound. Check the additional symptoms listed below to determine a course of action.

Symptom: PF/S is greater than 10 to 15

Solutions:

- See your System Administrator, who may wish to add more memory.
- Using the MAXSCH command, reduce the number of concurrent processes.
- Add a disk drive and use an alternate paging partition on this drive.
- Edit the configuration file and change the PRATIO (Paging Ratio) directive setting. See the System Administrator's Guide (Rev. 19.1 or beyond) for a discussion of the PRATIO directive.
- Reduce the working set of PRIMOS by making the values of configuration directives such as NUSEG, NIUSR, and AMLBUF as small as possible.
- Check the Mem column of the USAGE per-user report. Your System Administrator may wish to modify and re-order the load sequence of certain applications to reduce the working set.

Symptom: %Miss is higher than 20%

Solutions:

- Edit the configuration file and increase the value of the NLBUF configuration directive. This should only be done if PF/S is not greater than 10 to 15.
- See your System Administrator, who may want to add another disk drive.
- Check the I/O time column of the per-user USAGE report. Your System Administrator may wish to try to reduce the number of I/O requests made by certain applications.

Symptom: The %Util in the disk report is 60% or greater for one drive

Solution:

- Move directories and/or paging partitions to disks which are not heavily accessed.

SYSTEM AND NETWORK EVENT LOGGING

Each Prime computer contains an event logging mechanism that records information about significant system events such as cold and warm starts, machine checks, disk errors, and so on, in an internal buffer. The operator should dump this buffer to a disk file from time to time, and then format and print the disk file. Systems connected to a network use a network event logging mechanism as well as a system event logging mechanism. Both logging mechanisms are controlled at system startup time by the CONFIG directives LOGREC and NETREC, as explained in the System Administrator's Guide. While the system is running, the event loggers are turned on and off by the EVENT_LOG command.

The following section describes the operation of both the system and network event logging mechanisms.

See Appendixes H and I for a list of event logger messages.

The EVENT_LOG Command

The command `EVENT_LOG` is used to turn system or network event logging off or on. It is described in Chapter 13. To turn system event logging on, enter the command:

```
EVENT_LOG -ON
```

To turn network event logging on, enter the command:

```
EVENT_LOG -NET -ON
```

Turning system event logging on causes a file to be opened in the UFD `LOGREC*`. The name of the file is `LOG.mm/dd/yy`, where `mm`, `dd`, and `yy` numerically represent the month, day, and year the command is issued. This file may be specified as the input event file to `LOGPRT`. For network logging, the UFD used is `PRIMENET*`, and the filenames are of the format `NET_LOG.mm/dd/yy`. These two UFDs must be present for event logging to take place.

To turn event logging off, use the `-OFF` option instead of the `-ON` option in the `EVENT_LOG` command.

Files in `LOGREC*` and `PRIMENET*`: A file unit will be opened for user 1 whenever system event logging is turned on. This file cannot be closed through use of the `CLOSE` command. The unit can only be closed by turning system event logging off.

Whenever network event logging is turned on, a file is opened by `NETMAN`. The file can only be closed by turning network event logging off.

Access to `LOGREC*` and `PRIMENET*`: User `SYSTEM` must have ALL rights to UFD `LOGREC*`. User `NETMAN` must have ALL rights to UFD `PRIMENET*`. All other users should have LUR rights to both directories.

Using the System Event Logger: The PRIMOS event logger automatically records major system events (such as warm starts, cold starts, disk startups, and so on) and writes them to a file, named `LOG.mm/dd/yy`, in the UFD `LOGREC*`. The contents of this file are examined with the `PRINT_SYSLOG` command. A detailed description of the event logging system can be found in Chapter 13.

The System Administrator can define the maximum amount of disk space allocated for event logging files according to your system's particular needs, using the `SET_QUOTA` command.

Using the Network Event Logger: The PRIMOS network event logger automatically records major network events (such as operator shutdowns, event buffer overflows, out-of-sequence packets, and so on) and writes them to a file, NET_LOG.mm/dd/yy, in UFD PRIMENET*. The contents of this file are examined with the PRINT_NETLOG command. Details of the network event logging system are in Chapter 13. Also see Chapter 4 for information on PRIMENET and network event logging.

The System Administrator can define the maximum amount of disk space allocated for network event logging files according to your system's particular needs, using the SET_QUOTA command.

SUPERVISOR TERMINAL MESSAGES

Many problems encountered by users, by PRIMOS, and by other parts of the system cause messages to be sent to the supervisor terminal. As the system operator, you are responsible for observing these messages and taking appropriate action. These messages can be categorized as follows:

- VCP or Diagnostic Processor messages (See Volume I of this book.)
- User requests (See Chapter 3.)
- Magnetic tape assignment requests (See Chapter 4.)
- Batch messages (See Chapters 4 and 11 and Appendix G.)
- Spool messages (See Chapters 4 and 10.)
- FTS messages (See Chapters 4 and 12.)
- Disk error messages (See Appendix J.)

If you are unsure about the meaning of a message, contact your System Administrator for assistance.

If You Use a Video Display Unit as a Supervisor Terminal

On some systems, the supervisor terminal is a Video Display Unit (VDU) rather than a hard-copy terminal. Since VDU terminals do not automatically produce a printed copy of operator commands and system messages, the operator should maintain a COMOUTPUT file as a record.

The COMOUTPUT file should be started during system startup. The recommended starting procedure is shown in Volume I of this book. The file should be spooled frequently (probably once a day) and then deleted, in order to keep its size down.

For ease in reading and writing the file, it should be given a read-write lock of UPDT (allowing multiple readers plus one writer). To do this, enter the RWLOCK command (described in the PRIMOS Commands Reference Guide). For example:

```
OK, RWLOCK CMDNC0>SYSTEM.COMO UPDT
OK,
```

With ACL-protected systems, the directory containing the COMO file should allow SYSTEM at least DALURW access. On password-protected systems, the COMO file should be protected RWD RWD (7 7), or the directory in which it resides should have no password.

The COMOUTPUT file will work under PRIMOS only. Operators using VDUs as supervisor terminals are advised to run all operations under PRIMOS.

Informative Messages

Most of the messages appearing at the supervisor terminal are simply to inform you of changes in system status. In general, these messages do not indicate anything requiring your immediate attention. Typical informative messages appearing at the supervisor terminal are:

- user-id (user-number) logged in day, dd month yy hh:mm:ss

A user has logged in.

- user-id (user-number) logged out day, dd month yy hh:mm:ss
Time used: 00h 00m connect, 00m 00s CPU, 00m 00s I/O.

A user has logged out. Time used is printed. Times used are: connect time, CPU time, disk I/O time.

- User user-number: Phantom requested terminal input.

A phantom has requested terminal input and has been logged out.

- Phantom phantom-user-number: error-text

A phantom has encountered an error and has been logged out.

- user-id (user user-number on node-name) at hh:mm
BATCH Executing job job-name for user user-id (#job-id)

The Batch messages that appear at the supervisor terminal are explained in Appendix G. They are easily identified, since they all either begin with *BATCH* or are sent by user BATCH_SERVICE.

- DISK xx ER - other information.

A disk error has occurred under PRIMOS. xx is RD for a read error and WT for a write error. See Appendix J for a full explanation of the other information.

- DISK xx ERROR - other information.

A disk error has occurred under PRIMOS II. xx is RD for a read error and WT for a write error. See Appendix J for a full explanation of the other information.

PART II

Procedures

6

Formatting Disk Devices

INTRODUCTION

This chapter describes the MAKE utility, used for formatting disks and disk partitions. For information on the formatting and labeling of magnetic tapes (the LABEL utility), see the Magnetic Tape User's Guide.

Before a disk can be used by your computer, it must conform to Prime's software addressing method. Bringing a disk into conformance with your system's requirements is called "formatting" or "creating" a disk. At times, your System Administrator may decide to divide a disk's memory area into two or more subdivisions, called partitions. This kind of formatting is sometimes called "partitioning."

MAKE is the system utility for the creation and partitioning of disks. MAKE can format both user and paging disks. User disks are those areas of the disk pack utilized for the actual storage of a user's work. Paging disks are those areas of the disk used for the temporary storage of data by the virtual memory management mechanism of PRIMOS. After a user disk is created, it has the following PRIMOS files and directories written to it:

- The Master File Directory (MFD), the top level of the file system that contains the list of all directories and files stored on the partition. (When the disk is first made, there are only two or three files and two directories in the MFD, as described below.)
- The BOOT file, used when the disk is used to bootstrap the system.

- The Disk Record Availability Table (often referred to as the DSKRAT), containing information about the physical structure of the partition. The DSKRAT file has the same name as the partition.
- If badspots exist, a badspot file (BADSPT), used to indicate the location of any physical defects on the disk.
- An empty directory named CMDNC0.
- An empty directory named DOS.

MAKE may be run from a command file, in either PRIMOS or PRIMOS II.

WHAT TO DO BEFORE RUNNING MAKE

Prior to running the MAKE utility, the operator should take the following steps:

1. Determine the number of the physical disk (or partition of the disk) that is to be formatted. Appendix A explains how to do this. The physical disk number tells system the type of storage device being used, the drive unit on which the disk is mounted, the controller to which the drive unit is connected, and, for partitions, the size of the partition and its location on the disk pack.
2. If you are working under PRIMOS, take the following steps:
 - A. Add the device to the table of assignable disks with the DISKS command. This must be done at the supervisor terminal. The format for this command is:

DISKS physical-device-number
 - B. Assign to your terminal the disk to be formatted. To do this, use the command ASSIGN DISK with the physical device number. The format for this command is:

ASSIGN DISK physical-device-number
3. If you are working under PRIMOS II, write-protect all running disks except the disk to be created by MAKE, to prevent accidental erasure of data on a disk if a physical device number is mistyped. (Most disk drives have a switch labelled WRITE PROTECT. Push this switch.) The DISKS and ASSIGN DISKS commands are not given when MAKE is run under PRIMOS II; thus, there is no overlap protection.

RUNNING MAKE

MAKE is invoked with a number of options on the command line that specify the details of the disk creation operation. MAKE requires the specification of two options to obtain the following information:

- The physical disk number of the disk to be created (-DISK)
- The name to be given to the newly created disk partition (-PARTITION)

In addition, MAKE allows the following information to be specified via command line options:

- The type of disk being created (-DISK_TYPE)
- The number of records to be used for paging, if the disk is to be split between user and paging storage (-SPLIT)
- That the disk should be formatted (-FORMAT and -NEW_DISK) | 19.3
- The level of verification to be used in checking for badspots (-BADSPOT_LEVEL)
- That an existing badspot file is to be used (-COPY_BADSPOTS_BY_NAME and -COPY_BADSPOTS_BY_DEVICE) | 19.3
- That a list of known badspots will be input (-QUERY_BADSPOTS)
- The baud rate of the supervisor terminal, if the disk will be used for booting a system (-BAUD_RATE)
- That the disk is to be a pre-Rev. 19 format disk (-PRE_REV19)

As you read the next ten items, write down a list of options you need to achieve the desired results.

1. Specify the physical disk number: Use the -DISK option to specify the physical disk on which MAKE is to operate. (See Appendix A for information on determining physical disk numbers.) The -DISK option must be specified on the command line. For example:

OK, MAKE -DISK 21260

If you do not specify the physical disk number following the -DISK keyword, MAKE will prompt you for it.

Caution

Do not forget to include the -DISK keyword preceding the physical disk number on the command line. If the word -DISK is inadvertently omitted, or entered after the physical disk number, unpredictable behavior will occur. Should this happen, perform one of the following two procedures, depending on whether you are running under PRIMOS II or PRIMOS:

- Under PRIMOS II, immediately depress the MASTER CLEAR button on the front panel of the CPU. (On machines with old-style control panels, turn the rotary knob to STOP/STEP.) This is the fastest and most reliable way to stop the CPU. Then, reboot PRIMOS II and run MAKE again, including the -DISK option on the command line in the correct place.
- Under PRIMOS, enter a CONTROL-P to stop the program. If you have only assigned one disk (the one to be formatted) to your terminal, there is little chance that any harm has occurred. Issue the RLS -ALL command, then invoke MAKE again. Include the -DISK option on the command line in the correct place.

2. Name the partition: Use the -PARTITION option to specify the partition name. If you do not specify this option on the command line, or if you specify an illegal partition name, MAKE will prompt you. The partition name may be a maximum of six characters long. The first character must not be a digit or dash (-); the name can only contain alphabetic characters, digits, and the special characters _ # \$ & * - . /. For example:

OK, MAKE -DISK 21260 -PARTITION ZAPHOD

The partition name will serve as the name of both the partition and the file containing the disk record availability table (DSKRAT). Make sure each partition has its own unique partition name.

This name may be changed later by using the -RENAME option of the ADDISK or SHUTDN command, or by using the CNAME command.

3. Specify the disk type: You may specify the type of storage device being used, with the `-DISK_TYPE` option. If you do not, MAKE will assume that the disk is an 80MB or 300MB storage module.

Follow the `-DISK_TYPE` option on the command line with one of the keywords from the following list:

<u>Disk Type</u>	<u>Keyword</u>	<u>Comments</u>
80MB or 300MB storage module	SMD	(Default)
32, 64, or 96MB Cartridge Module Device	CMD	
68MB fixed-media disk	68MB	
158MB fixed-media disk	158MB	
160MB fixed-media disk	160MB	
300MB fixed-media disk	MODEL_4475	
675MB fixed-media disk	600MB	(Same as 600MB)
Floppy Disk	FLOPPY	(Same as Diskette)

| 19.3

If you do not follow the `-DISK_TYPE` option with the disk keyword, or if you specify an illegal disk type, MAKE will print a list of legal disk types and prompt you for the correct one. An example of using the `-DISK_TYPE` option follows:

OK, MAKE -DISK 21260 -PARTITION ZAPHOD -DISK_TYPE SMD

4. Split the disk: If you are creating a partition that is to be used for paging, specify the `-SPLIT` option on the command line. MAKE will print the total number of available records, and prompt you for the number of records to be used for paging. Both of these numbers are decimal. For example:

OK, MAKE -DISK 20061 -PARTITION PAGING -SPLIT
 **** MAKE <Rev. 19.3>

Total decimal number of records available: 7407
 Decimal records for paging? 7400

| 19.3

If you know how many records are to be used for paging and you do not want to be prompted for this information, specify the number of records to be used for paging following the `-SPLIT` option on the command line.

For example:

OK, MAKE -DISK 20061 -PARTITION PAGING -SPLIT 7400

If you do not specify the `-SPLIT` option on the command line, MAKE does not set aside any room on the disk for paging.

Note

It is recommended that the user portion of a split disk be used only for containing information on badspots on the disk. Only 8 to 32 records need to be set aside for this information. Using the user portion of a split disk for file storage results in a performance penalty, particularly when the paging portion of the disk is heavily used.

19.3 |

5. Format the disk: If the disk being used has never been used on a Prime system before, include the `-FORMAT` and `-NEW_DISK` options on the command line. (These options will be ignored if the disk being created is a floppy disk.) This tells MAKE to initialize the data patterns on the disk.

6. Specify the level of verification: MAKE can perform from 1 to 4 levels of verification on a newly created disk. This provides an automatic check for badspots on the disk.

Normally, one level of badspot checking is performed for storage module disks (SMD) and Cartridge Module Devices (CMD), and four levels of badspot checking are performed for fixed-media disks (FMD). Verification is not performed for floppy disks. To override these defaults, or to specify that no verification is to take place, include the `-BADSPOT_LEVEL` option on the command line, followed by a number from 0 to 4. This number represents the level of verification to be performed by MAKE. The 0 means no verification, 4 means four levels of verification (the most thorough). For example:

OK, MAKE -DISK 1060 -PARTITION IONIA -BADSPOT_LEVEL 4

If you include the `-BADSPOT_LEVEL` option on the command line, but you do not include the number representing the level of verification, MAKE will prompt you for it.

If you do not include the `-BADSPOT_LEVEL` option on the command line, the default level of verification is 1 for storage module disks (SMD) and Cartridge Module Devices (CMD), and 4 for fixed-media devices (FMD).

7. Copy badspot information from another partition: If you have already created a partition on the current disk, and have entered all badspots known on the entire disk, you may copy the badspot information from that partition.

To copy badspot information from a named partition on the current disk pack, use the following format. The partition must have been started up with the `ADDISK` command.

`OK, MAKE -DISK 20063 -PARTITION GOLD -COPY_BADSPOTS_BY_NAME IONIA`

To copy badspot information from a designated physical device, use the format described below. The partition must have been assigned with the `ASSIGN` command.

`OK, MAKE -DISK 20063 -PARTITION GOLD -COPY_BADSPOTS_BY_DEVICE 1060`

19.3

8. Enter known badspots: `MAKE` assembles a list of badspots on the disk as part of its verification procedure. However, you may already be aware of badspots on the disk.

On some disk packs, badspot locations are recorded inside the pack's bottom cover. Look for a list of pairs of numbers representing the track and head numbers of those parts of the disk determined by the disk manufacturer to be probable badspots. Other disks will list head, track, and byte numbers. If a list exists, you must include the `-QUERY_BADSPOTS` option on the command line.

If badspots have been identified on the disk, include the `-QUERY_BADSPOTS` option on the command line. For example:

`OK, MAKE -DISK 1060 -PARTITION IONIA -QUERY_BADSPOTS`

When you invoke `MAKE` with the `-QUERY_BADSPOTS` option, you will be asked to identify the locations of the badspots. The section on Recording Badspots, below, shows you how to do this.

9. Set the baud rate: If the partition being made will be used for booting PRIMOS, `MAKE` is responsible for setting the baud rate of the supervisor terminal when the boot is executed. This is because `MAKE` writes the boot program onto the disk when it creates a disk.

If the baud rate setting of the partition disagrees with the actual baud rate set on the supervisor terminal, you will be unable to use the supervisor terminal after booting from the newly created disk. Normally, the baud rate is 300 bits per second (bps). To set a different baud rate, include the

-BAUD_RATE option on the command line, followed by the desired baud rate in decimal bits per second. Valid baud rates are: 110, 300, 1200, 9600. On a Prime 2250, you might type:

OK, MAKE -DISK 460 -PARTITION BUNNY -BAUD_RATE 9600

If you include the -BAUD_RATE option on the command line, but do not specify the baud rate, MAKE will prompt for a valid one. If you specify an invalid baud rate, MAKE will prompt you for a correct rate. If you do not include the -BAUD_RATE option on the command line, MAKE will set the baud rate to 300 bps.

10. Create a pre-Rev. 19 partition: Normally, MAKE creates a Rev. 19 format badspot file when badspots exist in the partition. If the disk being created is to be run on a pre-Rev. 19 system, include the -PRE_REV19 option on the command line. This will cause MAKE to create a pre-Rev. 19 format badspot file.

Recording Badspots

When you invoke MAKE with the -QUERY_BADSPOTS option, MAKE will prompt you for the locations of badspots on the disk. You can select one of two forms for entering badspots:

<u>Badspot Information</u>	<u>Numerical</u>	<u>Offset From</u>
<u>Format</u>	<u>Radix</u>	
Head, Track, Sector	decimal	beginning of disk
Record number	octal	beginning of partition

In most cases you will probably prefer to use the Head, Track, and Sector format. Here, head 0 corresponds to head 0 of the disk, not necessarily to the first head of the partition being made. This allows input of known badspots anywhere on the disk, not merely in the partition currently being made. Both formats are described below.

Entering Badspots by Record Number: If you choose to enter badspots by record number, answer YES when MAKE prompts:

Enter as record based badspots?

In this case, you enter each record number in octal, as offset from the start of the partition. MAKE will continue prompting for record numbers until you enter a record number of 0 to indicate that you are done.

Entering Badspots by Head, Track, and Sector: Answer NO to the Enter as record based badspots? prompt. MAKE will repeatedly prompt for three values: HEAD, TRACK, and SECTOR. (If the -PRE_REV19 option was included on the command line, MAKE will not prompt for the SECTOR, and the entire track will be considered a badspot.)

Enter the HEAD number from the flaw map when MAKE prompts for it. Enter the CYL number from the flaw map when MAKE prompts for TRACK number. Use the table below to convert the byte offset from the flaw map to the SECTOR number.

<u>Sector</u>	<u>Byte Number</u>
0	0000 - 2239
1	2240 - 4479
2	4480 - 6719
3	6720 - 8959
4	8960 - 11199
5	11200 - 13439
6	13440 - 15679
7	15680 - 17919
8	17920 - 20159

19.3

To terminate the dialog, input 0 after the TRACK=, HEAD=, and SECTOR= prompts. MAKE will then prompt:

Verify data?

If you answer YES, MAKE prints a list of the badspot head, track, and sector numbers at the terminal.

After Known Badspots Have Been Entered and Verified: At this point, MAKE will ask:

Parameters OK?

Check the list of badspot locations printed by MAKE (assuming you requested the list by answering YES to Verify data?). If the listing is correct, type YES. MAKE will include this information in the badspot file, which appears in the MFD with the filename BADSPT. MAKE also initializes the DSKRAT file to ensure that badspot records are not available for file system use.

19.3

A NO answer returns you to the RECORD= question (for record-based badspot mapping) or to the TRACK= question (for track-based badspot mapping).

Keeping Redundant Badspot Files: As a result of a trend towards higher data storage densities, newer varieties of disks tend to be more susceptible to badspots. For example, a 675MB disk might have as many as 1200 badspots. To guard against loss of badspot information,

19.3

Rev. 19.3 MAKE allows you to maintain redundant badspot files for a disk with multiple partitions. This is done by allowing the badspot file on each partition to contain information about all the badspots on the disk, regardless of the partition they are in.

It is advisable to MAKE an entire disk before allowing users to access and store data on any portion of the disk.

First, make a list of the physical device number for each partition. (Typically, a disk is made starting with the partition at head offset 0, but partitions can be made in any order.) Then make the first partition, entering the HEAD, TRACK, and SECTOR values of all badspots on the flaw map.

Next, make the second partition on your list. This time, include the `-COPY_BADSPOTS_BY_DEVICE` option on the command line to copy the badspot information from the first partition. Make the third partition, including `-COPY_BADSPOTS_BY_DEVICE` to copy information from the second partition, and so on, always using the most recently created badspot file.

19.3 MAKE may find additional badspots as it makes the additional partitions. For example, if MAKE finds an additional badspot on the third partition of a five partition disk, the badspot files for partitions 3, 4, and 5 will contain the entry for the new badspot found on the third partition. The badspot files for partitions 1 and 2 will lack information on the new badspot. If MAKE then finds a new badspot on partition 5, only the badspot file on partition 5 will be complete.

Once all partitions on the disk have been made, but before any users have been allowed to access the disk, you can update the badspot files on the earlier partitions. To do this, you will need to remake the earlier partitions using the `-COPY_BADSPOTS_BY_DEVICE` option. To minimize the time needed, use the `-NO_INIT` and `-BADSPOT_LEVEL 0` options. The file system records have already been initialized, and badspots have already been found and recorded for the partitions you are remaking.

For example, to make a 675MB disk on unit 2 of controller 0 as a four partition disk with 10 heads per partition, you would construct a list such as the following:

```
2464
52464
122464
172464
```

You would ASSIGN these four partitions and MAKE them as follows. (Command lines using the abbreviations, rather than complete dialogs, are shown.)

```
MAKE -DISK 2464 -PARTITION BILBO -NEWDSK -FORMAT -QUERY_BADSPOTS -DISK_TYPE 600MB
MAKE -DISK 52464 -PARTITION FRODO -NEWDSK -FORMAT -CPYDEV 2464 -DISK_TYPE 600MB
MAKE -DISK 122464 -PARTITION MERRY -NEWDSK -FORMAT -CPYDEV 52464 -DISK_TYPE 600MB
MAKE -DISK 172464 -PARTITION PIPPIN -NEWDSK -FORMAT -CPYDEV 122464 -DISK_TYPE 600MB
MAKE -DISK 2464 -PARTITION BILBO -NO_INIT -BADSPOT_LEVEL 0 -CPYDEV 172464 -DISK_TYPE 600MB
MAKE -DISK 52464 -PARTITION FRODO -NO_INIT -BADSPOT_LEVEL 0 -CPYDEV 172464 -DISK_TYPE 600MB
MAKE -DISK 122464 -PARTITION MERRY -NO_INIT -BADSPOT_LEVEL 0 -CPYDEV 172464 -DISK_TYPE 600MB
```

WARNING

Although the `-NO_INIT` option prevents initialization of file system records, it does not prevent initialization of the file system itself. All file system objects on the partition will be effectively deleted even if `-NO_INIT` is specified. Use of `-NO_INIT` merely reduces the amount of time needed to remake a particular partition. Never use `MAKE` on a partition if it contains the only copy of any data that you want to keep.

19.3

Invoking MAKE

After you determine the options you wish to specify, invoke `MAKE` as follows:

MAKE options

The disk you specify via the `-DISK` option will be created as directed by the other options you specify on the command line.

When MAKE Finishes

`MAKE` signals its completion by printing:

```
Done checking for badspots.
    nn lost records.
Disk created
```

19.3

`MAKE` then returns you to the operating system prompt (OK, if in PRIMOS; OK: if in PRIMOS II). `nn` is the number of records recorded in the badspot file (and therefore "lost" to badspots). If no badspots exist, the "lost records" message is not printed.

SPECIAL MESSAGES FROM MAKE

These messages indicate that MAKE is unable to create the disk according to the options specified on the command line or other input.

- Cannot add records <=16 to bad spot file. MAKE

A badspot has been found on records 0-16. Records 0 to 16 contain the bootstrap file, badspot file, DSKRAT, and MFD, and cannot contain any badspots. A disk with flaws in these locations is not usable by PRIMOS. MAKE aborts.

This error may have occurred because the disk has not been formatted. If you did not specify -FORMAT on the command line, reenter the command line and include the -FORMAT option.

- Cannot handle badspots on records <=16. MAKE

You entered a record number (or track/head combination that represented a record number) that was less than or equal to 16. Records 0 to 16 contain the bootstrap and badspot file and cannot contain any badspots. A disk with flaws in these locations is unusable by PRIMOS. MAKE aborts.

- DISK WT ERROR device # PRIMOS record # status-word

A disk write error has occurred. The offending record is added to the badspot file, and MAKE prints a message documenting this fact. Status 177776 indicates a DISK-NOT-READY status has been detected. The software will wait for the disk to become ready, then retry the write.

This error may have occurred because the disk has not been formatted. If you did not specify -FORMAT on the command line, reenter the command line and include the -FORMAT option.

Another possibility is that the disk is write-protected. If this is the case, set the disk drive to permit writing, and issue the MAKE command again.

- Invalid physical device number

An invalid physical device number was entered. The request will be repeated. See Appendix A for a complete guide to the construction of physical device numbers (including partitions).

- Invalid, retype last entry

An invalid track/head/sector number has been entered while entering badspots.

19.3

- nnnnn lost records

One or more badspots affected the current partition or platter. The number of lost records is printed.

19.3

- Read error, record = record-number

A read attempt was unsuccessful. MAKE prints the location of the badspot and modifies the BADSPT file and DSKRAT appropriately.

- Unable to get badspots from device: nnnn
Continuing with make. (MAKE)

MAKE was unable to read the badspot file on the specified partition. You probably did not specify -NEW_DISK for an unformatted disk. MAKE will continue and will create a badspot file if necessary.

19.3

- Unrecovered error

The write was not successful after 10 tries. MAKE has aborted and returned you to the operating system.

This error may have occurred because the disk has not been formatted. If you did not specify -FORMAT on the command line, reenter the command line and include the -FORMAT option.

- Write error, record = record-number

19.3

A write attempt was unsuccessful. MAKE prints the location of the badspot and modifies the BADSPT file and DSKRAT appropriately.

WHAT TO DO AFTER RUNNING MAKE

After running MAKE, you should set the ACL or owner password for the disk. First, if you are running under PRIMOS, unassign the disk with the UNASSIGN DISK command, and remove the disk from the table of assignable disks with the DISKS NOT command:

```
UNASSIGN DISK physical-device-number
DISKS NOT physical-device-number
```

At this point, if you are running PRIMOS II, start up PRIMOS. You may then add the disk to the operating system as follows:

ADDISK physical-device-number

The physical device number to be used in these commands is the one constructed for the disk just created.

Now, use the SET_ACCESS command to convert the partition to an ACL partition, or use the PASSWD command to set an owner password other than XXXXXX. To do this, you attach to the MFD of the newly created disk as follows:

ATTACH <name>MFD XXXXXX

where name is the name of the partition. Then, issue the SET_ACCESS MFD or PASSWD command. (If you use the PASSWD command, be sure to set the non-owner password to XXXXXX, or AVAIL will not be able to size the disk.)

If PRIMOS is to be bootstrapped from this disk, use COPY or FUTIL to copy the UFD DOS from a master disk to UFD DOS on the newly created disk. The BOOT file expects the *DOS64 file to be in UFD DOS in order to bootstrap PRIMOS using the newly created disk pack. If the disk is to be used only as a user partition, it is not necessary to copy the UFD DOS.

EXAMPLES

The first example formats a new disk. It uses the -FORMAT and -NEW_DISK options:

```
OK, DISKS 1062
OK, ASSIGN DISK 1062
OK, MAKE -DISK 1062 -PARTITION GOLD -FORMAT -NEW_DISK
**** MAKE <Rev. 19.3>
```

```
Making      4 head partition GOLD
Partition size in decimal records:      29628
Beginning format.
Format completed.
```

Checking for bad spots.

Done checking for bad spots.
Disk created.

```
OK, UNASSIGN DISK 20061
OK, DISKS NOT 20061
OK,
```

19.3

The next example of the MAKE command uses the -QUERY_BADSPOTS option, illustrating the input of known badspots:

```
OK, DISKS 20063
OK, ASSIGN DISK 20063
OK, MAKE -DISK 20063 -PARTITION GOLD -QUERY_BADSPOTS
**** MAKE <Rev. 19.3>
```

```
Making      1 head partition GOLD
Partition size in decimal records:      7407
Enter badspots by record number? YES
Record numbers are OCTAL.
Use a record number of 0 to end input.
Record = 456
Record = 567
Record = 4332
Record = 44556
Record = 443
Record = 34
Record = 0
Verify data? YES
Record = 00000000456
Record = 00000000567
Record = 00000004332
Record = 00000044556
Record = 00000000443
Record = 00000000034
Parameters OK? OK
```

Checking for bad spots.

Done checking for bad spots.
6 lost records.

Disk created.

```
OK, UNASSIGN DISK 20063
OK, DISKS NOT 20063
OK,
```

The example below illustrates the MAKE command with the -SPLIT and -BADSPOT_LEVEL options, to make a paging partition:

```
OK, DISKS 20063
OK, ASSIGN DISK 20063
OK, MAKE -DISK 20063 -PARTITION PAGING -SPLIT -BADSPOT_LEVEL 4
**** MAKE <Rev. 19.3>
```

```
Total decimal number of records available:      7407
Decimal records for paging? 7400
```

```
Making      1 head partition PAGING
```

19.3

File system records: 7
Paging records: 7400

Checking for bad spots.

Done checking for bad spots.

6 lost records.

Disk created.

OK, UNASSIGN DISK 20063

OK, DISKS NOT 20063

OK,

19.3

The following is an example of the MAKE command using the -COPY_BADSPOTS_BY_NAME option to copy the badspots from another partition of the same disk.

OK, DISKS 10462

OK, ASSIGN DISK 10462

OK, MAKE -DISK 10462 -PARTITION HEART -COPY_BADSPOTS_BY_NAME GOLD -NEW_DISK

**** MAKE <Rev. 19.3.x1>

Making 2 head partition HEART

Partition size in decimal records: 14814

Checking for bad spots.

Done checking for bad spots.

0 lost records.

Disk created.

OK,

7

Backing Up

INTRODUCTION

Backing up is a procedure for making copies of current data files and programs and maintaining those copies in offline storage. These copies can then be restored, in part or in full, if the files on the system are lost or broken.

There are two kinds of backups, full and incremental. A full backup is one in which all specified files are copied onto offline storage. An incremental backup is one in which only files modified since the last backup are copied to offline storage.

Backup procedures may be installed in Command Procedure Language (CPL) files or in command files invoked by the COMINPUT command.

Note

CPL files offer many advantages over COMINPUT files; they may not, however, be used in PRIMOS II. Information on CPL is available in the CPL User's Guide.

This chapter explains:

- How to prepare the system for backups.
- How to use COPY_DISK to perform disk-to-disk backups.

- How to use MAGSAV and PHYSAV to perform disk-to-tape backups.
- How to use MAGRST and PHYRST to verify disk-to-tape backups.

SYSTEM PREPARATION PROCEDURES

A backup must be an accurate copy of a disk partition. Therefore, data on the partition must not be changed during the backup procedure. During restore procedures, the partition must be changed only by the restoration. If backups and restorations are attempted while a user is referencing a partition, inconsistencies may result, and data may be lost. Procedures are provided to protect existing data.

System preparation procedures provide a protected environment in which sensitive maintenance procedures can be performed without interference by users on the system. This section describes the system preparation procedures used when performing any of the following four system maintenance activities:

- Backing up data, described in this chapter
- Verifying backup tapes, also described in this chapter
- Restoring data from disks or tapes, described in Chapter 8
- Repairing disk partitions, described in Chapter 9

The procedures for preparing a system for maintenance activities are illustrated in the following figures in Chapter 7:

<u>Figure</u>	<u>System Preparation Procedures</u>
7-1	System shutdown. Used when backing up or restoring important system disks.
7-2	Partition shutdown. Used when backing up or restoring disks that are not required for normal system operation.

Procedures for backing up disk partitions and verifying backup tapes are shown in the figures listed below. These procedures are performed as part of one of the system preparation procedures described above.

<u>Figure</u>	<u>Backup and Verification Procedures</u>
7-4	COPY_DISK for backup under PRIMOS
7-6	COPY_DISK for backup under PRIMOS II

7-7	MAGSAV for backup under PRIMOS
7-8	MAGRST for verification under PRIMOS
7-9	PHYSAV for backup under PRIMOS
7-10	PHYRST for verification under PRIMOS

Procedures for restoring data to disk partitions are illustrated in Chapter 8. These procedures are also performed as part of one of the system preparation procedures described above.

<u>Figure</u>	<u>Restoration Procedures</u>
8-2	COPY for partial restore under PRIMOS
8-3	COPY_DISK for partition restore under PRIMOS
8-4	COPY_DISK for partition restore under PRIMOS II
8-5	MAGRST for partition restore under PRIMOS
8-6	MAGRST for partial restore under PRIMOS
8-7	PHYRST for partition restore under PRIMOS
8-8	COPY_DISK for emergency restore of command disk
8-9	MAGRST for emergency restore of command disk

The procedure for repairing disk partitions by running FIX_DISK is illustrated in Figure 9-1 in Chapter 9.

Choosing the System Preparation Procedure

Two procedures can be used to prepare a system for maintenance activities:

- System shutdown
- Partition shutdown

System shutdown renders the system unavailable to users during the procedure, but allows all possible operations to be performed. Partition shutdown allows users to continue using the system, as it only involves shutting down carefully selected partitions. However, many installations will be unable to use this method, because of the limitations it places on the maintenance activities.

To determine which procedure is appropriate for a particular session, you must have both a list of all of the operations to be performed during the session and a list of your system configuration. The first list should include, for each operation to be performed:

- The program to be run (COPY, COPY_DISK, MAGSAV, MAGRST, PHYSAV, PHYRST, or FIX_DISK).
- The partition (or partitions) affected.

The second list should include:

- The names and physical device numbers of every disk partition on the system, including the disk controller and drive unit numbers on which the partitions reside.
- The name and physical device number of the command device (COMDEV).
- The physical device number of the paging device (PAGDEV).
- The physical device number of the alternate paging device, if any (ALTDEV).
- Information on which disk partitions contain origin directories for users.
- Information on which disk partitions contain information that is crucial for normal system operation.

Your System Administrator will provide you with this information.

Use these two lists, together with Table 7-1, to determine whether you need to use PRIMOS or PRIMOS II. Remember that partitions affected by a COPY_DISK operation include not only the source and target partitions, but also all the partitions residing on a disk pack that must be removed so that the pack containing the source or target partition may be inserted.

If any of the desired operations must be performed under PRIMOS II, use the system shutdown procedure to prepare the system. If all of the operations can be performed under PRIMOS, it may be possible to use the partition shutdown procedure. Even so, use the system shutdown procedure if the list of partitions affected during the session includes partitions on which many origin directories reside, or on which information crucial for typical use of the system is stored.

You will notice that some combinations (such as backing up the command device using COPY_DISK) require PRIMOS II, while others (such as running FIX_DISK on the command device) require PRIMOS. If a session includes two seemingly conflicting requirements, use the system shutdown procedure. The system shutdown procedure provides for running certain programs under PRIMOS and others under PRIMOS II. It will guide you in performing such complex operations.

Table 7-1
Operating System Restrictions on Backups

Disk	Operation					
	COPY_DISK	MAGSAV	MAGRST	PHYSAV	PHYRST	FIX_DISK
COMDEV	PRIMOS II	PRIMOS or PRIMOS II	PRIMOS or PRIMOS II	n/a (note 2)	n/a (note 2)	PRIMOS
PAGDEV, ALTDEV (split)	PRIMOS II (note 4)	PRIMOS or PRIMOS II	PRIMOS or PRIMOS II	n/a (note 3)	n/a (note 3)	n/a (note 3)
Other Disks	PRIMOS or PRIMOS II	PRIMOS or PRIMOS II	PRIMOS or PRIMOS II	PRIMOS	PRIMOS or PRIMOS II	PRIMOS

Notes

1. When using COPY_DISK, you must make a backup disk pack available. If the disk drive used to hold the backup disk pack normally holds the command disk (COMDEV) or one of the paging disks (PAGDEV or ALTDEV), then PRIMOS II must be used to perform the backup.
2. Because PHYSAV cannot be run under PRIMOS II, and requires the disk to be assigned, it cannot be used to back up the command disk (COMDEV). Therefore, PHYRST cannot be used to restore the command disk.
3. A disk that is being used for paging cannot be assigned. Therefore, PHYSAV and FIX_DISK cannot be used on a paging disk, even a split one, because they cannot be run under PRIMOS II. PHYRST cannot be used to restore such a disk, since PHYSAV cannot save it in the first place.
4. Because COPY_DISK will copy the entire disk, including both the file system and paging portions of a split disk, COPY_DISK is not an appropriate method for backing up a split paging disk. Should it be used for this purpose, badspot handling must be turned off (-NO_BADS), since FIX_DISK cannot properly prepare the disk.

Certain combinations are not available. If such a combination seems necessary, try to find an acceptable alternate operation. For example, you could use MAGSAV instead of PHYSAV on the command device. If no alternate operation is available, skip the operation. Inform your System Administrator of the operation that could not be performed and the result. For example:

- Unable to PHYSAV partition IM, a split disk being used for paging. Used MAGSAV instead.
- Unable to FIX_DISK partition IONIA, a split disk being used as an alternate paging device. No repair operation performed on IONIA.

Detailed descriptions of system preparation procedures follow.

System Shutdown

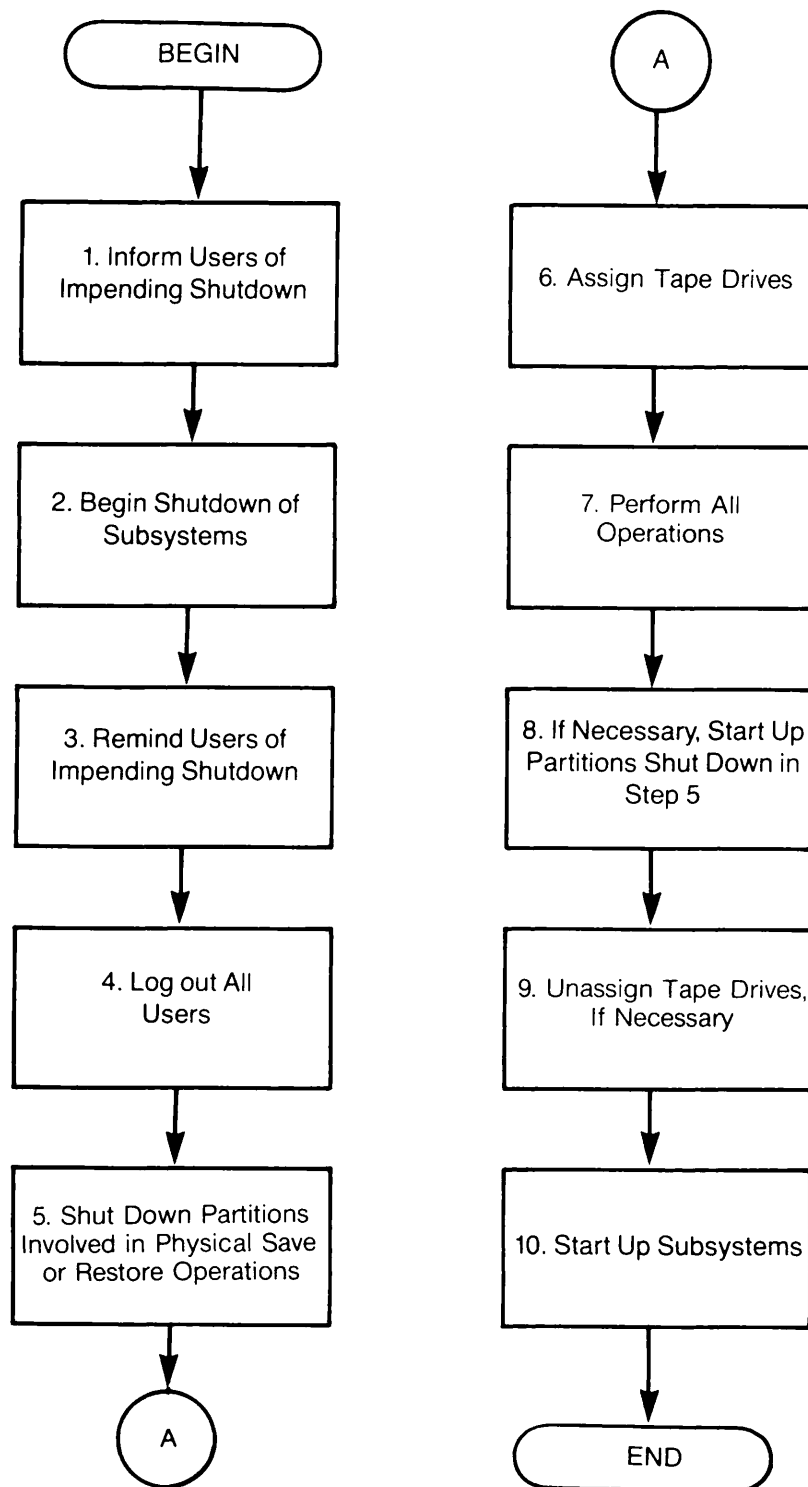
In most cases, the system must be shut down to perform activities such as backing up and restoring. Shutting down the system involves logging out all users, shutting down all subsystems (such as Batch, spoolers, FTS), and making the system unavailable for normal use for a period of time. Although it is the most drastic of the system preparation procedures, it provides an environment in which all of the system maintenance procedures may be performed at top speed. No other users are on the system during this time. For this reason, you may choose to shut down the system even when not required by the desired system maintenance procedure.

The system shutdown procedure is illustrated in Figure 7-1. It must be performed at the supervisor terminal. The steps, in detail, are:

1. Inform users of the impending shutdown. This is best handled with a series of messages; the earliest should provide users with plenty of advance warning. For example:

```
OK, MESSAGE
SYSTEM COMING DOWN TONIGHT AT 11:00
.
.
.
OK, MESSAGE -FORCE
SYSTEM COMING DOWN IN ONE HOUR -- PLEASE LOGOUT BY 11:00
OK, MESSAGE ALL -NOW
SYSTEM COMING DOWN IN ONE HOUR (11:00)
```

2. Prevent new users from logging in, and begin shutdown of subsystems. This should be done at least 10 minutes before the system is shut down. Have subsystems such as Batch, spool, and FTS continue any work in progress, but not begin new work.



System Shutdown Procedure
Figure 7-1

For example:

```
OK, MAXUSR 0 /* Keep Batch from starting new jobs, and
/* keep new users from logging in.
OK, PROP PR0 -STOP /* Shutdown all Spooler phantoms.
[PROP rev 19.2]
```

Wait... Acknowledged.

```
OK, FTOP -STOP_SRVR FTP /* Shutdown all FTS servers.
[FTOP rev 1.0]
```

Server notified to stop.

You should receive more messages from the spoolers and the FTS servers as they recognize your request to discontinue processing. If you wish all subsystem activity to cease before shutting down the system, then proceed with Step 3 only when all of the following conditions are true:

- A BATCH -STATUS command shows no executing jobs.
- All of the spooler phantoms log themselves out (as indicated by messages sent to the supervisor terminal).
- All of the FTS servers log themselves out (indicated by messages sent to the supervisor terminal).

3. Remind users of impending shutdown. For example:

```
OK, MESSAGE ALL -NOW -FORCE
SYSTEM COMING DOWN IN FIVE MINUTES -- PLEASE LOGOUT
.
.
.
OK, MESSAGE ALL -NOW -FORCE
SYSTEM COMING DOWN IN ONE MINUTE -- FORCED LOGOUT COMING!
```

It is particularly effective to include bells (CONTROL-G) in the messages. One bell per message is quite sufficient.

4. Log out all users. Do this at least one minute after performing Step 3. Issue the command:

```
LOGOUT ALL
```

This will log out all users including the Batch monitor. Any spooler phantoms or FTS servers that are still running will be logged out. Do not perform Step 5 until a STATUS USERS command shows that all users have logged out except user 1 (SYSTEM) and user NETMAN (if networks are enabled). This may take up to two minutes.

5. Shut down appropriate partitions. If you will be using COPY_DISK, PHYSAV, PHYRST, or FIX_DISK, shut down the disk partitions that are to be saved, restored, or repaired by these operations, and add them to the assignable disks table. For example:

```
OK, SHUTDN 21060
OK, DISKS 21060
```

Do not shut down partitions corresponding to the backup pack for a COPY_DISK operation; these will be shut down separately as part of the COPY_DISK procedure.

6. Assign necessary tape drives now. For example:

```
OK, ASSIGN MTO
```

7. Perform all desired system maintenance procedures.
8. If necessary, start up partitions shut down in Step 5. Normally, Step 7 involves a coldstart of PRIMOS, which adds all disks to the system automatically. If this is not the case, and you shut down any disk partitions in Step 5, add them now. For example, to add the disks shut down in the example shown in Step 5, you would issue the commands:

```
OK, DISKS NOT 21060
OK, ADDISK 21060
```

9. Unassign tape drives, if necessary. As with Step 8, this step is necessary only if Step 7 did not involve a coldstart of PRIMOS. If it did not, and you assigned any tape drives in Step 6, unassign them now. For example:

```
OK, UNASSIGN MTO
```

10. Start up subsystems, if necessary. If PRIMOS was not started up during Step 7, then start up the Batch, spool, and FTS subsystems, as appropriate. For example:

```
OK, BATCH -START
OK, PROP PRO -START
OK, FTOP -START_MNGR
OK, FTOP -START_SRVR FTP
```

For more information on subsystem startup, see Volume I of this book.

Partition Shutdown

In certain cases, system maintenance procedures do not require the system itself to be shut down. Instead, only the affected disk partitions are shut down. This provides more continuous system availability than the system shutdown procedure, but it also constrains the available maintenance activities. In addition, allowing users to continue using the system may slow down the maintenance activities. (This may be remedied by specifying a high priority with the CHAP command for the user who will be performing these activities. See Chapter 13 for further information.)

To determine whether this procedure is appropriate for your needs, make a list of all the disk partitions that will be affected during the maintenance activities. Affected disks include not only those that are to be backed up, restored, or repaired, but also those that must be physically removed in order to mount backup disk packs for COPY_DISK.

Then, check each affected disk partition to see if any of the following conditions make it necessary to use system shutdown procedures:

- The partition is the command disk (COMDEV), paging disk (PAGDEV), or alternate paging disk (ALTDEV), and must be removed to mount a backup pack.
- The partition is COMDEV, PAGDEV, or ALTDEV, and an operation other than MAGSAV or MAGRST is being performed on it. COPY_DISK, FIX_DISK, PHYSAV, and PHYRST cannot be run on COMDEV, PAGDEV, or ALTDEV while users are on the system.
- The partition contains a significant number of origin directories — more than 20% of the user community, for example. (When such a partition is shut down for maintenance activities, users with origin directories on that partition will be unable to log in. If they are already logged in, they may find it difficult or impossible to continue using the system without having access to their origin directories, and, presumably, to their files.)
- The partition contains information that is used by most of the users on the system.

If any of the above conditions are true for any of the affected disk partitions, use instead the system shutdown procedure described earlier in this chapter.

Due to the first two conditions in the above list, only certain procedures may be used when the partition shutdown procedure is in progress. They are identified by figure number and title.

<u>Procedure</u>	<u>Illustrated in Figure</u>
COPY_DISK for backup under PRIMOS	7-4
MAGSAV for backup under PRIMOS	7-7
MAGRST for verification under PRIMOS	7-8
PHYSAV for backup under PRIMOS	7-9
PHYRST for verification under PRIMOS	7-10
COPY for partial restore under PRIMOS	8-2
COPY_DISK for partition restore under PRIMOS	8-3
MAGRST for partition restore under PRIMOS	8-5
MAGRST for partial restore under PRIMOS	8-6
PHYRST for partition restore under PRIMOS	8-7
FIX_DISK under PRIMOS	9-1

The partition shutdown procedure is illustrated in Figure 7-2. It must be performed at the supervisor terminal, except where otherwise noted. The steps, in detail, are:

1. Inform users of the impending shutdown. This is best handled with a series of messages, the earliest of which will provide users with plenty of advanced warning. For example:

OK, MESSAGE
PARTITIONS <BEEBLE>, <IONIA>, AND <MAGMA> COMING DOWN TONIGHT AT 11:00

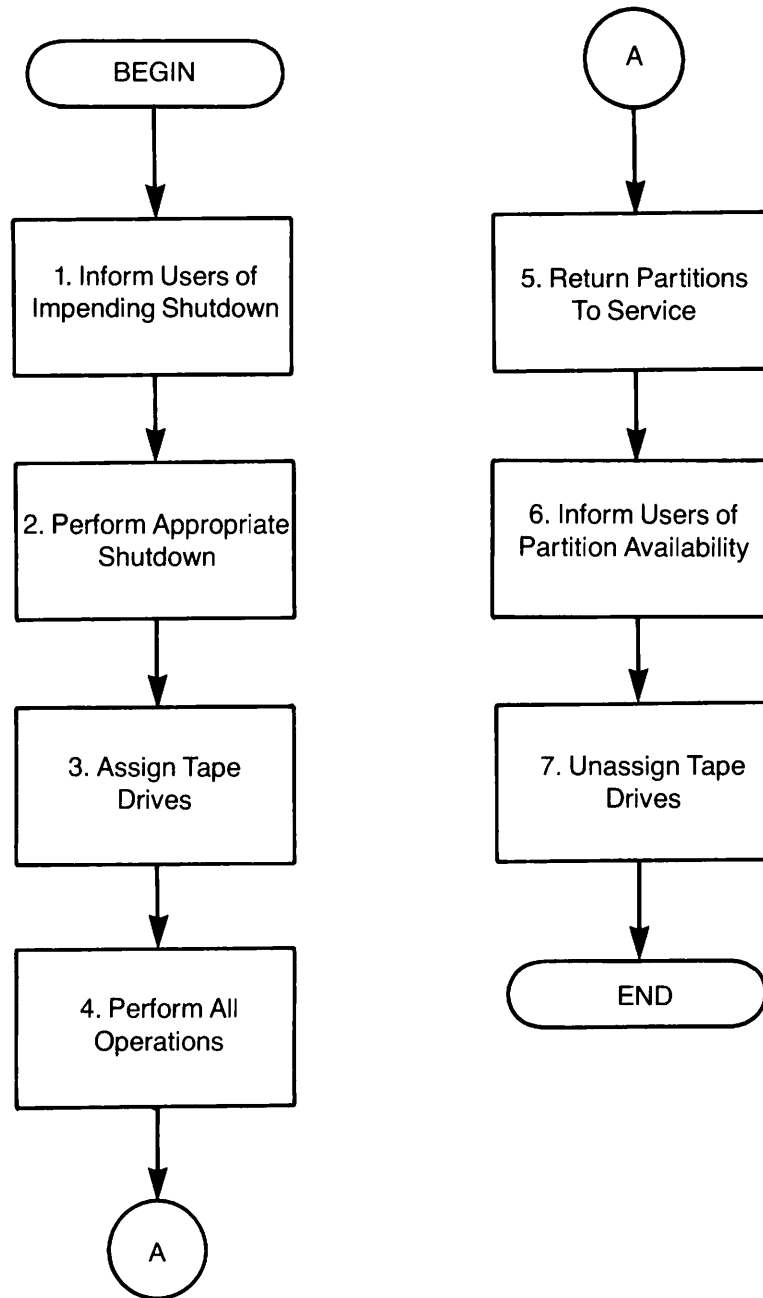
.
.
.

OK, MESSAGE -NOW
PARTITIONS <BEEBLE>, <IONIA>, AND <MAGMA> COMING DOWN IN ONE HOUR

.
.
.

OK, MESSAGE ALL -NOW -FORCE
PARTITIONS <BEEBLE>, <IONIA>, AND <MAGMA> COMING DOWN IN FIVE MINUTES

.
.
.



Partition Shutdown Procedure
Figure 7-2

OK, MESSAGE ALL -NOW -FORCE
PARTITIONS <BEEBLE>, <IONIA>, AND <MAGMA> COMING DOWN IN ONE MINUTE

·
·
·

OK, MESSAGE
PARTITIONS <BEEBLE>, <IONIA>, AND <MAGMA> UNAVAILABLE UNTIL 12:30 A.M.

2. Perform the appropriate shutdown for each disk, as follows:

Disk being saved via MAGSAV:

SET_PRIORITY_ACCESS diskname SYSTEM:PLUR \$REST:NONE

Disk being restored via MAGRST or COPY:

SET_PRIORITY_ACCESS diskname SYSTEM:ALL \$REST:NONE

Other:

SHUTDN pdev
DISKS pdev

diskname is the name of the disk (example: BEEBLE). pdev is the physical disk number (example: 21060).

3. Assign the necessary tape drives now, at the user terminal at which you wish to work. For example:

OK, ASSIGN MTO

4. Perform all operations. You may only perform the procedures listed at the beginning of this section. No procedure involving PRIMOS II may be performed. You may perform these procedures at a user terminal or at the supervisor terminal.
5. Return the disk partitions to service, at the supervisor terminal, as follows:

Disk saved via MAGSAV:

REMOVE_PRIORITY_ACCESS diskname

Disk restored via MAGRST or COPY:

REMOVE_PRIORITY_ACCESS diskname

Other:

DISKS NOT pdev
ADDISK pdev

6. Inform users of partition availability. Cancel the broadcast message and immediately notify all users that the disk partitions are back. The supervisor terminal must be used. For example:

```
OK, MESSAGE
(CR)
OK, MESSAGE ALL -NOW
PARTITIONS <BEEBLE>, <IONIA>, AND <MAGMA> ARE NOW AVAILABLE
```

7. Unassign tape drives. Unassign the drives you assigned in Step 3. For example:

```
OK, UNASSIGN MT0
```

Tape Verification

Tapes can be verified while the system is used for other purposes, as long as at least one extra tape drive is available. This is because verification does not involve use of the disk that was backed up. There are three approaches to verifying tapes:

1. Verify tapes on a system different from the one on which the backups are being performed. As each tape is made during the backup procedure, mount it on a different system and use MAGRST or PHYRST to verify it, while continuing the backup procedure on the next tape. This is the best approach, because verification and backup can proceed concurrently if more than one reel of tape is involved. The amount of time you must spend on the whole process is reduced and the performance of the system being backed up is not affected. However, you need another system, with an appropriate available tape drive, that is not being backed up.
2. Verify tapes on the same system on which the backups are being performed, during the backup procedure. As each tape is made, you mount it on a different tape drive and use another terminal to verify it with MAGRST or PHYRST. As with method 1, you continue the backup procedure on the next tape during the verification of each tape. This will have an impact on the performance of the backup procedure, because the same system is now performing backup and verify operations simultaneously. It still makes good use of your time, because two tape drives, rather than just one, are in use while you wait. However, this does require the availability of a second tape drive and another (nearby) terminal.

3. Verify tapes on the same system on which the backups are being performed, after the backup procedure. After the backup procedure is complete, the system may be brought back up and made available to users once again. You can then verify the backup tapes while other users are using the system.

DISK-TO-DISK BACKUP (COPY_DISK)

COPY_DISK is an external command that copies data from one physical device (the source disk or partition) to another (the target disk or partition). FIX_DISK should be run on the source disk before COPY_DISK is used to ensure file system integrity. After COPY_DISK is used, FIX_DISK should be run on the target disk for badspot handling if COPY_DISK so indicates. Instructions for running FIX_DISK are included in the two COPY_DISK procedures described below.

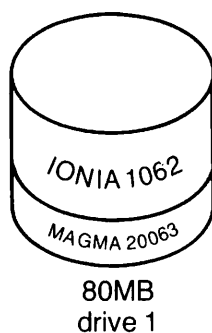
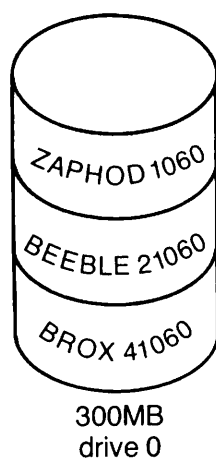
The Procedures for Running COPY_DISK

Disk-to-disk backups are performed between two separate disk drives, or between the removable and non-removable portions of a CMD. When a backup is performed between two disk drives, the disk pack containing the source partition remains in place, and the target disk pack is inserted into a second disk drive, temporarily replacing the disk pack that normally resides there. When the backup is performed between the removable and non-removable portions of a CMD, the non-removable portion is usually being backed up. One or more extra platters are inserted as the removable portion, serving as the target disk.

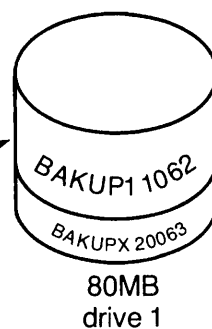
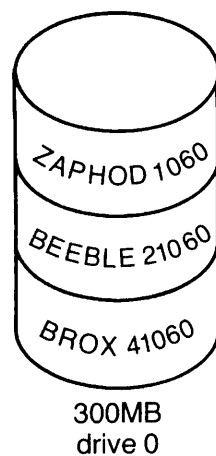
All of the partitions that reside on the disk pack that is normally mounted on the second disk drive must be shut down before the disk pack is removed.

Figure 7-3 shows the three major steps performed during a disk-to-disk backup. In this example, partition BEEBLE, residing on a 300MB pack, is to be copied to an 80MB pack mounted on the second disk drive. First, the BEEBLE partition is shut down. Next, the partitions residing on the second disk drive, IONIA and MAGMA, are shut down. The 80MB pack in the second disk drive is then replaced with a backup pack. Partition BEEBLE is copied to physical disk 1062; although labeled BAKUPl, the partition name will also be BEEBLE after the copy operation. Finally, the disk pack containing the IONIA and MAGMA partitions is replaced, and the BEEBLE, IONIA, and MAGMA partitions are started up again.

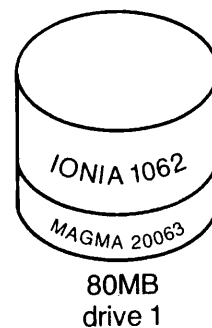
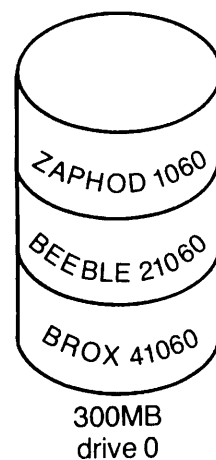
The procedure for running COPY_DISK depends upon whether it is to be run under PRIMOS or PRIMOS II.



Normal system operation.
(Paging disk not shown.)
Partitions BEEBLE, IONIA,
and MAGMA are shut down,
to back up partition
BEEBLE.



The IONIA pack is replaced
with a backup pack.
Partition BEEBLE is then
copied to it.



After the operation is
complete, the IONIA pack
is re-mounted. Partitions
BEEBLE, IONIA, and MAGMA
are started up again.
System operation is
restored.

Example of Disk-to-disk Backup Procedure Under PRIMOS
Figure 7-3

Running COPY_DISK Under PRIMOS: This procedure is summarized in Figure 7-4. It uses the example shown in Figure 7-3 for clarification. It assumes that you have already shut down the source partition involved (BEEBLE) and added it to the assignable disks table. If you have not done this, follow the procedures outlined earlier in this chapter. This procedure should be performed at a user terminal, except where otherwise specified.

1. Assign the source partition at the user terminal. Example:

OK, ASSIGN DISK 21060

2. Run FIX_DISK on the source partition; include the -FIX option. Example:

OK, FIX_DISK -DISK 21060 -FIX -LEVEL 0

See Chapter 9 for a discussion of FIX_DISK.

3. At the supervisor terminal, shut down all of the disk partitions that reside on the disk pack that you will be replacing with the backup pack. In the example, this means the IONIA and MAGMA partitions are shut down, as follows:

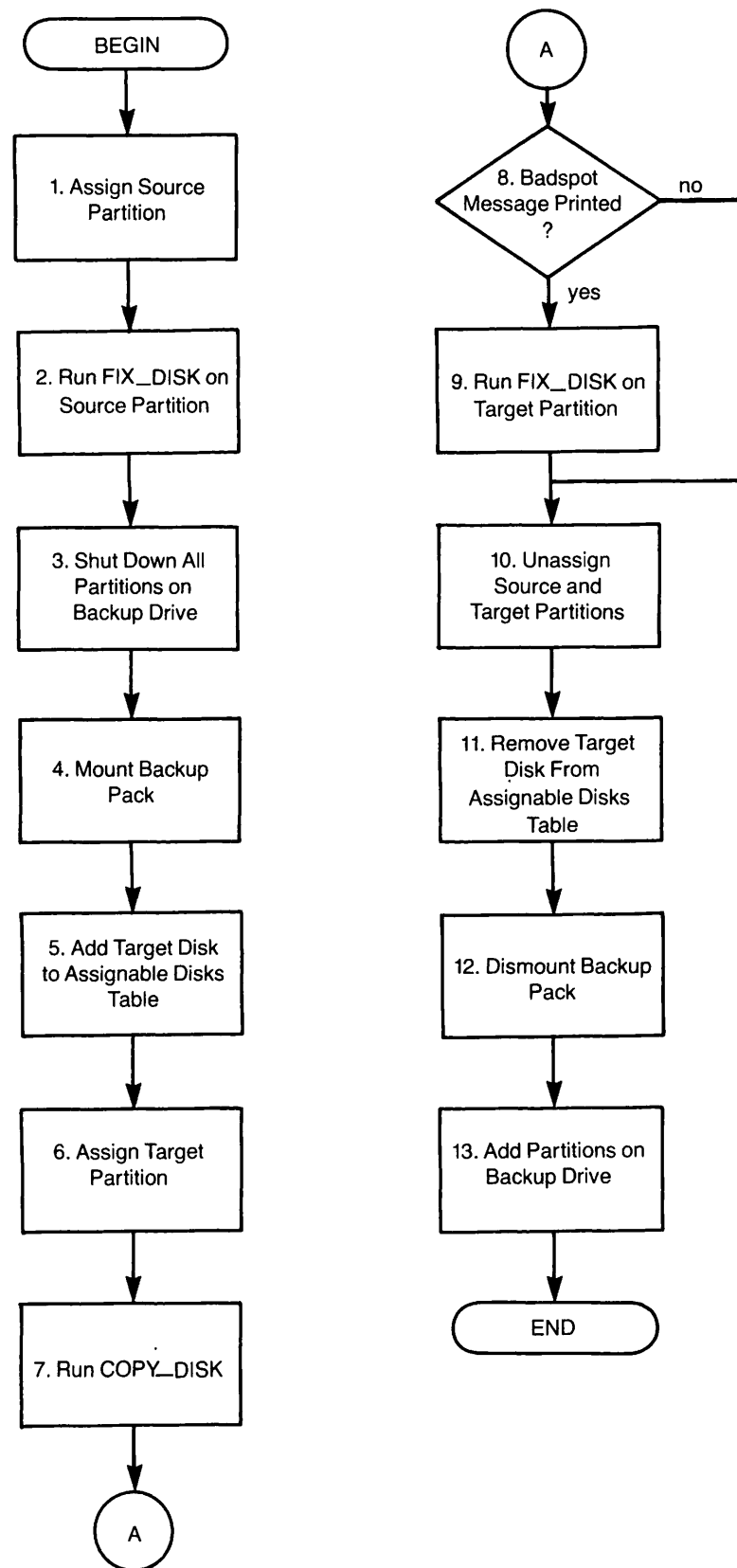
OK, SHUTDN 1062
OK, SHUTDN 20063

4. Remove the pack from the disk drive to be used as the target drive, and replace it with the backup pack. In the example, this means the pack containing the IONIA and MAGMA partitions is removed, and replaced with the pack labeled BACKUP1.
5. Add the target partition to the assignable disks table, at the supervisor terminal. In the example, this is done with the command:

DISKS 1062

Notes

That the physical disk number in this example, 1062, is identical to one of the numbers shown in the example for Step 3, is purely coincidental. In this example, IONIA happens to be a four-head partition, as is the disk being backed up (BEEBLE), and therefore corresponds to the partitioning of the backup pack (BACKUP1).



Procedure for Running COPY_DISK Under PRIMOS
Figure 7-4

Steps 3 through 5 may be done even as Step 2 proceeds, since FIX_DISK is running on the first disk drive at a user terminal, and you are replacing the disk pack on the second disk drive and entering commands at the supervisor terminal.

6. Assign the target partition at the user terminal. Example:

OK, ASSIGN DISK 1062

7. Run COPY_DISK, as described below. Example:

OK, COPY_DISK
COPY_DISK 19.2

 FROM PHYS DISK=21060
 600MB STORAGE MOD? NO
 TO PHYS DISK=1062
 40MB STORAGE MOD? NO
 600 MB STORAGE MOD? NO
 FROM, TO, RECORDS = 021060, 001062, 29628
 PARAMETERS OK?YES

8. COPY_DISK may print the message:

BADSPOTS HANDLED ON PARTITION pdev - PLEASE RUN FIX_DISK

If it does not print the message before returning to PRIMOS, go to Step 10. If it does, continue with Step 9.

9. Run FIX_DISK on the target partition (pdev); include the -FIX option. Example:

OK, FIX_DISK -DISK 1062 -FIX -LEVEL 0

10. Unassign the source and target partitions. Example:

OK, UNASSIGN DISK 21060
 OK, UNASSIGN DISK 1062

11. Remove the target partition from the assignable disks table, using the supervisor terminal. Example:

OK, DISKS NOT 1062

12. Remove the backup pack mounted in the target disk drive, and replace it with the original pack. In the example, this means the pack labeled BAKUP1 is removed, and replaced with the pack containing the IONIA and MAGMA partitions.

Caution

Step 12 must be performed after Step 9 is completed, if it was necessary to perform Step 9.

13. Add the partitions now mounted on the backup drive, using the supervisor terminal. In this example, this means the IONIA and MAGMA partitions:

OK, ADDISK 1062
OK, ADDISK 20063

Running COPY_DISK Under PRIMOS II: This procedure is summarized in Figure 7-6. It uses the example shown in Figure 7-5 for clarification. In this illustration, the command device, ZAPHOD, will be copied to the backup disk. This procedure assumes that you have already logged out all users. If you have not done this, follow the procedures outlined earlier in this chapter. You must perform all of the steps in this procedure at the supervisor terminal.

1. If the source partition is the command device (logical disk 0, or COMDEV), go to Step 5. Otherwise, continue to Step 2. In the example, ZAPHOD is the command device, but Steps 2 through 4 will include examples as if it weren't.

2. Shut down the source partition. Example:

OK, SHUTDN 1060

3. Add the source partition to the assignable disks table. Example:

OK, DISKS 1060

4. Assign the source partition. Example:

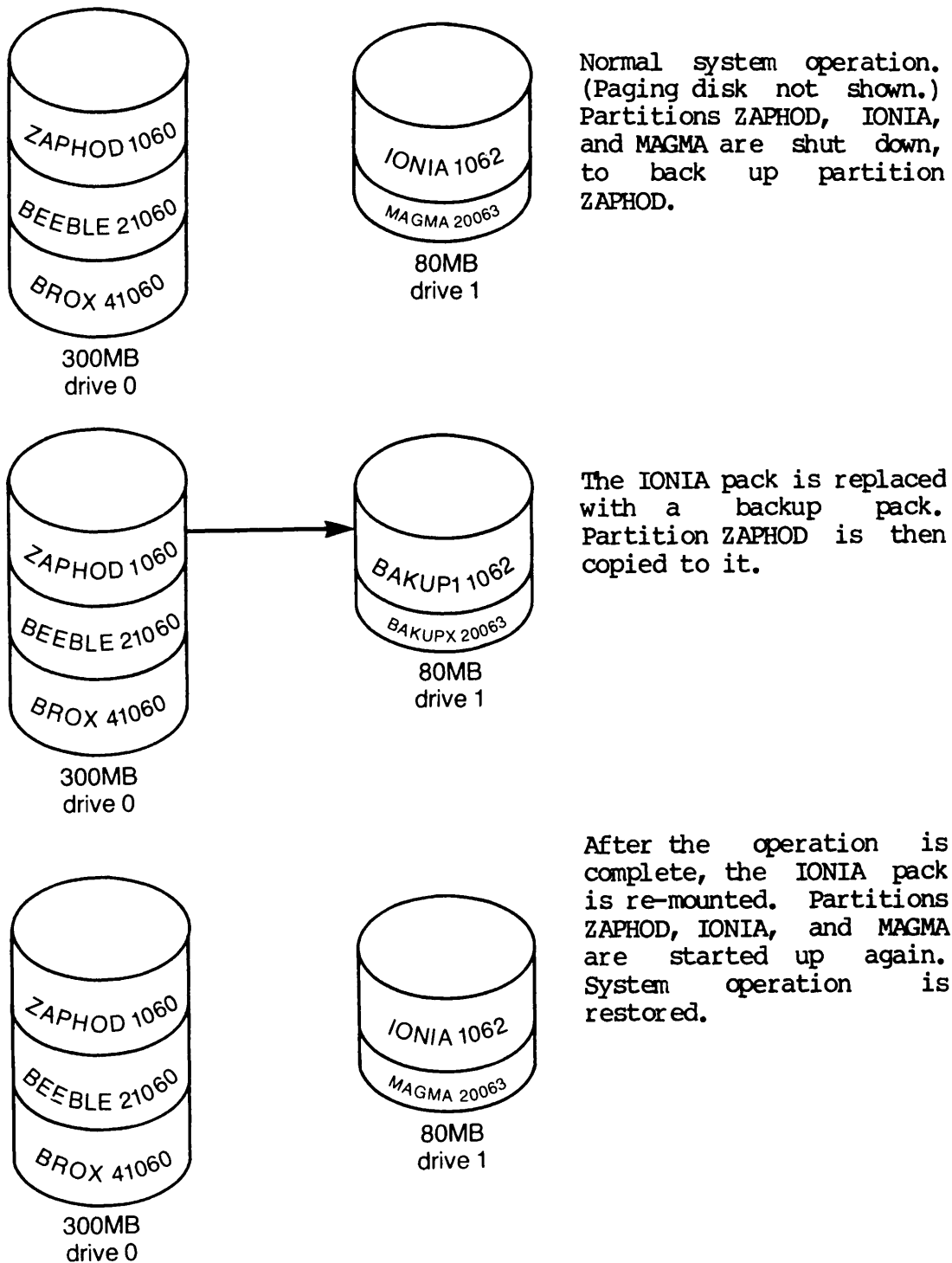
OK, ASSIGN DISK 1060

5. Run FIX_DISK on the source partition. Include the -FIX option. Include the -COMMAND_DEVICE option if the source partition is the command device. Example:

OK, FIX_DISK -DISK 1060 -FIX -COMMAND_DEVICE

6. Shut down PRIMOS. Example:

OK, SHUTDN ALL
REALLY? YES
WAIT,
PRIMOS NOT IN OPERATION



Example of Disk-to-disk Backup Procedure Under PRIMOS II
Figure 7-5

7. Boot in PRIMOS II. See Volume I of this book for information of how to boot PRIMOS II on your system. See Chapter 14 for information on PRIMOS II. If you accidentally boot PRIMOS, simply go to Step 6 again.
8. Remove the pack mounted in the disk drive to be used as the target drive, and replace it with the backup pack. In the example, this means the pack containing the IONIA and MAGMA partitions is removed and replaced with the pack labeled BAKUP1.
9. Run COPY_DISK, as described below. Example:

```
OK: COPY_DISK
COPY_DISK 19.2

FROM PHYS DISK=1060
40MB STORAGE MOD? NO
600MB STORAGE MOD? NO
TO PHYS DISK=1062
40MB STORAGE MOD? NO
600MB STORAGE MOD? NO
FROM, TO, RECORDS = 001060, 001062,      29628
PARAMETERS OK?YES
```

10. COPY_DISK may print the message:

```
BADSPOTS HANDLED ON PARTITION pdev - PLEASE RUN FIX_DISK
```

If it does not print the message before returning to PRIMOS II, go to Step 17. If it does, continue with Step 11.

11. Start up PRIMOS, but do not allow users to log in (do not issue the MAXUSR command). Example:

```
OK: PRIMOS
```

Note

The startup sequence for your system will probably try to add the target partition (disk 1062 in this example). It will fail to do this, because the disk has the same name as the source partition -- it is a copy of the source partition. Do not be alarmed if an error message occurs.

However, if the target partition was successfully added, you must shut it down before proceeding. Example:

```
OK, SHUTDN 1062
```


12. Add the target partition to the assignable disks table.
Example:

OK, DISKS 1062

13. Assign the target partition. Example:

OK, ASSIGN DISK 1062

14. Run FIX_DISK on the target partition; use the -FIX option.
Example:

OK, FIX_DISK -DISK 1062 -FIX

15. Shut down PRIMOS. This is done at this point because the cold start of PRIMOS just performed did not start up all of the disks that are normally present on the system. One of the disk drives contained a backup pack instead. Example:

OK, SHUTDN ALL
REALLY? YES
WAIT,
PRIMOS NOT IN OPERATION

16. Boot in PRIMOS II.

17. Remove the pack mounted in the target disk drive, and replace it with the original pack. In this example, this means the pack labeled BAKUP1 is removed, and replaced with the pack containing the IONIA and MAGMA partitions.

18. Start up PRIMOS.

Running COPY_DISK

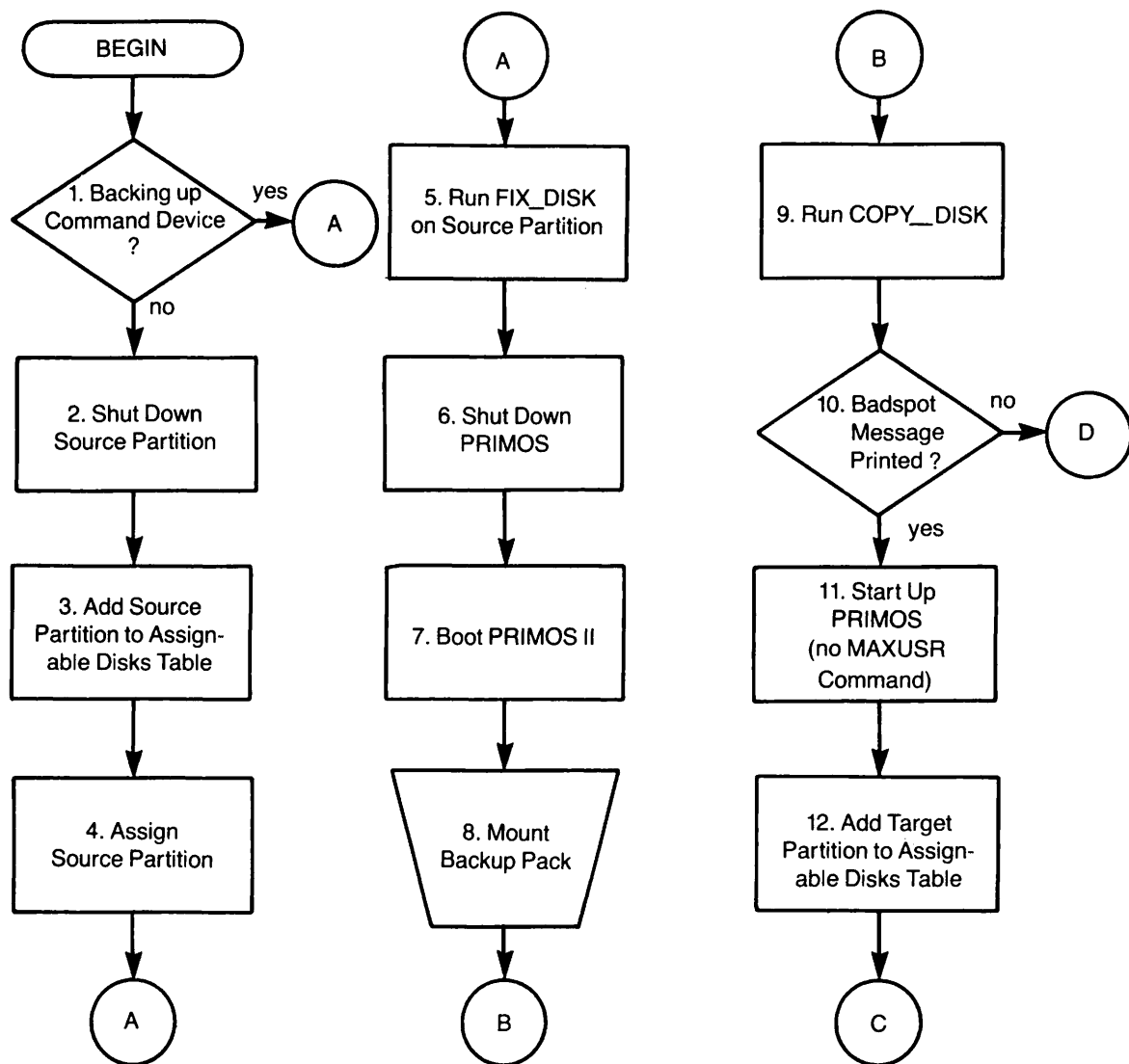
COPY_DISK is invoked by the command format:

COPY_DISK [options]

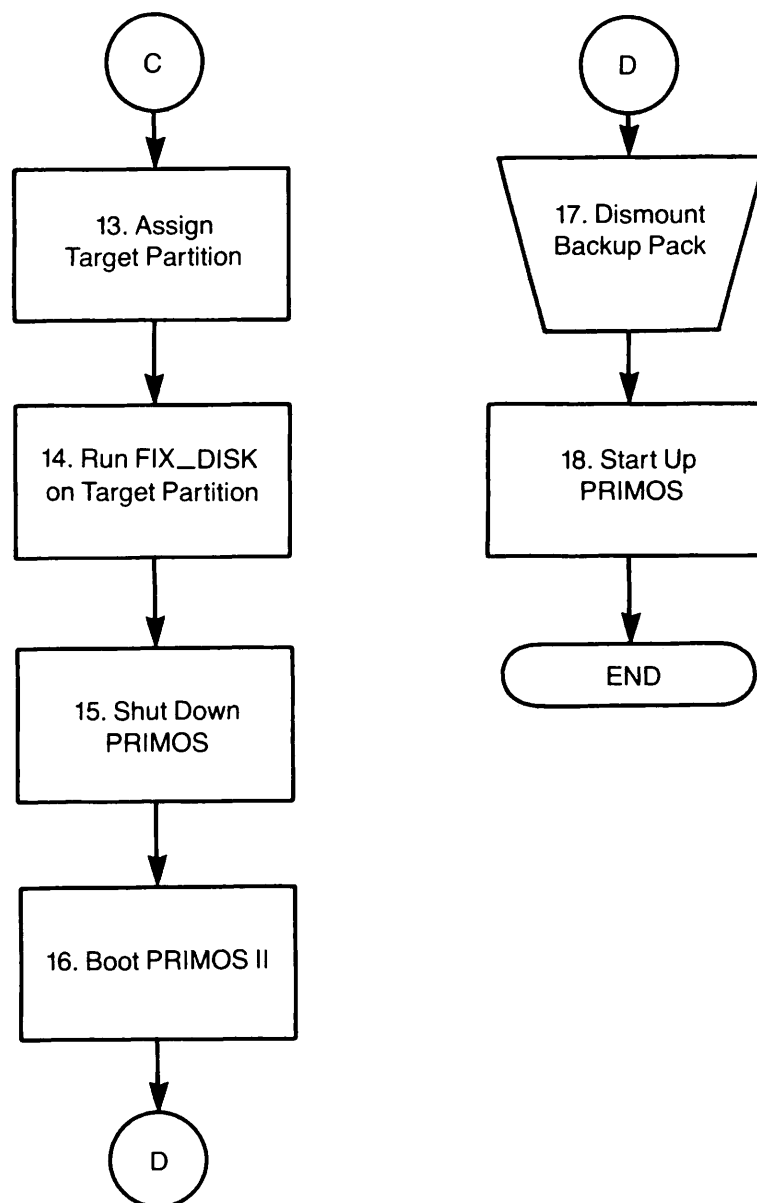
The options specify variations from the standard disk-to-disk copy operation. As you read the next two items, write down the options that you wish to specify, if any.

1. Specify if verification should occur: Normally, COPY_DISK will not perform a verification of the disk-to-disk copy operation. If you wish it to perform this verification, include the -DO_VERIFY option on the command line, as follows:

OK, COPY_DISK -DO_VERIFY



Procedure for Running COPY_DISK Under PRIMOS II
Figure 7-6



Procedure for Running COPY_DISK Under PRIMOS II
Figure 7-6 (continued)

After COPY_DISK finishes making the disk-to-disk copy, it will then verify that the contents of the source and target disks are the same. (The verification takes about as much time as the copying does.)

2. When your system is not a Prime 750 or 850: If your CPU is not a Prime 750 or Prime 850, you may wish to include the -LOWEND option on the command line, as follows:

OK, COPY_DISK -LOWEND

This will speed up COPY_DISK's operation. However, if you use the -LOWEND option on a Prime 750 or 850, it will slow down the operation.

COPY_DISK has additional options that you may find useful under certain circumstances. See Chapter 13 for more information.

After you invoke COPY_DISK with the desired options, you must respond to several questions asked by COPY_DISK. These questions concern disk partition specification, and have four purposes:

- Identifying the unit to be backed up
- Identifying the backup unit
- Clearing up ambiguities and incorrect input
- Verifying your input

You must first indicate the disk partition to be copied from (FROM), and the disk partition to be copied to (TO). Each disk partition must be specified by its physical device number. (See Appendix A for information on physical disk numbers.)

FROM PHYS DISK=physical_disk_number
TO PHYS DISK= physical_disk_number

If there is any uncertainty as to the type of disk, COPY_DISK will follow your input with questions such as:

40MB STORAGE MOD?

or

600MB STORAGE MOD?

The questions require YES or NO responses.

If the FROM and TO disk partitions are not of equal size, the following message is printed:

REC LENGTH AND NR RECS MUST BE = FOR BOTH DEVS

The FROM prompt is then repeated.

If the disk partitions are of equal size, your input is verified with the message:

```
FROM, TO, RECORDS = pch, pch, number_of_records
PARAMETERS OK?
```

If the physical disk numbers given under FROM or TO are not valid, or if the user replies NO to the query PARAMETERS OK?, COPY_DISK repeats the series of questions and waits for a reply. If the new parameters are acceptable, COPY_DISK initiates the copy operation.

What to Do After Running COPY_DISK

When COPY_DISK finishes, it may print the message:

```
BADSPOTS HANDLED ON PARTITION pdev - PLEASE RUN FIX_DISK
```

If this happens, you should run FIX_DISK on the target disk partition (pdev). Until you do this, you must not try to use this disk partition for any other purpose. FIX_DISK cannot be run under PRIMOS II, so if you have been running COPY_DISK under PRIMOS II, you will have to start up PRIMOS before running FIX_DISK. This procedure is described earlier in this chapter. For information on FIX_DISK, see Chapter 9.

If the above message was not printed, then continue with your backup procedures (as described earlier in this chapter).

Examples

A typical use of COPY_DISK might look like this:

```
OK, COPY_DISK
COPY_DISK 19.00

FROM PHYS DISK=460
40MB STORAGE MOD? NO
600MB STORAGE MOD? NO
TO PHYS DISK= 462
40MB STORAGE MOD? NO
600MB STORAGE MOD? NO
FROM, TO, RECORDS = 000460, 000462,      14814
PARAMETERS OK?OK

OK,
```

In the next example, the backup is done via a COMINPUT file named COPIER that is located in the directory BACKUP.

```

OK, A BACKUP
OK, CO COPIER
OK, /* In this example COPY_DISK is run from a COMINPUT file
OK, /* This COMINPUT file copies MOD0 to a backup pack on MOD1
OK, /* We are copying the partitions SALES and PERSON
OK, /* Copy SALES
OK, COPY_DISK
COPY_DISK 19.00

```

```

FROM PHYS DISK=460
40MB STORAGE MOD? NO
600MB STORAGE MOD? NO
TO PHYS DISK= 462
40MB STORAGE MOD? NO
600MB STORAGE MOD? NO
FROM, TO, RECORDS= 0000460, 000462,      14814
PARAMETERS OK?YES

```

```

OK, * Now Copy PERSON
OK, COPY_DISK
COPY_DISK 19.00

```

```

FROM PHYS DISK=110061
600MB STORAGE MOD? NO
TO PHYS DISK= 110063
600MB STORAGE MOD? NO
FROM, TO, RECORDS= 110061, 110063,      7407
PARAMETERS OK?YES

```

```

OK, * That's it! We're done!
OK, CO TTY

```

As the above example illustrates, COMINPUT files (and CPL files) can save a good deal of repetitive input. The COMINPUT file used in the above example looks like this:

```

/* In this example COPY_DISK is run from a COMINPUT file
/* This COMINPUT file copies MOD0 to a backup pack on MOD1
/* We are copying the partitions SALES and PERSON
/* Copy SALES
COPY_DISK
460
NO
NO
462
NO
NO
YES
* Now Copy PERSON
COPY_DISK
110061
NO
110063

```

```

NO
YES
* That's it! We're done!
CO TTY

```

Special Events During the Copying Procedure

COPY_DISK copies disk records from the FROM disk to the TO disk. If the -DO_VERIFY option was given, COPY_DISK then verifies the copy by reading each record from both disks and performing a word-by-word comparison in memory. On machines without a VCP, COPY_DISK displays the track number it is processing in the data lights on the control panel, bits 2-16, during this process. Bit 1 is off during the copy operation and on during the verify operation.

If any disk read errors occur during the copy, the read is retried nine times. The error results in an error message as follows (in which all numbers are octal):

```
DISK RD ERROR device-number PRIMOS-record-number status
```

If the read operation is not successful after 10 tries, COPY_DISK ignores that record and prints the message:

```
DISK RD ERR, DISK=device-number record-number
ERROR IGNORED, COPY_DISK CONTINUED
```

If any disk write errors occur, COPY_DISK retries nine times. Each error results in an error message of the form:

```
DISK WT ERROR device-number PRIMOS-record-number status
```

If the write operation is not successful after 10 tries, COPY_DISK stops trying, prints the error message UNRECOVERED ERROR, and continues.

If on either read or write a DISK-NOT-READY status is detected, a single disk error message is printed with the status '177776. The software then retries the read or write, waiting for the disk to become ready.

If while verifying the copy a discrepancy is detected, COPY_DISK prints VERIFY ERROR, the record number of the track that contained the error, and the word number within the track that did not compare correctly; verification then continues. (See Appendix J for a complete list of status words, and Appendix E for a listing of COPY_DISK error messages.)

Special Events During Badspot Handling

Following is a discussion of events that will affect badspot handling.

- If a BADSPT file on either disk has bad contents (perhaps from being overwritten or truncated) the message:

BAD BADSPT FILE ON PARTITION pdev - IGNORED

is printed, and COPY_DISK attempts to continue the copying operation as if the BADSPT file did not exist. When badspots on the disk are encountered, disk read errors are produced. (See Appendix E.) The backup pack should be free of badspots.

- If the BADSPT file of a source partition contains an EQUIVALENCE block, then the program will abort with the message:

BADSPT FILE ON PARTITION pdev HAS AN EQUIVALENCE BLOCK
PLEASE RUN FIX_DISK

- Free records are necessary on the target disk so that COPY_DISK can map a route around badspots. If no free records are available, you will receive the message:

WARNING - BADSPOT HANDLING DISCONTINUED ON PARTITION pdev(NXTRAT)
NO FREE RECORDS ON PARTITION pdev
IF YOU DO NOT WISH TO CONTINUE WITHOUT BADSPOT
HANDLING YOU WILL NEED TO RE-MAKE PARTITION pdev
OK TO WRITE IT WITHOUT BADSPOT HANDLING (YES/NO)?

A YES causes the partition to be copied without badspot handling. A NO causes COPY_DISK to terminate. In both cases, the badspot information for the target disk (pdev) will be lost. Therefore, you will have to re-MAKE the partition before it can be used again, to reconstruct the badspot information. MAKE is described in Chapter 6.

- COPY_DISK will not handle badspots on pre-Rev. 19 partitions. When a pre-Rev. 19 partition is encountered, you will receive the message:

WARNING - SOURCE PARTITION IS PRE REV 19
NO BADSPOT HANDLING WILL OCCUR ON PARTITION pdev

Whenever badspot handling occurs, the following message appears:

BADSPOTS HANDLED ON PARTITION pdev

This is an indication that FIX_DISK must be run on the target partition prior to its being used again.

Using COPY_DISK With Cartridge Module Device (CMD)Caution

Badspot handling must not be performed during a multi-platter (64MB or 96MB) CMD disk-to-disk copy operation. Rev. 19.2 COPY_DISK will prevent badspot handling from occurring in this situation. However, if you are using an earlier version of COPY_DISK, you must include the -NO_BADS option on the COPY_DISK command line, as follows:

OK, COPY_DISK -NO_BADS

This tells COPY_DISK to not perform badspot handling. At Rev. 19.2 and beyond, this option is superfluous when copying a multi-platter (64MB or 96MB) CMD. Therefore, it is recommended that you always use this option when copying a multi-platter CMD, regardless of what version of COPY_DISK is being used.

Multi-head partitions on the nonremovable section of a CMD may be backed up by copying to a set of removable CMD cartridges. Partitions are copied one disk surface at a time. Cartridges must be restored in the order in which they were copied. The target partition must be equal in size to the original partition.

COPY_DISK asks if it is time to change the removable cartridge with the prompt:

SURFACE n READY?

If the disk is ready, power down the CMD and insert the next cartridge. To continue the COPY_DISK operation, power up the CMD and type:

YES

If YES is typed before the disk is ready, COPY_DISK aborts with the message:

SWREC NOT READY
ER!

Restart the program at the next surface by typing:

S

It is not necessary to recopy the previous surfaces.

Notes

If part of the CMD is being used for paging, COPY_DISK should be run only under PRIMOS II.

When restoring a partition, if a surface is mounted in the wrong order, COPY_DISK will indicate which surface has been mounted in error and then request the correct surface again.

LOGICAL DISK-TO-TAPE BACKUP (MAGSAV)

MAGSAV is the PRIMOS disk-to-tape backup subsystem. It allows you to copy disk partitions onto 7- or 9-track magnetic tape.

A complete discussion of the capabilities of MAGSAV may be found in the Magnetic Tape User's Guide.

Incremental Backups

By performing one or more incremental backups between occasional full backups, you can keep up-to-date copies of all files with a saving of time and tapes. To do this, use MAGSAV's -UPDT and -INC options, as follows:

1. Start with a full backup, using the -UPDT option. When MAGSAV has finished this save, all files will have their DUMPED bits set to 1, as a signal that they have been saved.
2. As users continue to use the system, they will add new files and modify old ones. Every file that is added or modified has its DUMPED bit set to 0, as a sign that it has not been saved in its current form.
3. Now an incremental backup may be done, using both the -INC and -UPDT options. (You may also use -SAVE_UFD, if you wish.) The -INC option causes MAGSAV to check the DUMPED bit on each file before saving it. The file will be saved only if the DUMPED bit is 0. (For example: If a full save with the -UPDT option were done on Monday, and an incremental save were done on Thursday, only files created or modified between the save on Monday and the save on Thursday would be written onto Thursday's tape.) Once the file is saved on tape, the -UPDT option causes MAGSAV to set its DUMPED bit to 1. Therefore, at the end of this save, all files again have the DUMPED bits of 1.

If the -SAVE_UFD option is not used on an incremental save, only altered files are saved. When restored, these files must be copied into an existing directory having the same name as the directory from which they were backed up.

If `-SAVE_UFD` is used, a skeleton directory structure is saved along with the altered files. This allows the files to be restored either into an existing directory or onto a blank disk.

The Procedure for Running MAGSAV

The procedure for running MAGSAV is summarized in Figure 7-7. The steps should be performed at a user terminal near the tape drives.

1. Attach to the MFD of the partition to be saved. For example:

`ATTACH <ZAPHOD>MFD`

2. Mount a tape. It is recommended that you label the tape before you mount it, with at least the following information:

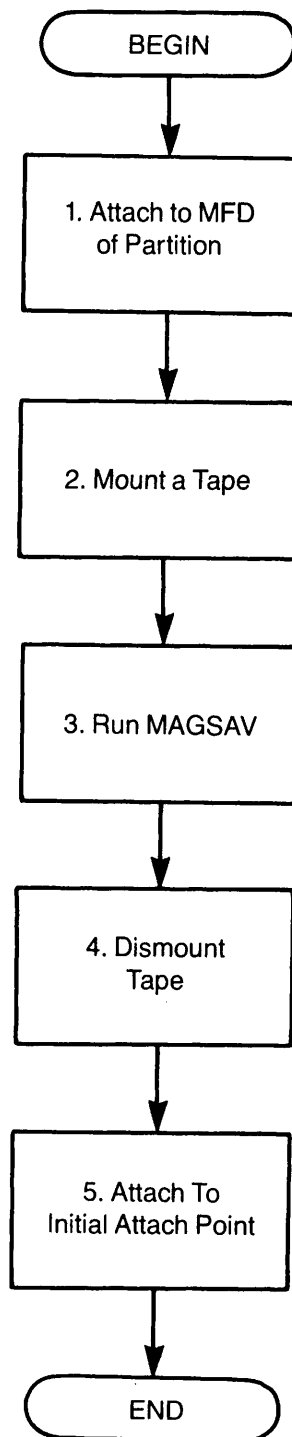
- Your initials
- The date and time
- The name of the partition being saved
- The name of the system
- The recording density (usually 1600 or 6250 bpi).
- The tape number of this set, as in "Tape 1 of ____". You will fill in the blank when you have determined how many tapes were needed to save the partition.
- The fact that MAGSAV is being used to save the tape — include the revision level of PRIMOS.

3. Run MAGSAV, as described below. During this procedure, you may mount and dismount more tapes.
4. When MAGSAV has finished, dismount the last tape. It is very important to fill in the blanks on the tape labels to indicate the number of tapes in the set.
5. Attach back to your origin directory. Enter the command:

`ORIGIN`

Invoking MAGSAV

MAGSAV has a number of options that specify details concerning the backup operation. All of these options are fully described in Chapter 13. However, only some of them are necessary during normal backup



Procedure for Running MAGSAV Under PRIMOS
Figure 7-7

operations. Study the following three items and write down the options that you wish to specify. Then, type:

OK, MAGSAV options

Replace options with the (possibly empty) list of options you wrote down. MAGSAV will then initiate a dialog with you — this will be described later.

1. Performing a full backup: If you are performing a full backup, and you intend to perform an incremental backup on the same partition some time later, specify the -UPDT option on the command line. Example:

OK, MAGSAV -UPDT

If it is not your intention to perform incremental backups later, you do not need to specify this option.

2. Performing an incremental backup: If you are performing an incremental backup, specify the -UPDT, -INC, and -SAVE_UFD options on the command line. Example:

OK, MAGSAV -UPDT -INC -SAVE_UFD

3. For 7-track tape drives: Most tape drives are 9-track. If you are using a 7-track tape drive, specify the -7TRK option on the command line. For example:

OK, MAGSAV -7TRK

The MAGSAV Dialog: After you invoke MAGSAV and specify any options on the command line, the MAGSAV dialog begins. MAGSAV requests information from you in the following order. Appropriate user responses are shown.

<u>Request</u>	<u>Response</u>
Tape unit (9-Trk):	Supply the physical or logical tape unit number (0-7).
Enter logical tape number:	Enter 1.
Tape name:	Supply the name of the partition being backed up.
Date (MM DD YY):	Reply to this request with a carriage return (CR). The current date will be written to the tape.

Rev no: Supply any decimal integer. If you reply to this request with a carriage return (CR), the revision number field is set to 0. Because this field is used only for visual identification during restore operations, you may wish to enter the revision of PRIMOS running on your system. Example:

192

Name or command: If you wish to generate an index, enter the command:

\$I [pathname] [n]

19.2

The index will be sent to the file specified by pathname, or to the terminal if pathname is not specified. n indicates the number of levels to be included in the index. (The default is two levels.) You cannot specify passwords in pathname.

After using this command, MAGSAV will issue the "Name or command" prompt again.

Name or command: Enter MFD to save the disk partition. The tape should start spinning. As the disk partition is saved, an index is generated if one was requested. If MAGSAV reaches the end of the reel of tape, it will tell you and ask for a new tape unit. Meanwhile, it will rewind the tape it was just using. Label a new tape as you did the first, as in "Tape 2 of ____". Mount the new tape on another tape drive, or mount it on the drive just used, after dismounting the previous tape. Then enter the new tape drive unit number.

When MAGSAV is finished, it will issue the "Name or command" prompt again.

Name or command: Enter \$R. This tells MAGSAV to rewind the tape.

Caution

Do not use CONTROL-P or BREAK to exit out of MAGSAV at this point. If you do, it may not be possible to read the tape back in later. Use \$R instead.

Sample MAGSAV Session

The following example illustrates a supervisor terminal session during which a disk partition, GEORGE, is saved on tape. A priority ACL is set on the partition, allowing user SYSTEM to read the disk, and preventing other users from accessing the disk. This allows the supervisor terminal to back up all data on the disk without interference from any users. (Backup to tape does not have to be done at the supervisor terminal. It is done so in this example so that the order of commands, including the priority ACL commands, can be seen. Priority ACL commands can only be issued at the supervisor terminal or by the System Administrator.)

```
OK, ASSIGN MFI
Device MFI assigned.
OK, SET_PRIORITY_ACCESS GEORGE SYSTEM:PLUR $REST:NONE
*** From PRIMOS: Priority ACL set on partition "GEORGE"
    by user "SYSTEM" (#1) at 23 Feb 83 12:03:04 Wednesday
OK, ATTACH <GEORGE>MFD
OK, MAGSAV
[MAGSAV Rev. 19.2]
Tape unit (9 Trk): 1
Enter logical tape number: 1
Tape free for us
Tape name: GEORGE
Date (MM DD YY): (CR)
Rev no: (CR)
Name or Command: MFD
Name or Command: $R
OK, ORIGIN
OK, REMOVE_PRIORITY_ACCESS GEORGE
OK, UNASSIGN MFI
Device released.
OK,
```

How MAGSAV Handles ACLs

A file or directory using ACL protection can be protected in one of three ways: it can be protected by a specific ACL; it can be protected by an Access Category; or it can use the default protection of the directory in which it resides.

Specific ACLs: Specific ACLs are always saved (unless the user gives the -NO_ACL option). Specific ACLs are saved immediately after the objects they protect.

Access Categories: When saving the MFD, all Access Categories within each directory are saved. They are written onto the tape before any other files or directories in each directory. Each object protected by an Access Category is followed on the tape by the information that it is so protected.

Default Protection: No ACL information is saved for file system objects that use default protection.

How MAGSAV Handles Quotas

MAGSAV saves quotas when it saves an MFD.

LOGICAL TAPE VERIFICATION (MAGRST)

MAGRST is the PRIMOS tape-to-disk restoration subsystem. It also allows you to verify tapes written by MAGSAV. A complete discussion of the capabilities of MAGRST may be found in the Magnetic Tape User's Guide.

The Procedure for Running MAGRST

The procedure for running MAGRST is summarized in Figure 7-8. The steps should be performed at a user terminal near the tape drives.

1. Mount the first tape. (The tape should be labeled "Tape 1 of n".)
2. Run MAGRST, as described below. During this procedure, you may mount and dismount more tapes.
3. After MAGRST has finished, dismount the last tape. (The tape should be labeled "Tape n of n".)

Invoking MAGRST

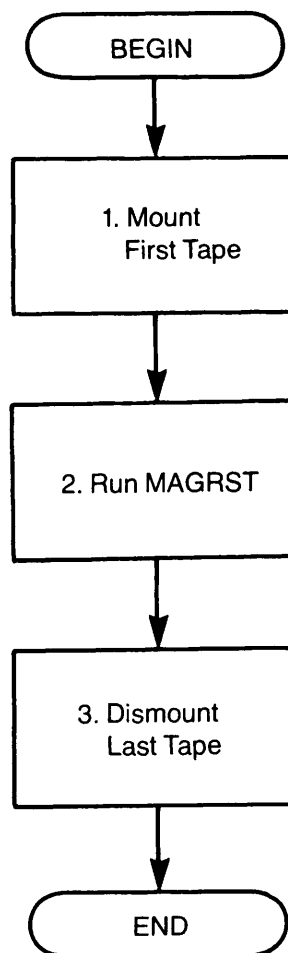
If you are using a 9-track tape drive, as is usually the case, enter the command:

MAGRST

If you are using a 7-track tape drive, enter the command:

MAGRST -7TRK

The MAGRST Dialog: After you invoke MAGRST, the dialog begins. MAGRST requests information from you in the following order. Appropriate user responses are shown.



Procedure for Running MAGRST for Verification Under PRIMOS
Figure 7-8

<u>Request</u>	<u>Response</u>
Tape unit (9-Trk):	Supply the physical or logical tape unit number (0-7).
Enter logical tape number:	Enter 1.
Name:	Verify that the information displayed here is correct. For example, the partition name should follow "Name:", the correct date should be displayed, and the reel number should be 1.
Date (MM DD YY):	
Rev no:	
Reel no:	
Ready to restore:	<p>If the information displayed above is not correct, enter NO. If the tape you mounted is the correct one, then the MAGSAV operation was not successful. Otherwise, mount the correct tape — MAGRST will start with the "Tape unit" request again.</p> <p>If the information is correct, start the verification process by entering the response:</p> <p style="text-align: center;">NW [n]</p> <p>If <u>n</u> is not specified, all filenames saved on the tape will be displayed. Specify 1 or 2 for <u>n</u> to reduce terminal output — only the first level or first two levels of filenames will be displayed.</p> <p>If MAGRST reaches the end of the reel of tape, it will tell you and ask for a new tape unit. Meanwhile, it will rewind the tape it was just using. Mount the next tape on another tape drive, or mount it on the drive just used, after dismounting the previous tape. Then enter the new tape drive unit number.</p>

When MAGRST finishes, it may print the message:

n RECOVERED ERRORS

This indicates that the MAGSAV operation was successful, but that the tape drives should be cleaned. If the message does not appear, then the operation was successful. If any other error messages appear, indicating an unrecovered error, then the MAGSAV operation was unsuccessful.

PHYSICAL DISK-TO-TAPE BACKUP (PHYSAV)

PHYSAV copies the contents of a disk partition to magnetic tape in physical track order. To minimize disk read time, all tracks (of all partitions) on one disk cylinder are written to tape, before moving the disk heads to the next cylinder.

Each disk track is written as two magnetic tape blocks of 10424 and 8192 bytes.

Since no attention is paid to logical file structure, it is not possible to restore a single file. The smallest unit that can be restored is a partition (as defined to the PHYSAV utility).

PHYSAV runs in V-mode under PRIMOS only.

After PHYSAV successfully writes the header of each tape section, it informs you with a message identifying the reel, the logical tape, and the corresponding section. A new section begins with 1 for the beginning of a logical tape, and is incremented by 1 each time a new physical magnetic tape reel is started.

Using the RAT: A more efficient save may be performed by using the Record Availability Table (RAT) of the partition being saved. When the RAT is used, PHYSAV copies only those tracks that contain records currently in use by the file system.

The one drawback to using the RAT is that information can be lost if the RAT is not correct. To guard against this, run `FIX_DISK` before running PHYSAV.

Using `FIX_DISK`: It is usually a good idea to run `FIX_DISK` on a partition before saving it. This ensures that the partition's Record Availability Table is correct. It is particularly important that the RAT be correct if:

- The RAT is to be used in the save.
- Badspot handling (on Rev. 19 disks) is to be performed when the partition is restored.

On a Rev. 19 disk, running `FIX_DISK` also takes care of any remapping of records caused by previous badspot handling, and thus ensures that the `BADSPT` file will not contain an equivalence block. If an equivalence block is present when the save is done, PHYSAV will not be able to save the partition.

The Procedure for Running PHYSAV

The procedure for running PHYSAV is summarized in Figure 7-9. The steps should be performed at a user terminal near the tape drives.

1. Assign the partition to be saved. Example:

OK, ASSIGN DISK 21060

2. Run FIX_DISK on the partition. Include the -FIX option. Example:

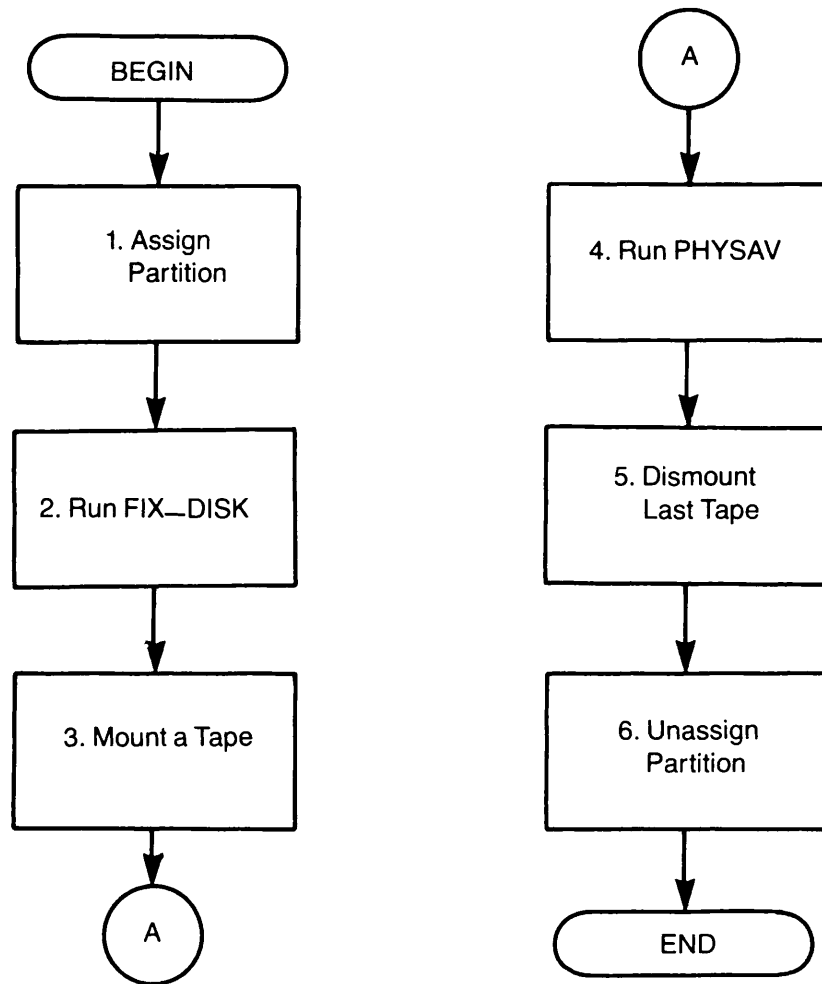
OK, FIX_DISK -DISK 21060 -FIX

3. While FIX_DISK is running, mount a tape. It is recommended that you label the tape before you mount it, with at least the following information:

- Your initials
- The date and time
- The name of the partition being saved
- The physical disk number of the partition being saved
- The name of the system
- The recording density (usually 1600 or 6250 bpi)
- The tape number of this set, as in "Tape 1 of ____". You will fill in the blank when you have determined how many tapes were needed to save the partition.
- The fact that PHYSAV is being used to save the tape — include the revision level of PRIMOS

4. Run PHYSAV, as described below. During this procedure, you may mount and dismount more tapes.
5. After PHYSAV has finished, dismount the last tape. It is very important to fill in the blanks on the tape labels to indicate the number of tapes in the set.
6. Unassign the partition. Example:

UNASSIGN DISK 21060



Procedure for Running PHYSAV Under PRIMOS
Figure 7-9

Invoking PHYSAV

If you are using a Prime 750, 850, or 9950, enter the command:

PHYSAV

Otherwise, enter the command:

PHYSAV -LOWEND

The PHYSAV Dialog: When invoked, PHYSAV responds with a series of questions. PHYSAV requests information in the order shown below. Appropriate user responses are shown.

<u>Request</u>	<u>Response</u>
UNIT NO:	Supply the logical tape unit number (0-7), or you may type CONTROL-P, as explained below. (Reenter the subsystem by typing REN.)
LOGICAL TAPE:	Enter 1.

Note

There is no check for the previous existence of logical tape 1; the tape will be written from the beginning.

COMMENT:	Supply a comment up to 80 characters long. Include the name of the partition being saved. For example: SAVE OF PARTITION BEEBLE ON 3/5/83.
PHYS.DEV.NO:	Specify the physical device number of the partition to be saved (21060, for example).
USE THE RAT (YES/NO)?	Answer YES.
PHYS.DEV.NO:	Enter a carriage return (CR).
PARAMETERS OK (YES/NO)?	Enter YES to begin the save operation.

If PHYSAV reaches the end of the reel of tape, it will tell you and ask for a new tape unit. Meanwhile, it will rewind the tape it was just using. Label a new tape as you did the first, as in "Tape 2

of ____". Mount the new tape on another tape drive, or mount it on the drive just used, after dismounting the previous tape. Then enter the new tape drive unit number.

NO — Exits from the subsystem. You may reenter by typing REN.

WRITE NEXT LOG.TAPE (YES/NO)?

Answer with NO.

Reentering PHYSAV

If you exit from PHYSAV for any reason, you may continue from the exit point by typing S. Restart facilities are also available at different points in the PHYSAV dialog by typing REN. You may do this anytime up to the final NO answer to the question:

WRITE NEXT LOG.TAPE (YES/NO)?

PHYSAV asks this question after it delivers a message announcing a completed save operation. With a NO response, you exit from the subsystem completely. If you respond with YES, you reenter the subsystem.

If PHYSAV is reentered either this way or with the REN command, the subsystem does not start from the beginning. Rather, it begins from the latest, most convenient point. For example:

- If the magnetic tape unit is not assigned, you can type CONTROL-P in response to the UNIT NO: question, assign the magnetic tape unit, then reenter the subsystem at this point by typing REN.
- If you have not assigned the partition to be saved, the subsystem exits. You may then assign the partition and restart the subsystem from this point by typing REN. You will have to reenter the physical disk number.
- If you exit from the subsystem once the save operation has begun, you may continue from the exit point by typing START, or restart the latest section (logical tape or current reel, whichever is the most recent) by typing REN.

Sample PHYSAV Session

The following example illustrates a terminal session using PHYSAV:

```

OK, PHYSAV
REV 19.0
DATE : JAN 18, 1982    TIME : 08.41
UNIT NO: 0
LOGICAL TAPE: 1
COMMENT:
This is a save of the BEEBLE partition.
PHYS.DEV.NO: 21060
USE THE RAT (YES/NO)? YES
PHYS.DEV.NO: (CR)

DISK                                HEAD OFFSET, #HEADS,  SAVING
021060    BEEBLE                    4          4      RECORDS USED

PARAMETERS OK (YES/NO)? YES
REEL:      1  LOG.TAPE:      1  SECTION:      1
SAVE COMPLETE
WRITE NEXT LOG.TAPE (YES/NO)? NO
OK,

```

PHYSICAL TAPE VERIFICATION (PHYRST)

PHYRST can be used to verify tapes generated by PHYSAV. For more information on PHYRST, see Chapters 8 and 13.

The Procedure for Running PHYRST

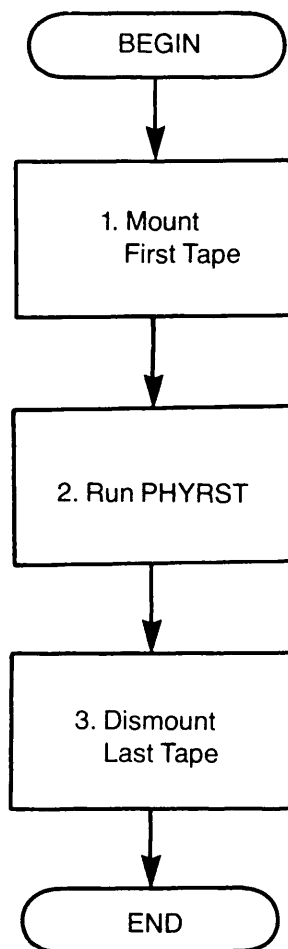
The procedure for running PHYRST is summarized in Figure 7-10. The steps should be performed at a user terminal near the tape drives.

1. Mount the first tape. This is the tape labeled "Tape 1 of n".
2. Run PHYRST, as described below. During this procedure, you may mount and dismount more tapes.
3. After PHYRST has finished, dismount the last tape. This is the tape labeled "Tape n of n".

Invoking PHYRST

Enter the command:

```
PHYRST
```

Procedure for Running PHYRST for Verification Under PRIMOS
Figure 7-10

The PHYRST Dialog: After you invoke PHYRST, the dialog begins. PHYRST requests information from you in the following order. Appropriate user responses are shown.

<u>Request</u>	<u>Response</u>
UNIT NO:	Supply the physical or logical tape unit number (0-7).
LOGICAL TAPE:	Enter 1.
CORRECT TAPE (YES/NO)?	<p>Verify that the information displayed is correct. For example, make sure the partition name is correct, and that the date and time displayed represent the date and time you believe the tape was made.</p> <p>If the information displayed is not correct, enter NO. If the tape you mounted is the correct one, then the PHYSAV operation was not successful. Otherwise, mount the correct tape — PHYRST will start with the "UNIT NO" request again.</p> <p>If the information is correct, enter YES.</p>
RESTORE OR VERIFY (RE/VE)?	Enter VE. This starts the verification process. If PHYRST reaches the end of the tape, it will tell you and ask for a new tape unit. Meanwhile, it will rewind the tape it was just using. Mount the next tape on another tape drive, or mount it on the drive just used, after dismounting the previous tape. Then enter the new tape drive unit number.
VERIFY COMPLETE	This message indicates that the tape is readable. If an error message is displayed instead, the tape is unreadable and the PHYSAV operation has failed to write a readable tape.
RESTORE/VERIFY NEXT LOG.TAPE (YES/NO)?	Enter NO.

8

Restoring Files From Backups

INTRODUCTION

Restoring files from backups is a procedure for recovering files, directories, or entire partitions that have been accidentally deleted or overwritten, as a result of user error or hardware failure. If backup copies are available, the most recent copy can be restored. Modifications made since the backup copy was written will have been lost, but quite often the amount of time needed to reinstate recent modifications is minimal in comparison to the time required to regenerate a new copy from scratch.

There are two kinds of restores, partial and partition. In a partial restore, specific files and directories are restored from a backup copy of the disk partition involved. Partial restores are often performed as a result of user error — the accidental deletion or overwriting of a file or directory. In a partition restore, an entire partition is restored from its backup copy. Partition restores are often performed following a hardware failure, such as a head crash.

This chapter explains:

- Partial restore from disk via the COPY command
- Partition restore from disk via the COPY_DISK command
- Partial restore from tape via the MAGRST command
- Partition restore from tape via the MAGRST or PHYRST command

- Emergency restore of command disk from backup disk via the COPY_DISK command
- Emergency restore of command disk from tape via the MAGRST command

Note

If your installation uses incremental backups, which save time and tape during the backup procedure, the restore procedure may take more time. If incremental backups are being performed, you must first restore the file, directory, or partition from the most recent full backup copy of the partition involved. Then, you must restore each available succeeding incremental backup copy of the file, directory, or partition, because a more recent copy of it may exist.

SYSTEM PREPARATION PROCEDURES

As with backup procedures, restore procedures must be insulated from user interference to prevent inconsistent results. To provide this protection, use the system preparation procedures described in Chapter 7 to prepare your system for the restore procedures you will be performing.

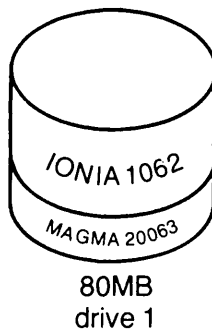
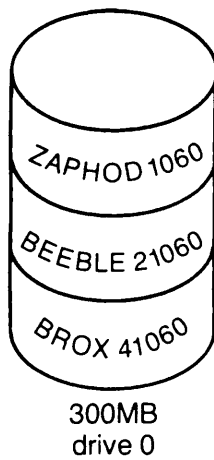
PARTIAL RESTORE FROM DISK (COPY)

To restore selected files and directories from a backup disk pack, the COPY command is performed between two separate disk drives. The disk pack containing the corrupted or missing information remains in place, and the backup pack containing the correct (although dated) copy of the information is inserted into a second disk drive, temporarily replacing the disk pack that normally resides there.

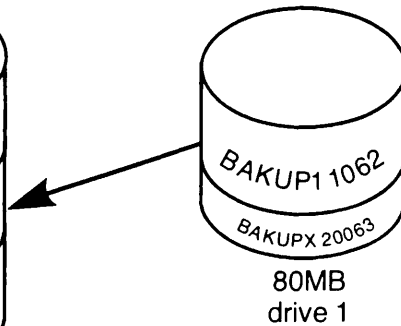
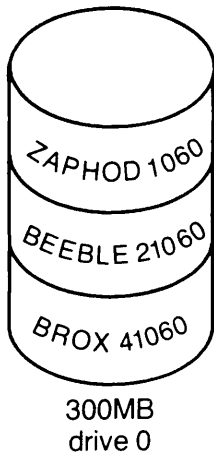
Therefore, all of the partitions that reside on the disk pack that is normally mounted on the second disk drive must be shut down before the disk pack is removed. Figure 8-1 illustrates an example of this procedure.

The partial restoration from disk procedure is summarized in Figure 8-2. It uses the example shown in Figure 8-1 for clarification. It assumes that you have already set a priority ACL on the target partition involved (BEEBLE). If you have not done this, follow the procedures outlined in Chapter 7. This procedure should be performed at the supervisor terminal, except where otherwise specified.

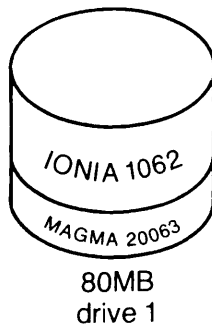
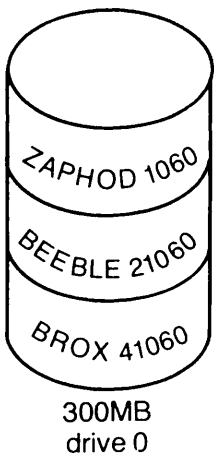
Restoring Files From Backups



Normal system operation.
(Paging disk not shown.)
Partitions BEEBLE, IONIA,
and MAGMA are shut down,
to restore partition
BEEBLE.

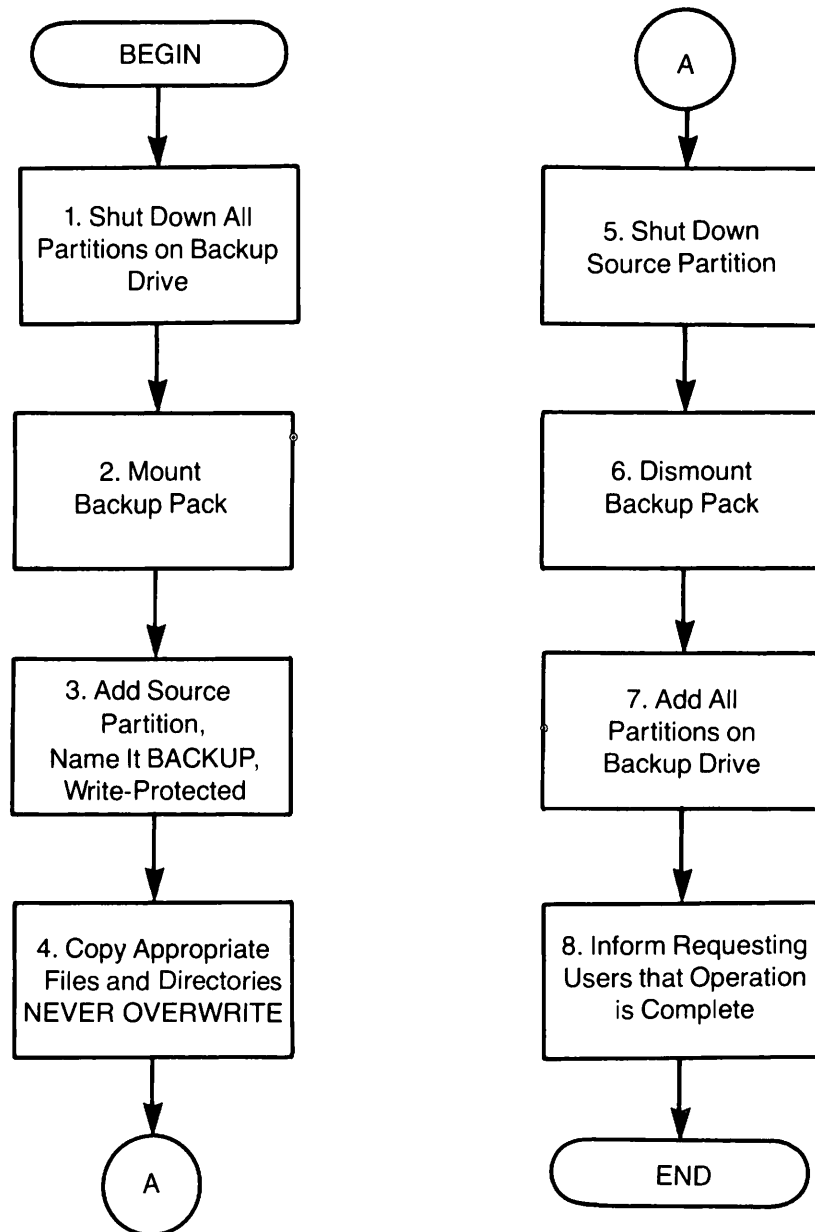


The IONIA pack is replaced
with the backup pack. The
backup partition is then
copied to partition
BEEBLE.



After the operation is
complete, the IONIA pack
is re-mounted. Partitions
BEEBLE, IONIA, and MAGMA
are started up again.
System operation is
restored.

Example of Disk-to-disk Restore Procedure Under PRIMOS
Figure 8-1



Partial Restoration Using COPY
Figure 8-2

1. At the supervisor terminal, shut down all of the disk partitions that reside on the disk pack that you will be replacing with the backup pack. In the example, this means the IONIA and MAGMA partitions are shut down.

```
OK, SHUTDN 1062
OK, SHUTDN 20063
```

2. Remove the pack mounted in the disk drive to be used as the backup drive, and replace it with the backup pack. In the example, this means the pack containing the IONIA and MAGMA partitions is removed, and replaced with the pack labeled BAKUPl.

3. Add the source partition with the name BACKUP, write-protected. Example:

```
OK, ADDISK 1062 -RENAME BACKUP
OK, SET_PRIORITY_ACCESS BACKUP SYSTEM:LUR $REST:NONE
```

4. Copy the appropriate files and directories from the source partition to the target partition. Change the name of the destination file or directory so as to avoid overwriting an existing file or directory, and write down the destination pathname — you will need it later. Use the -COPY_ALL option to specify that all attributes are to be copied. For example:

```
OK, COPY <BACKUP>FRED>MY_FILE <BEEBLE>FRED>MY_FILE.BAK -COPY_ALL
```

When copying a directory, COPY will request verification as follows:

OK to copy directory "<BACKUP>FRED" to "<BEEBLE>FRED.BAK"?

Answer YES. However, COPY might ask a question of the form:

"pathname" already exists, do you wish to overwrite it?

If this happens, answer "NO". Reissue the COPY command with a different destination pathname.

5. When you have finished copying the appropriate files and directories, shut down the source partition, restoring its original name. For example:

```
OK, SHUTDN 1062 -RENAME BEEBLE
```

6. Remove the backup pack mounted in the source disk drive, and replace it with the original pack. In the example, this means the pack labeled BAKUPl is removed, and replaced with the pack containing the IONIA and MAGMA partitions.

7. Add the partitions on the backup drive. For example:

```
OK, ADDISK 1062
OK, ADDISK 20063
```

8. Inform the requesting user that the operation is complete. Include the destination pathname of the restored file or directory in the information.

PARTITION RESTORE FROM DISK (COPY_DISK)

COPY_DISK may be used to restore data from one physical disk (the source disk or partition) to another (the target disk or partition). After COPY_DISK is used, FIX_DISK should be run on the target disk for badspot handling if COPY_DISK so indicates. Instructions for running FIX_DISK are included in the two COPY_DISK procedures described below.

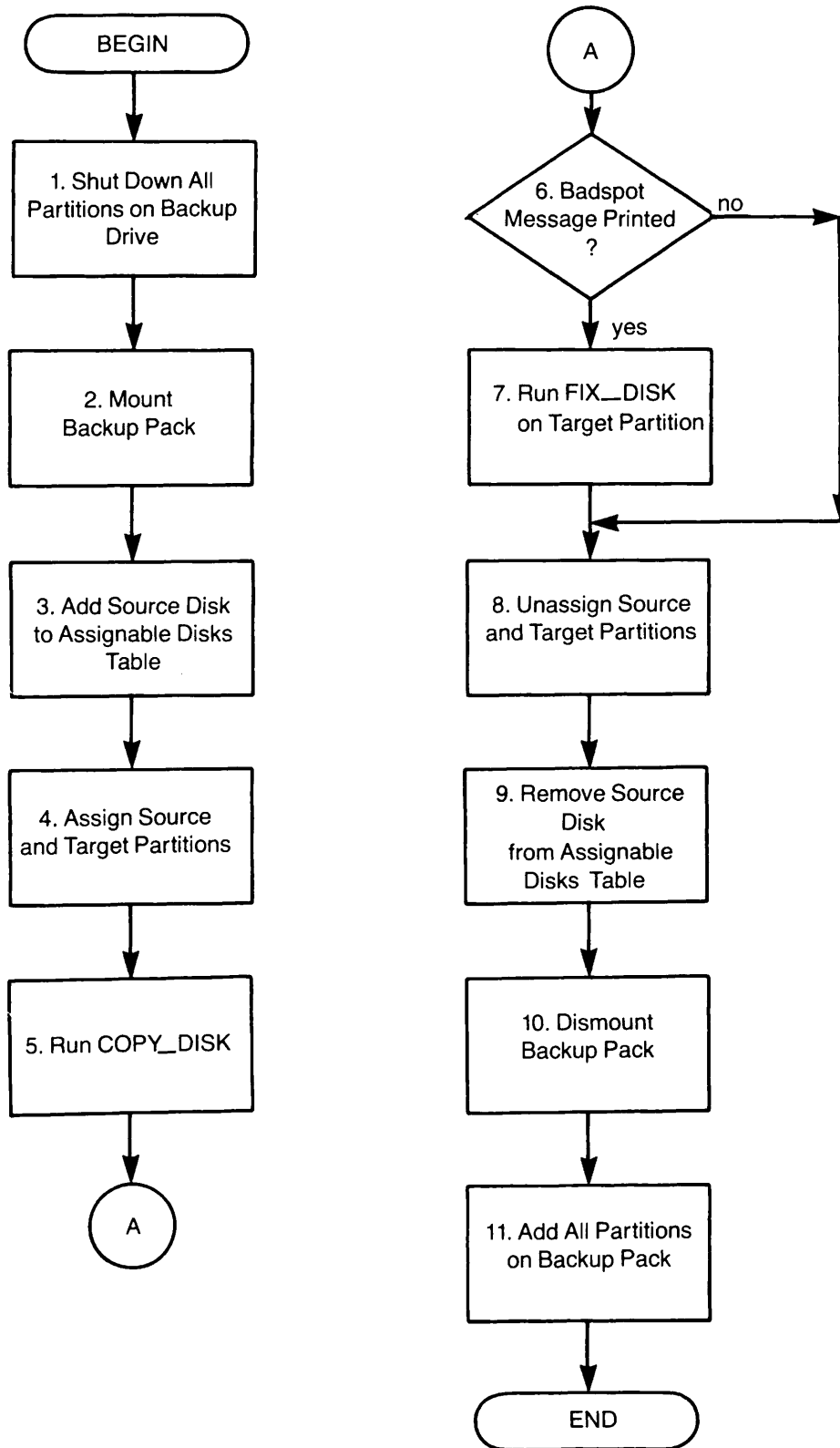
The Procedure for Running COPY_DISK

Disk-to-disk partition restores are performed either between two separate disk drives, or between the removable and non-removable portions of a CMD. When the restoration is performed between two disk drives, the disk pack containing the target partition remains in place, and the backup disk pack is inserted into the second disk drive, temporarily replacing the disk pack that normally resides in the second disk drive. When the restore is performed between the removable and non-removable portions of a CMD, the non-removable portion is usually being restored, with one or more extra platters being inserted as the removable portion and serving as the backup disk.

Therefore, all of the partitions that reside on the disk pack that is normally mounted on the second disk drive (or the removable portion of the CMD) must be shut down before the disk pack is removed.

The procedure for running COPY_DISK depends upon whether it is to be run under PRIMOS or PRIMOS II.

Running COPY_DISK Under PRIMOS: This procedure is summarized in Figure 8-3. It uses the example shown in Figure 8-1 for clarification. It assumes that you have already shut down the target partition (BEEBLE) and added it to the assignable disks table. If you have not done this, follow the procedures outlined in Chapter 7. This procedure should be performed at a user terminal, except where otherwise specified.



Partition Restoration Using COPY_DISK Under PRIMOS
Figure 8-3

1. At the supervisor terminal, shut down all of the disk partitions that reside on the disk pack that you will be replacing with the backup pack. In the following example, this means the IONIA and MAGMA partitions are shut down.

```
OK, SHUTDN 1062
OK, SHUTDN 20063
```

2. Remove the pack mounted in the disk drive to be used as the backup drive, and replace it with the backup pack. In the example, this means the pack containing the IONIA and MAGMA partitions is removed and replaced with the pack labeled BAKUPl.
3. At the supervisor terminal, add the source partition to the assignable disks table. In the example, this is done with command:

```
OK, DISKS 1062
```

Note

That the physical disk number in this example, 1062, is identical to one of the numbers shown in the example for step 3, is purely coincidental. In this example, IONIA happens to be a four-head partition, as is the disk being backed up (BEEBLE), and therefore corresponds to the partitioning of the backup pack (BAKUPl).

4. Assign the source and target partitions at the user terminal. For example:

```
OK, ASSIGN DISK 1062
OK, ASSIGN DISK 21060
```

5. Run COPY_DISK, as described below. Example:

```
OK, COPY_DISK
COPY_DISK 19.2

FROM PHYS DISK=1062
40MB STORAGE MOD? NO
600MB STORAGE MOD? NO
TO PHYS DISK= 21060
600MB STORAGE MOD? NO
FROM, TO, RECORDS = 001062, 021060,      29628
PARAMETERS OK? YES
```

6. COPY_DISK may print the message:

BADSPOTS HANDLED ON PARTITION pdev - PLEASE RUN FIX_DISK

If it does not print the message before returning to PRIMOS, go to Step 8. If it does, continue with Step 7.

7. Run FIX_DISK on the target partition (pdev); include the -FIX option. Example:

OK, FIX_DISK -DISK 21060 -FIX -LEVEL 0

8. Unassign the source and target partitions. Example:

OK, UNASSIGN DISK 21060

OK, UNASSIGN DISK 1062

9. Remove the source partition from the assignable disks table, using the supervisor terminal. Example:

OK, DISKS NOT 1062

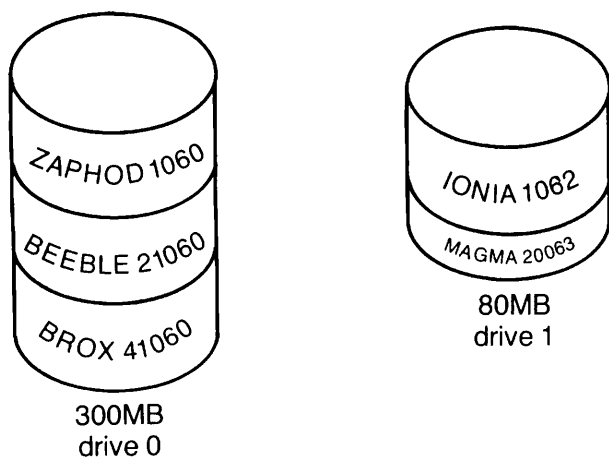
10. Remove the backup pack mounted in the source disk drive, and replace it with the original pack. In the example, this means the pack labeled BAKUP1 is removed, and replaced with the pack containing the IONIA and MAGMA partitions.

11. Add the partitions now mounted on the backup drive, at the supervisor terminal. In this example, this means the IONIA and MAGMA partitions:

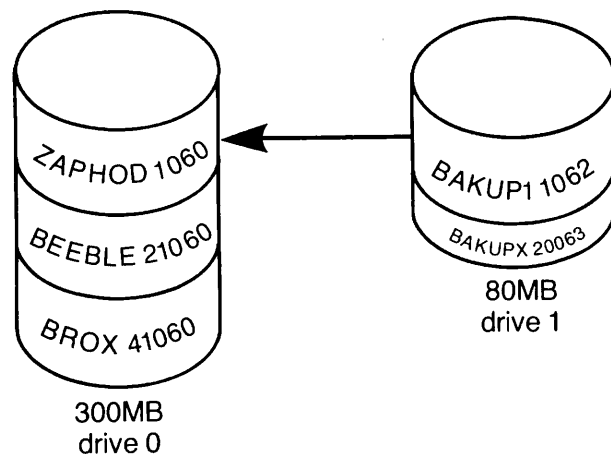
OK, ADDISK 1062

OK, ADDISK 20063

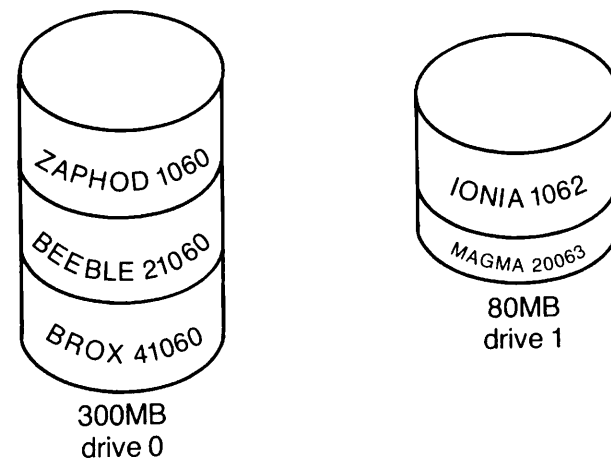
Running COPY_DISK Under PRIMOS II: This procedure is summarized in Figure 8-5. It uses the example shown in Figure 8-4 for clarification. It assumes that you have already logged out all users. If you have not done this, follow the procedures outlined earlier in this chapter. You must perform all of the steps in this procedure at the supervisor terminal.



Normal system operation.
(Paging disk not shown.)
Partitions ZAPHOD, IONIA,
and MAGMA are shut down,
to restore partition
ZAPHOD.

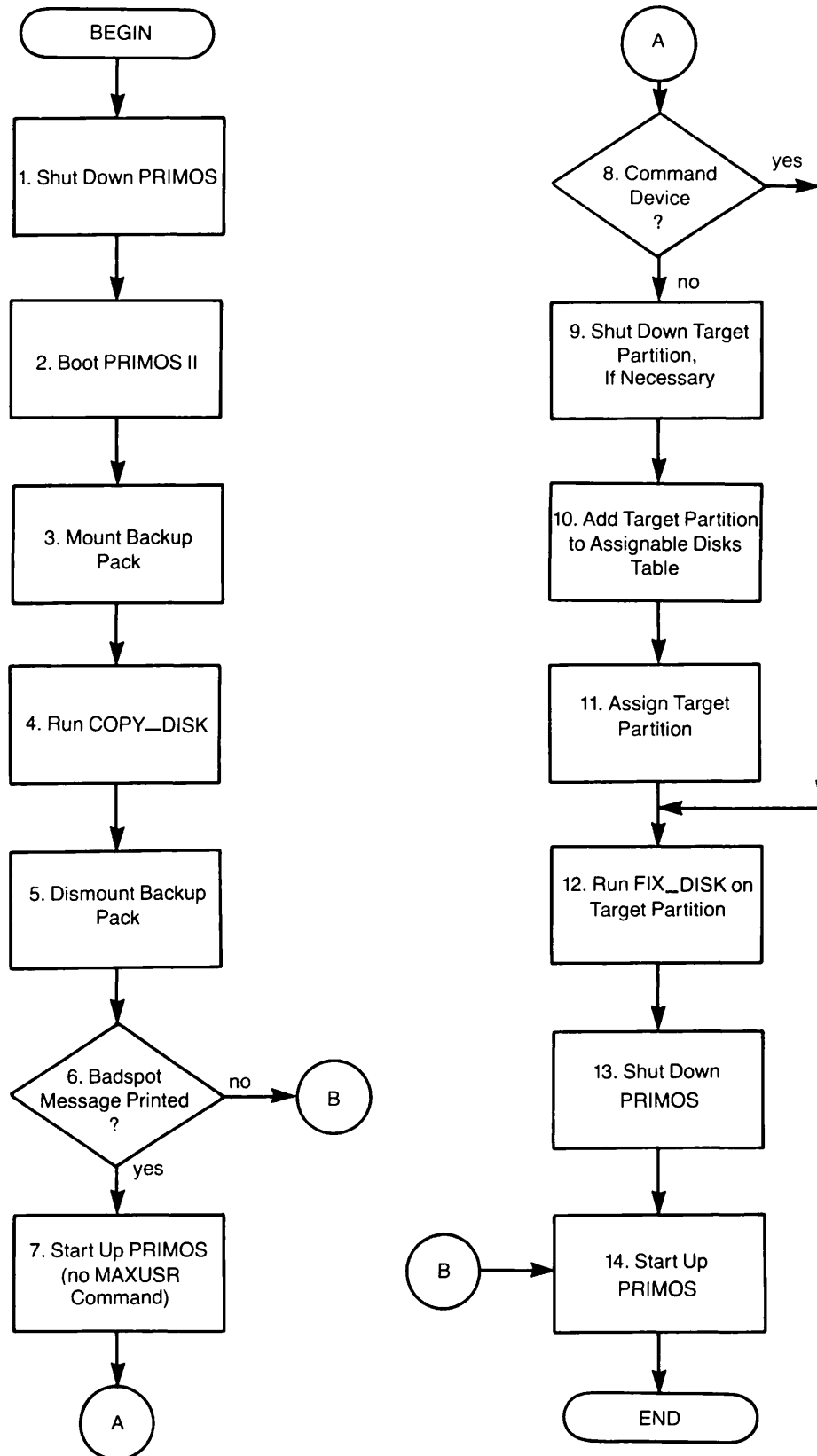


The IONIA pack is replaced
with the backup pack. The
backup partition is then
copied to partition
ZAPHOD.



After the operation is
complete, the IONIA pack
is re-mounted. Partitions
ZAPHOD, IONIA, and MAGMA
are started up again.
System operation is
restored.

Example of Disk-to-disk Restore Procedure Under PRIMOS II
Figure 8-4



Partition Restoration Using COPY_DISK Under PRIMOS II
Figure 8-5

1. Shut down PRIMOS. Example:

```
OK, SHUTDN ALL
REALLY? YES
WAIT.
PRIMOS NOT IN OPERATION
```

2. Boot in PRIMOS II. See Volume I of this book for information of how to boot PRIMOS II on your system. See 14 for information on PRIMOS II. If you accidentally boot PRIMOS, simply go to Step 1 again.
3. Remove the pack mounted in the disk drive to be used as the backup drive, and replace it with the backup pack. In the example, this means the pack containing the IONIA and MAGMA partitions is removed, and replaced with the pack labeled BAKUP1.
4. Run COPY_DISK, as described below. Example:

```
OK: COPY_DISK
COPY_DISK 19.2

FROM PHYS DISK=1062
40MB STORAGE MOD? NO
600MB STORAGE MOD? NO
TO PHYS DISK = 1060
40MB STORAGE MOD? NO
600MB STORAGE MOD? NO
FROM, TO, RECORDS = 001062, 001060,      29628
PARAMETERS OK?YES
```

5. When COPY_DISK has finished, remove the pack mounted in the backup disk drive, and replace it with the original pack. In this example, this means the pack labeled BAKUP1 is removed, and replaced with the pack containing the IONIA and MAGMA partitions.
6. COPY_DISK may print the message:

```
BADSPOTS HANDLED ON PARTITION pdev - PLEASE RUN FIX_DISK
```

If it does not print the message before returning to PRIMOS II, go to Step 14. If it does, continue with Step 7.

7. Start up PRIMOS, but do not allow users to log in (do not issue the MAXUSR command). Example:

```
OK: PRIMOS
```

8. If the target partition is the command device (logical disk 0, or COMDEV), go to Step 12. Otherwise, continue to Step 9. In the example, ZAPHOD is the command device, but Steps 9 through 11 will include examples as if it weren't.

9. Shut down the target partition. Example:

OK, SHUTDN 1060

10. Add the target partition to the assignable disks table. Example:

OK, DISKS 1060

11. Assign the target partition. Example:

OK, ASSIGN DISK 1060

12. Run FIX_DISK on the target partition. Include the -FIX option. Include the -COMMAND_DEVICE option if the target partition is the command device. Example:

OK, FIX DISK -DISK 1060 -FIX -COMMAND_DEVICE

13. Shut down PRIMOS. (This is done because the supervisor terminal may no longer have an origin directory. You may skip Steps 13 and 14 if you wish. However, you should issue the MAXUSR command to allow user login.) Example:

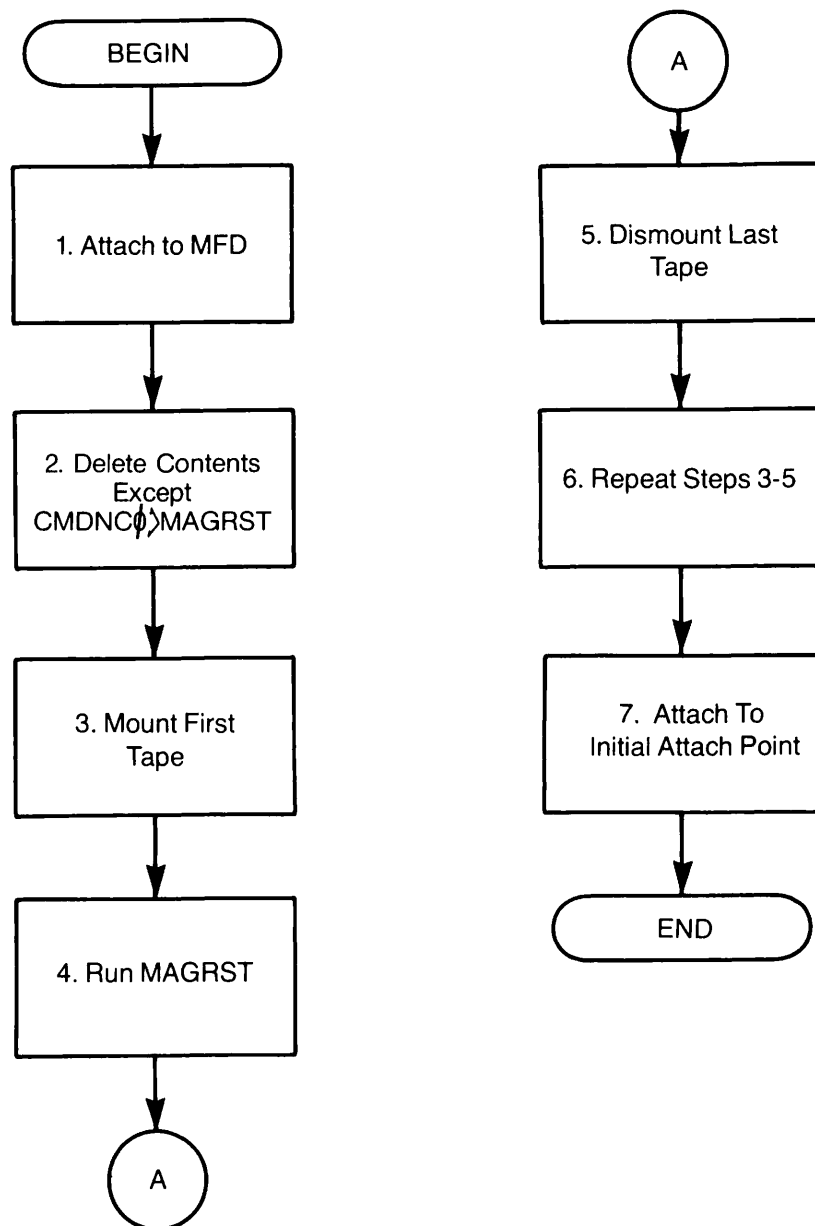
OK, SHUTDN ALL
 REALLY? YES
 WAIT,
 PRIMOS NOT IN OPERATION

14. Start up PRIMOS.

PARTITION RESTORE FROM TAPE (MAGRST)

The MAGRST command can be used to restore an entire disk partition from a tape generated by MAGSAV. Before restoring a partition, the current contents of the partition should be deleted. This is not necessary if the partition has just been created using the MAKE utility described in Chapter 6, as it will effectively have no contents. But if the partition contains many files and directories, attempting to restore an older copy of the partition from tape may produce inconsistencies, and may also result in a disk-full situation.

The procedure for restoring a partition from tape using MAGRST is summarized in Figure 8-6. It assumes that you have already set a priority ACL on the partition. If you have not done this, follow one of the system preparation procedures outlined in Chapter 7. In the examples that follow, this partition is named GEORGE. This procedure should be performed at a user terminal.



Partition Restoration Using MAGRST
Figure 8-6

1. Attach to the MFD of the partition to be restored. If you just created the partition using MAKE, according to the procedure described in Chapter 6, then you have already converted the partition to an ACL partition, or set the correct MFD password. If you have not done this, do it now. Then attach to the MFD again. (If the partition is not an ACL partition, you must specify the owner password of its MFD when you attach to it.) For example:

OK, ATTACH <GEORGE>MFD

2. If you did not just create the partition using MAKE, then you should delete the contents of the partition. (Presumably the current contents on the disk are of no value, since you are restoring an older backup copy of the partition.) Delete the contents by entering the command:

OK, DELETE ^CMDNC0 -NO_QUERY -NO_VERIFY -FORCE

(The ^CMDNC0 specifies that all files and directories except CMDNC0 are to be deleted. The ^ character is part of the wildcard capability of PRIMOS. See the Prime User's Guide.) Several error messages will be immediately displayed — ignore them. They occur because special files in the MFD — the DSKRAT, BOOT, MFD, and BADSPT files — cannot be deleted as requested. This is normal, because you don't want to delete them, even though the command you typed requested it. However, the ^CMDNC0 in the command told PRIMOS not to delete the CMDNC0 directory. If you are restoring the command partition (COMDEV), you need to preserve CMDNC0 so that you can run MAGRST.

The delete process may take a while, depending on the number of records in use on the disk. Ignore any "File open on delete" and "The directory is not empty" messages that may be displayed. When the delete process finishes, it will issue the prompt:

ER!

Again, this is normal — it reflects the fact that error messages were displayed. Now, delete most of CMDNC0 by typing the command:

ER! DELETE *>CMDNC0>^MAGRST -NO_QUERY -NO_VERIFY -FORCE

Ignore any error messages that may be displayed. Such error messages include:

Not found. *>CMDNC0 (std\$cp)
File open on delete. ">CMDNC0>DELETE.RUN" (delete)

3. Mount the first tape, with the write ring removed, to protect against accidental erasure. The tape should be labeled "Tape 1 of n". You may perform this step while step 2 proceeds.
4. Run MAGRST, as described below. During this procedure, you may mount and dismount more tapes.
5. After MAGRST has finished, dismount the last tape. This tape should be labeled "Tape n of n".
6. If there are any available incremental tapes saved after the full save that you just restored, repeat steps 3 through 5 using each set of incremental tapes in the order they were saved.
7. Attach back to your origin directory, as follows:

OK, ORIGIN

Invoking MAGRST

If you are using a 9-track tape drive, as is usually the case, enter the command:

MAGRST

If you are using a 7-track tape drive, enter the command:

MAGRST -7TRK

The MAGRST Dialog: After you invoke MAGRST, the dialog begins. MAGRST requests information from you in the following order. Appropriate user responses are shown.

<u>Request</u>	<u>Response</u>
Tape unit (9 Trk):	Supply the physical or logical tape unit number (0-7).
Enter logical tape number:	Enter 1.
Name:	Verify that the information displayed here is correct. For example, the partition name should follow "Name", the correct date should be displayed, and the reel number should be 1.
Date (MM DD YY):	
Rev. no.:	
Reel no.:	

Ready to restore: If the information displayed above is not correct, enter NO. If the tape you mounted is the correct one, then the MAGSAV operation was not successful. Otherwise, mount the correct tape -- MAGRST will start with the "Tape unit" request again.

If the information is correct, start the restore process by entering the response:

YES

If MAGRST reaches the end of the reel of tape, it will tell you and ask for a new tape unit. Meanwhile, it will rewind the tape it was just using.

Mount the next tape on another drive, or mount it on the drive just used, after dismounting the previous tape. Then enter the new tape drive unit number.

PARTIAL RESTORE FROM TAPE (MAGRST)

The MAGRST command can be used to restore selected files and directories from a backup tape of a partition generated by MAGSAV. The selected files and directories are restored into an empty directory. They are then individually copied to the directory where each belongs, using the COPY command.

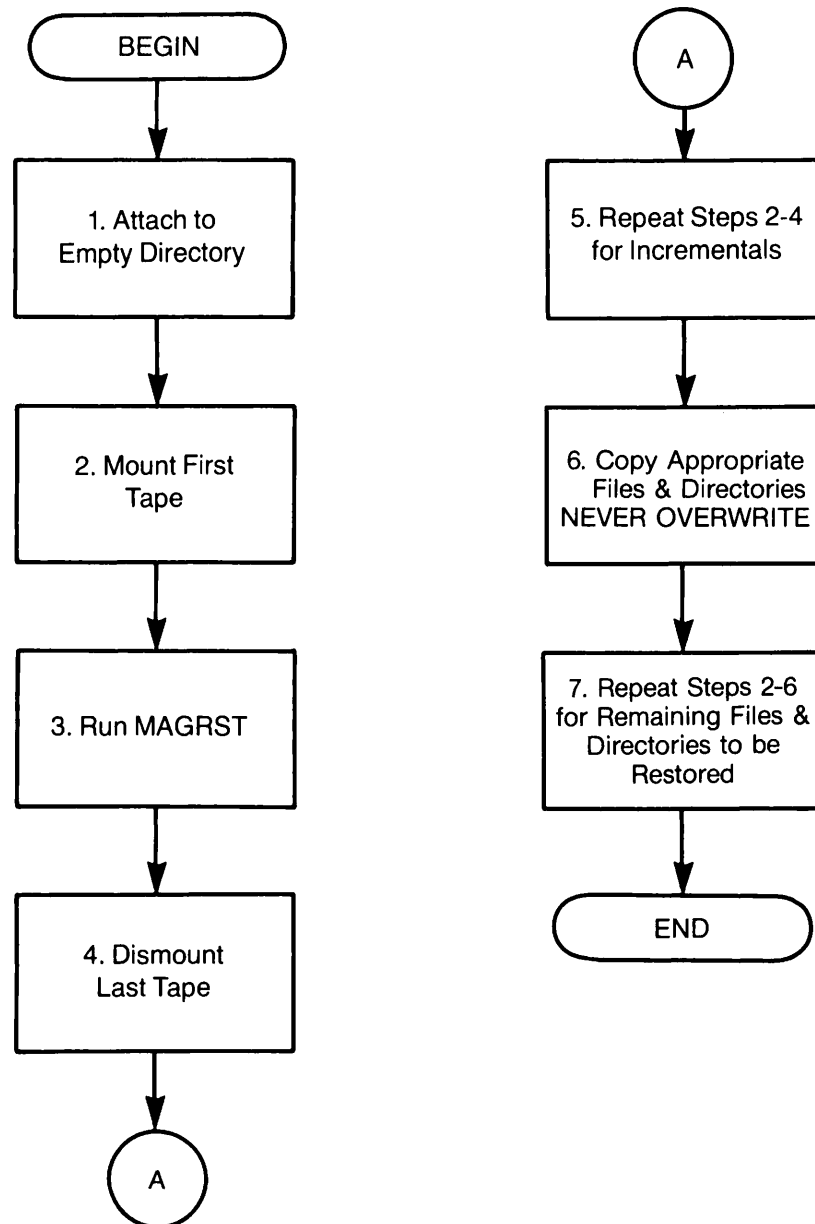
The procedure for restoring selected files and directories from tape using MAGRST is summarized in Figure 8-7. It assumes that you have already set a priority ACL on the partition to which the files and directories are being restored. If you have not done this, follow one of the system preparation procedures outlined in Chapter 7. In the examples that follow, the partition to which the files and directories are being restored is named GEORGE. The files and directories being restored are:

<GEORGE>FRED>MY_BIG_FILE	(file)
<GEORGE>SUSAN>SCHEDULE_GRID.DIR	(directory)
<GEORGE>FRED>IMPORTANT_PHONE_NUMBERS	(file)

This procedure should be performed at a user terminal.

1. Attach to an empty directory on the partition. You may have to create one first. For example:

```
OK, CREATE <GEORGE>TEMPORARY_DIR
OK, ATTACH <GEORGE>TEMPORARY_DIR
```



Partial Restoration Using MAGRST
Figure 8-7

2. Mount the first tape, with the write ring removed, to protect against accidental erasure. The tape should be labeled "Tape 1 of n".
3. Run MAGRST, as described below. During this procedure, you may mount and dismount more tapes.
4. After MAGRST has finished, dismount the last tape. This tape should be labeled "Tape n of n".
5. If there are any available incremental tapes saved after the full save you just restored, repeat steps 2 through 4 using each set of incremental tapes in the order they were saved.
6. The files and directories restored from the tape now reside in the directory to which you are attached. Copy each file and directory to the directory in which it belongs. Change the name of the destination file or directory so as to avoid overwriting an existing file or directory, and write down the destination pathname — you will need it later. Use the `-COPY_ALL` option to specify that all attributes are to be copied, and use the `-DELETE` option to delete the file or directory from your current directory after it has been successfully copied. For example:

OK, COPY MY_BIG_FILE <GEORGE>FRED>MY_BIG_FILE.BAK -COPY_ALL -DELETE

When copying a directory, COPY will request verification as follows:

OK to copy directory "SCHEDULE_GRID.DIR" to
"<GEORGE>SUSAN>SCHEDULE_GRID.DIR.BAK"?

Answer YES. However, COPY might ask a question of the form:

"pathname" already exists, do you wish to overwrite it?

If this happens, answer "NO". Reissue the COPY command with a different destination pathname.

An error message of this form may be displayed by the COPY command:

Not found. "name" (COPY)

If name is not a pathname (does not contain the ">" symbol), then the file does not exist in the directory to which you are attached. This means that a file or directory of that name was not restored during Step 3. If you misspelled the name, reissue the COPY command with the correct spelling. If you did not misspell the name, then that file or directory did not exist when the backup tapes you restored were made. Continue with the restoration procedure.

When you are finished, notify the user who requested that file to be restored that the file did not exist on the backup tapes. The user might ask you to try an earlier set of tapes. (This problem can occur if the user discovered that the file or directory was missing, but did not know the date on which the file or directory was deleted.)

If name is a pathname (contains the ">" symbol), then the directory to which you are attempting to copy the file or directory does not exist. If you misspelled the name, simply reissue the COPY command with the correct spelling — the file or directory will not have been deleted from your current directory, since the COPY operation failed. If you did not misspell the name, then the user who requested that the file or directory be restored did not correctly specify the directory in which the restoration was to take place. Leave the restored file or directory in the directory to which you are attached, and continue with the restoration procedure.

When you are finished, notify the user that the file or directory was successfully restored, but not copied to the desired destination. Determine the correct directory to which the file or directory is to be copied. Prepare the system for a restoration using COPY by performing the partition shutdown procedure described in Chapter 7, so as to set a priority ACL on the destination partition (GEORGE in this example). When you reach Step 4 of that procedure, attach to the directory to which you earlier restored the file or directory, and use the COPY command as described above, specifying the correct destination directory. Then, continue with Step 5 of the partition shutdown procedure.

7. If there are any more files or directories to be restored, repeat Steps 2 through 6 to restore them.

Invoking MAGRST

If you are using a 9-track tape drive, as is usually the case, enter the command:

MAGRST

If you are using a 7-track tape drive, enter the command:

MAGRST -7TRK

The MAGRST Dialog: After you invoke MAGRST, the dialog begins. MAGRST requests information from you in the following order. Appropriate user responses are shown.

<u>Request</u>	<u>Response</u>
Tape unit (9 Trk):	Supply the physical or logical tape unit number (0-7).
Enter logical tape number:	Enter 1.
Name: Date (MM DD YY): Rev. no.: Reel no.:	Verify that the information displayed here is correct. For example, the partition name should follow "Name", the correct date should be displayed, and the reel number should be 1.
Ready to restore:	<p>If the information displayed above is not correct, enter NO. If the tape you mounted is the correct one, then the MAGSAV operation was not successful. Otherwise, mount the correct tape -- MAGRST will start with the "Tape unit" request again.</p> <p>If the information is correct, enter the response:</p> <p>PA</p> <p>This specifies that a partial restore is being performed.</p>
Tree name:	<p>Enter the pathname of each file or directory to be restored, one per line, subject to the constraints listed below. Replace the "<partition>" part of the pathname with MFD>. For example:</p> <p>MFD>FRED>MY_BIG_FILE</p> <p>MAGRST will continue prompting for pathnames in this fashion. There are two constraints when entering more than one pathname:</p> <ol style="list-style-type: none"> 1. No more than ten pathnames may be entered. 2. No two pathnames may have the same final component. <p>The second constraint refers to the actual name of each file or directory being restored. This is the portion of the pathname that follows the last > character. For example, the final component of the example pathname above is MY_BIG_FILE.</p>

MAGRST will restore this file in the current directory.

If you were to specify another pathname with the same final component, such as:

MFD>CHARLES>WORK>MY_BIG_FILE

the last file named MY_BIG_FILE read from the tape would overwrite the first file.

If you cannot enter all the pathnames that you wish to restore due to the above constraints, Step 7 of the partial restore procedure will explain how to restore the remaining files and directories.

Tree name:

When you have entered all the pathnames, enter a null pathname by hitting the carriage return (CR). This will start the restore process. If MAGRST reaches the end of the reel of tape, it will tell you and ask for a new tape unit. Meanwhile, it will rewind the tape it was just using. Mount the next tape on another drive, or mount it on the drive just used, after dismounting the previous tape. Then enter the new tape drive unit number.

How MAGRST Handles Protection

At Rev. 19, users can choose between ACL and password protection for their directories. MAGRST can handle both types of protection.

The general rule under which MAGRST works is this: if there is a conflict in a matter of protection between an object being restored from tape and an object of the same name resident on the disk, the protection on the disk is retained. The reason for this is that the version on the disk will probably be the more recent one, and thus is more likely to reflect the owner's current wishes.

ACL vs. Password UFDs: Whenever possible, MAGRST restores ACL UFDs as ACL UFDs and password UFDs as password UFDs. However, if a UFD currently on the disk has the same name as a UFD being restored from tape but is a different type, the type of the UFD on the disk will be maintained. For example, if an ACL UFD STATS is being restored from tape, but a password UFD STATS already exists on the disk, the UFD STATS will be restored as a password UFD. Similarly, if the password UFD ACCTS is restored into an ACL UFD ACCTS, it is restored as an ACL UFD, with all files and subdirectories taking on the default protection of the existing UFD.

If the tape was saved with the `-NO_ACL` option, then the UFDs will be restored as the same type (ACL or password) as the parent directory.

Caution

If MAGRST is running under PRIMOS II, all UFDs are restored as password UFDs.

How MAGRST Handles ACLs: MAGRST always tries to restore the ACL protection saved by MAGSAV. The following exceptions exist:

- If an object being restored already exists on the disk, the protection on the disk is retained, and the protection on the tape is ignored.
- An Access Category is restored only if no Access Category of that name exists on the disk. (If an Access Category of the same name does exist, the user will be warned of that fact.)
- If a file protected by an Access Category has been saved by name or is restored by name, the Access Category is not restored. (In the former case, the Access Category may not even be on the tape.) If the Access Category already exists on the disk, then the object will be protected by that ACL. Otherwise, the object will receive default protection, and an error message will be printed.

Therefore, if you wish to restore specific files, together with their Access Categories, you should first restore the Access Categories, and then restore the protected files.

Note

MAGSAV writes ACL information to the tape after it writes the object itself. Therefore, MAGRST restores ACL information to disk after it restores the object to disk. It writes the object (including all its subentries), then sets the access. This means that a user who has no rights to an object can restore the object from tape with MAGRST, but may not be able to use it once it is restored.

How MAGRST Handles Quotas

Quotas will be restored if the corresponding UFD does not already exist on the disk. If the UFD does exist, its existing quota will remain in effect.

PARTITION RESTORE FROM TAPE (PHYRST)

PHYRST restores a partition saved with PHYSAV to an assigned partition of the same size. After PHYRST is used, FIX_DISK should be run on the target disk for badspot handling if PHYRST so indicates. Instructions for running FIX_DISK are included in the PHYRST procedure described below.

The Procedure for Running PHYRST

The procedure for running PHYRST is summarized in Figure 8-8. It assumes that you have already shut down the partition to be restored and assigned a tape drive. If you have not done this, follow one of the system preparation procedures described in Chapter 7. The steps in the procedure for running PHYRST should be performed at a user terminal near the tape drives.

1. Assign the partition to be restored. For example:

OK, ASSIGN DISK 21060

2. Mount the first tape. The tape should be labeled "Tape 1 of n".
3. Run PHYRST, as described below. During this procedure, you may mount and dismount more tapes.
4. After PHYRST has finished, dismount the last tape. This tape should be labeled "Tape n of n".
5. PHYRST may print the message:

BADSPOTS HANDLED ON PARTITION pdev - PLEASE RUN FIX_DISK

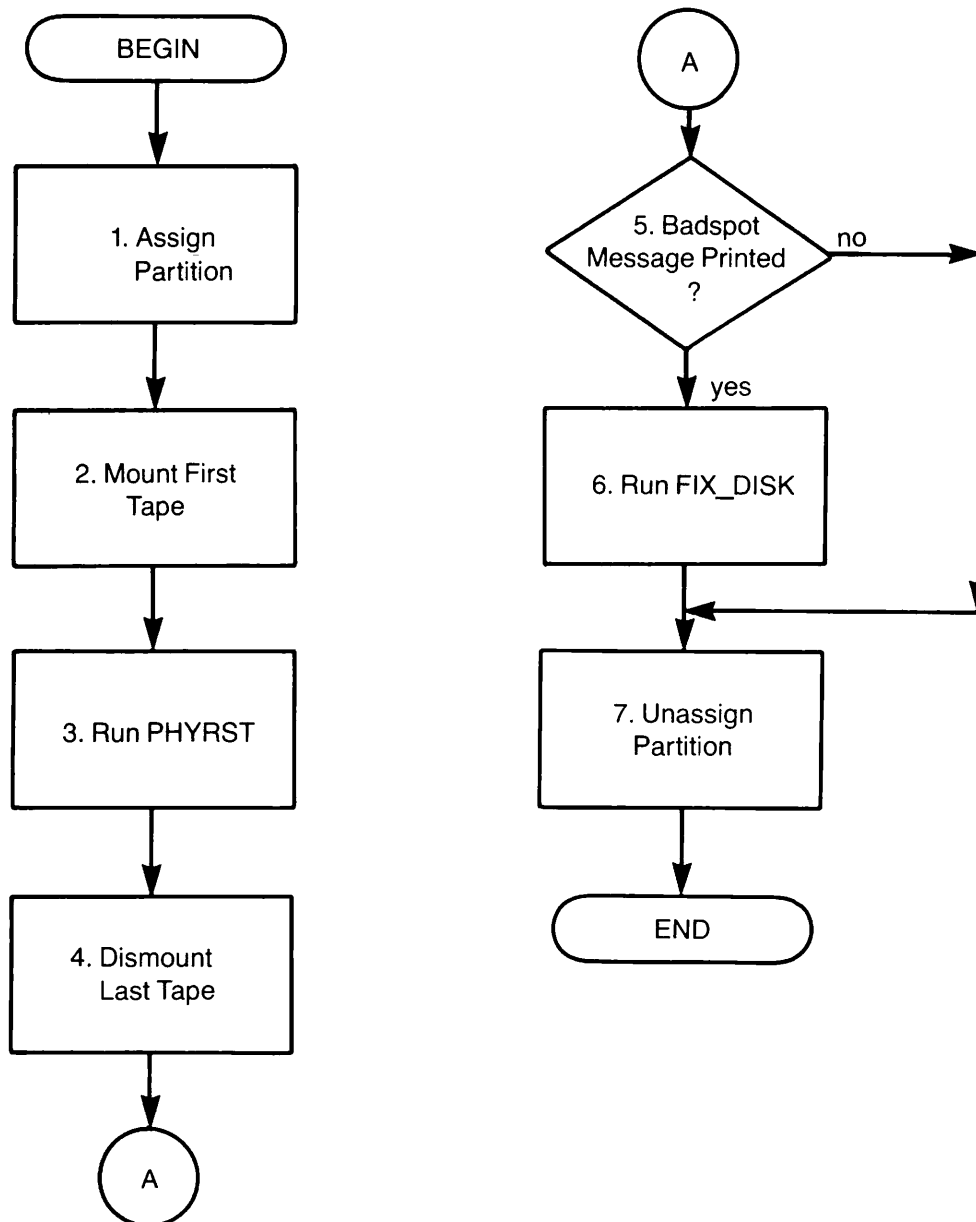
If it does not print the message before returning to PRIMOS, go to Step 7. If it does, continue with Step 6.

6. Run FIX_DISK on the partition (pdev); include the -FIX option. Example:

OK, FIX_DISK -DISK 21060 -FIX -LEVEL 0

7. Unassign the partition just restored. Example:

OK, UNASSIGN DISK 21060



Partition Restoration Using PHYRST
Figure 8-8

Invoking PHYRST

Enter the command:

PHYRST

The PHYRST Dialog: After you invoke PHYRST, the dialog begins. PHYRST requests information from you in the following order. Appropriate user responses are shown.

<u>Request</u>	<u>Response</u>
UNIT NO:	Supply the physical or logical tape unit number (0-7).
LOGICAL TAPE:	Enter 1.
CORRECT TAPE: (YES/NO)?	<p>Verify that the information displayed is correct. For example, make sure the partition name is correct, and that the date and time displayed represent the date and time you believe the tape was made.</p> <p>If the information displayed is not correct, enter NO. If the tape you mounted is the correct one, then the PHYSAV operation was not successful. Otherwise, mount the correct tape — PHYRST will start with the "UNIT NO" request again.</p> <p>If the information is correct, enter YES.</p>
RESTORE OR VERIFY (RE/VE)?	Enter RE.
RESTORE ALL PARTITIONS TO ORIGINAL POSITIONS (YES/NO)?	Enter YES.
PARAMETERS OK (YES/NO)?	<p>Check the parameters displayed to make sure the right tape is mounted. The numbers under DISK and TO BE RESTORED AS should both match <u>pdev</u> (ignoring leading zeroes), and the name of the partition should appear to the right of the <u>pdev</u> number under DISK. If the parameters are not correct, enter NO, and mount the correct tape. PHYRST will repeat the UNIT NO prompt.</p>

If the parameters are correct, enter YES. This starts the verification process. If PHYRST reaches the end of the tape, it will tell you and ask for a new tape unit. Meanwhile, it will rewind the tape it was just using. Mount the next tape on another tape drive, or mount it on the same drive just used after dismounting the previous tape. Then enter the new tape drive unit number.

RESTORE COMPLETE

This message indicates that the partition has been successfully restored. If an error message is displayed instead, the tape is unreadable and the partition has not been restored.

RESTORE/VERIFY NEXT LOG.TAPE (YES/NO)?
Enter NO.

Special Messages

If PHYRST is performing badspot handling (because the partition has badspots), it may find insufficient room on the partition to store all the records. If this happens, it will print the message:

NO FREE RECORDS AVAILABLE ON PARTITION pdev
OK TO WRITE IT WITHOUT BADSPOT HANDLING (YES/NO)?

If you answer YES, PHYRST finishes copying the partition without badspot handling. If you answer NO, PHYRST exits. You may then run PHYRST again, using a partition that has fewer badspots. In either case, the badspot information on the target disk (pdev) is destroyed. You must re-MAKE the disk to reconstruct the badspot information before it can be used again. MAKE is described in Chapter 6.

If badspot handling has occurred, then PHYRST prints the following message at the end of the copy:

BADSPOTS HANDLED ON PARTITION pdev

If this message is received, the operator should run FIX_DISK on the partition specified. FIX_DISK will correct the pointers involved in the badspot remapping, and will then erase the equivalence block.

Reentering PHYRST

You cannot reenter PHYRST with the REN command as you can with PHYSAV. If you exit or quit from PHYRST for any reason (a partition is not assigned, for example), you can reenter by typing S 1000. PHYRST restarts from the latest most convenient point. For example, if a partition is not assigned, PHYRST restarts from the beginning of the section that specifies partitions to be restored.

If you type CONTROL-P during a restore operation, PHYRST restarts from the beginning of the latest section (logical tape or current reel, whichever was most recently started). Entering the START command will always continue the subsystem.

Sample PHYRST Session

The following example illustrates a terminal session using PHYRST:

```

OK, PHYRST
REV 19.0
UNIT NO: 0
LOGICAL TAPE: 1

REEL: 1 LOG.TAPE: 1 SECTION: 1

DATE: JAN 18, 1982 AT 14:22
This is a save of the BEEBLE partition.
PARTITIONS SAVED
021060 BEEBLE                        80 or 300 MB

CORRECT TAPE (YES/NO)? YES
RESTORE OR VERIFY (RE/VE)? RE
RESTORE ALL PARTITIONS TO ORIGINAL POSITIONS (YES/NO)? YES
DISK                                TO BE RESTORED AS
021060 BEEBLE                        021060

PARAMETERS OK (YES/NO)? YES
REEL: 1 LOG.TAPE: 1 SECTION: 1
RESTORE COMPLETE
RESTORE/VERIFY NEXT LOG.TAPE (YES/NO)? NO
OK,

```

EMERGENCY RESTORE OF COMMAND DISK (COPY_DISK)

If your command disk (COMDEV) is destroyed, you will be unable to start up PRIMOS or PRIMOS II. If there is a backup pack available for this disk, you may use it to get the system running again quickly.

WARNING

Never mount a pack in a disk drive that is suspected of malfunctioning. If a disk pack seems to have been rendered unreadable, it may be due to a malfunction in the disk drive. Mounting another pack in the drive can result in the loss of the second pack as well as the first.

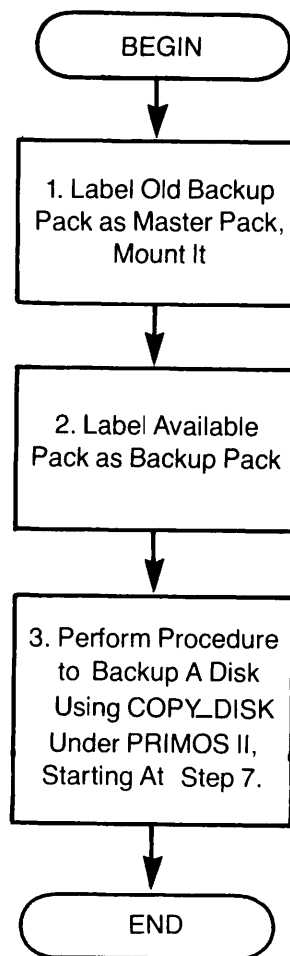
If you suspect that a disk drive may be malfunctioning, contact your System Administrator or Customer Service Representative. Do not mount another disk pack in the drive. Leave the corrupted pack in the suspect drive. Similarly, do not test the disk pack by mounting it in another drive.

In short, treat any suspicious disk pack or drive problem as a highly contagious disease. Quarantine both the drive and pack involved.

The procedure for restoring the command disk is essentially the same as the procedure to back up the command disk. In short, it involves re-labeling the backup pack as the master pack, labeling a new (available) pack as the backup pack, and performing most of the procedure to back up the command disk under PRIMOS II. The portions of this procedure that are specific to an emergency restoration are summarized in Figure 8-9, and described in detail below. This procedure assumes that the system is shut down.

Note

If your backup disk pack is not of the same type as your master disk pack, you must first copy the information on your backup disk pack to a new pack of the same type as the master disk pack. For example, if you back up a 300MB pack onto several 80MB packs, and you lose the information on the 300MB pack, you must copy the 80MB packs onto a new 300MB pack. Once you do this, you do not have to make another copy of the information on the new 300MB disk, as you already have a backup for it (the 80MB packs).



Emergency Restoration of Command Disk Using COPY_DISK
Figure 8-9

1. Label what is currently the backup pack as the master pack, and mount it in the disk drive that normally contains the master pack. Modify the label on the corrupted master pack to indicate that it is corrupted.
2. Label an available pack as the new backup pack. This pack will be used as a backup for the new master pack, in case the new master pack is also corrupted.

Strictly speaking, the old corrupted master pack is available for this purpose. However, if you use it as the new backup pack, you preclude its examination by your Customer Service Representative. Such an examination may:

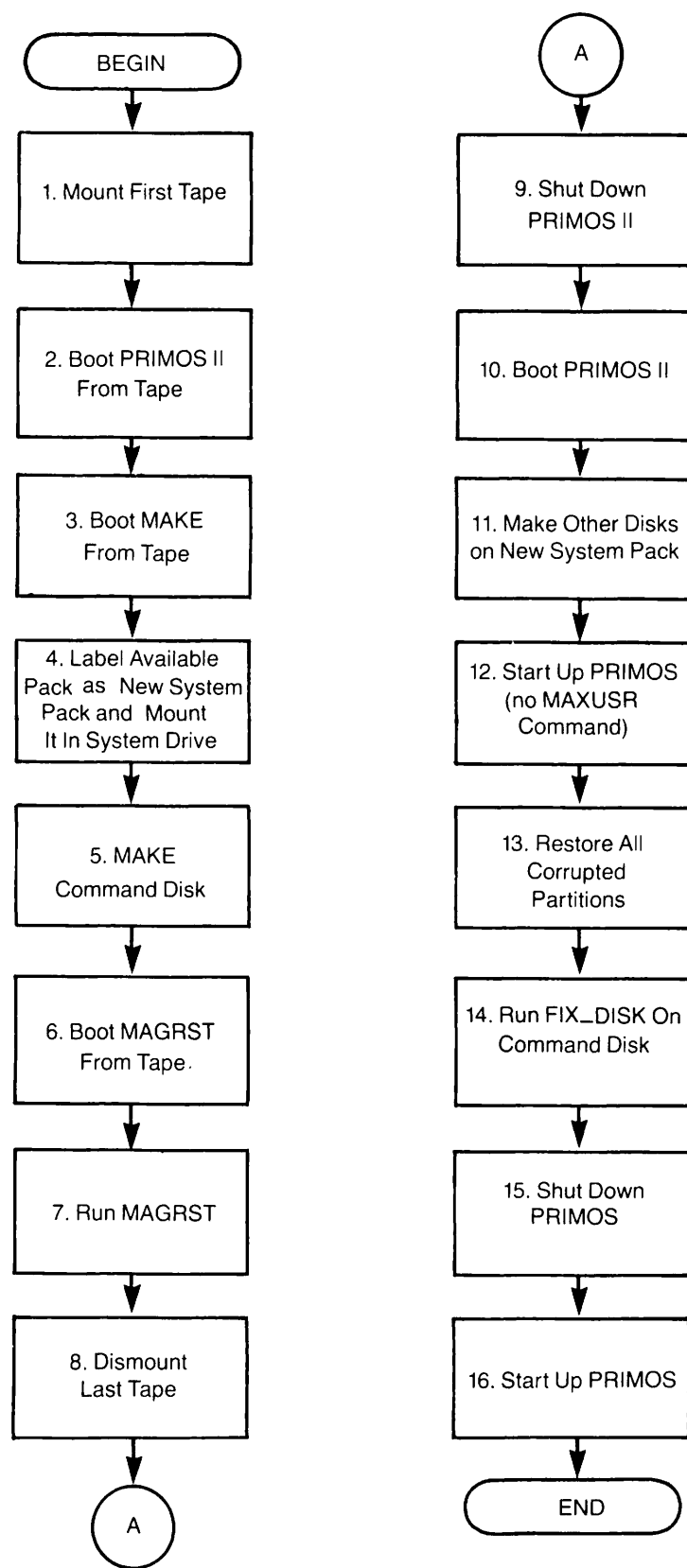
- Determine the cause of the corruption.
 - Allow the reclamation of crucial files for which the backup copies are not sufficient.
3. Perform the procedure illustrated in Figure 7-8 starting at Step 7.

EMERGENCY RESTORE OF COMMAND DISK (MAGRST)

If your command disk (COMDEV) is destroyed, you will be unable to start up PRIMOS or PRIMOS II. If there is a set of backup tapes available for this disk, you may use these tapes to get the system running again.

The procedure for restoring the command disk from tape is summarized in Figure 8-10. It assumes that the system is already shut down.

1. Mount the first tape, with the write ring removed, to protect against accidental erasure. The tape should be labeled "Tape 1 of n", where n represents the number of tapes in the set. The tape must be mounted in drive 0.
2. Boot PRIMOS II from tape. The technique for booting from tape is described in detail in Volume I of this book, but is also shown here because the structure of the tape contents is different. (The tape being used here is a backup tape of a partition, whereas the tape being used in Volume I of this book is a tape created by BOOT_CREATE. The tape used here has treenames on the tape.)



Emergency Restoration of Command Disk Using MAGRST
Figure 8-10

The procedure is:

CP> SYSCLR

*** CPU VERIFIED ***

CP> BOOT 15

TRENAME= MFD>DOS>*DOS64

PRIMOS II REV 19.1 11/22/82 (AT 170000)

OK:

3. Boot MAKE from tape, as follows:

OK: <ESC><ESC> (press the ESCAPE key twice)
> STOP

HALTED AT 172347: 003776

CP> SYSCLR

*** CPU VERIFIED ***

CP> BOOT 505

TRENAME= MFD>CMDNC0>MAKE

HALTED AT 072427: 000001

CP> SYSCLR

*** CPU VERIFIED ***

CP> BOOT 170000

OK:

4. Label an available pack as the new system pack, and mount it in the system drive. Modify the label on the corrupted system pack to indicate that it is corrupted.

Strictly speaking, the old corrupted system pack is available for this purpose. However, if you use it as the new system pack, you preclude its examination by your Customer Service Representative. Such an examination may:

- Determine the cause of the corruption.
- Allow the reclamation of crucial files for which the backup copies are not sufficient.

5. Use MAKE to format and create the command disk. MAKE is described in Chapter 6. Use the command START 1000 instead of MAKE. For example:

OK: START 1000 -DISK 4460 -PARTITION ARTHUR

6. Boot MAGRST from tape, start up the disk, and attach to the MFD, as follows:

OK: <ESC><ESC> (press the ESCAPE key twice)
> STOP

HALTED AT 154012: 003776

CP> SYSCLR

*** CPU VERIFIED ***

CP> BOOT 505
TREENAME= MFD>CMDNC0>MAGRST

HALTED AT 072427: 000001

CP> SYSCLR
CP> BOOT 170000
OK: STARTUP 4460
OK: ATTACH MFD XXXXXX

7. Run MAGRST. Use the START 1000 command instead of the MAGRST command. During this procedure, you may mount and dismount more tapes. For example:

OK: START 1000

Follow the instructions for responding to MAGRST queries as described earlier in this Chapter as part of the procedure for restoring a partition (Step 4 of Figure 8-6).

8. After MAGRST has finished, dismount the last tape. This tape should be labeled "Tape n of n".
9. Shut down the system, as follows:

OK: SHUTDN
OK: <ESC><ESC> (press the ESCAPE key twice)
> STOP

HALTED AT 154012: 003776

CP>

10. Boot PRIMOS II, as follows:

CP> SYSCLR

*** CPU VERIFIED **

CP> BOOT 10114

PRIMOS II REV 19.1 11/22/82 (AT 170000)

OK:

11. Use MAKE to format and create any other corrupted disk partitions on the system pack, or on other packs. For example:

OK: MAKE -DISK 110061 -PARTITION PAGING -SPLIT 7400

12. Start up PRIMOS, but do not let users log in. (Do not issue the MAXUSR command.)

OK: PRIMOS

13. Restore all partitions that were corrupted, including the command disk partition. Use the procedure illustrated in Figure 8-6 to restore the command disk. Use appropriate restore procedures for other disks.

Note

The command disk is restored again in this step because the restoration performed in Step 7 of this procedure did not restore ACL and quota information, as it was run under PRIMOS II. Therefore, the contents of the command disk (except MAGRST itself) are deleted and restored under PRIMOS, as shown in Figure 8-6.

14. Run FIX_DISK on the command disk. This corrects the quota information. See Figure 9-1 for an illustration of this procedure. For example:

OK, FIX_DISK -DISK 4460 -FIX -COMMAND_DEVICE

15. Shut down the system. This is necessary because the system may not have been able to fully initialize itself, due to the partially restored state of the command device and other disks. For example:

```
OK, SHUTDN ALL  
REALLY? YES  
WAIT,  
PRIMOS NOT IN OPERATION
```

16. Start up PRIMOS. Users may be allowed to log in.

9

Repairing File Partitions

INTRODUCTION

This chapter discusses the `FIX_DISK` command, PRIMOS's disk repairing utility. Repairs to disks that have Rev. 19 format must be done under PRIMOS with `FIX_DISK`. Repairs to Rev. 18 partitions may be done under PRIMOS with `FIX_DISK`, or under PRIMOS II using the `FIXRAT` command. (See Chapter 14.) Use of `FIX_DISK` is recommended.

WHAT IS `FIX_DISK`?

`FIX_DISK` is an operator's command that:

- Reads every physical record in use on a disk or partition, including those in files, UFDs, and segment directories.
- Checks the quota information on Rev. 19 disks.
- Checks that the information in each record header is consistent with the UFD that contains the record.
- Checks the `DSKRAT` file for discrepancies.
- Checks ACLs.
- Checks file system pointers.

When any error is identified on a partition, an appropriate error message is displayed. See Appendix F for an explanation of each error message.

An important feature of `FIX_DISK` is its repair facility. When requested to do so, `FIX_DISK` can:

- Repair mismatched pointers.
- Correct quota information.
- Either replace defective records with new empty records or truncate files containing defective records.
- Delete defective files.
- Replace a defective `DSKRAT` file.

19.3

However, `FIX_DISK` has other uses as well. It can:

- Add new badspot information.
- Convert pre-Rev. 19 partitions to Rev. 19 partitions.
- Remap an equivalence block created by `COPY_DISK` or by `PHYRST`.
- Check a partition to see if it needs repair.

WHEN SHOULD `FIX_DISK` BE USED?

`FIX_DISK` should be run in the following situations:

- As a routine procedure during a system backup.
- If there is reason to suspect that the file structure is damaged (indicated by a warning message from `PRIMOS`).
- If there is reason to suspect that the quota system is damaged (indicated by a warning message from `PRIMOS`).
- If there is a problem with attaching to or using a file (for example, a data base).
- If a message from `COPY_DISK` or `PHYRST` indicates that an equivalence block was created.

WARNING

If a problem with the disk drive, disk pack, or controller is suspected, `FIX_DISK` should not be run until the hardware is checked. A hardware problem could cause `FIX_DISK` to incorrectly delete files and directories.

Why Run `FIX_DISK` as Part of System Backup?

System backups are an important aspect of protecting the data on your system against accidental loss. Therefore, it is equally important that all data be correctly backed up. By running `FIX_DISK` before backing up a disk, you ensure the integrity of the backup procedure itself. Several aspects of the backup procedure may depend on the integrity of the disk being backed up. By running `FIX_DISK` after backing up a disk on the backup disk when badspot handling has taken place, you reduce the amount of time it takes to recover from a loss, should the backup disk be needed for data recovery.

How Does PRIMOS Warn Me to Run `FIX_DISK`?

Whenever a disk is added to the system, PRIMOS checks to see if its file structure or quota information has been damaged. When it starts up the disk, it will print:

Starting up revision 19 partition "name".

If it finds that the file structure or quota information may be damaged, it will then print:

(Quota system may be incorrect; run `FIX_DISK`.)

Most of the time, disks are only added during system coldstart. Therefore, you should look at the output generated by the system during each coldstart to determine if `FIX_DISK` should be run.

How Will I Know if There Are Problems With a Disk?

Occasionally, users may complain about receiving the following error messages:

- Pointer mismatch found (not the same as "POINTER_FAULT\$").
- The directory is damaged.
- Directory too large.
- Bad DAM file.

- Bad truncate of segment directory.
- Segment directory error.
- The file is too long.
- Too many subdirectory levels.
- Disk format does not support this revision of PRIMOS.

When this happens, it may be desirable to run `FIX_DISK` on the disk on which the error occurred. However, if the error message "The directory is damaged" appears when referencing several different directories, the disk drive, pack, and controller should be checked before running `FIX_DISK`.

What Happens When Equivalence Blocks Are Created?

When `COPY_DISK` or `PHYRST` have finished copying to a disk on which they created one or more equivalence blocks, they print the message:

BADSPOTS HANDLED ON PARTITION `pdev`, PLEASE RUN `FIX_DISK`

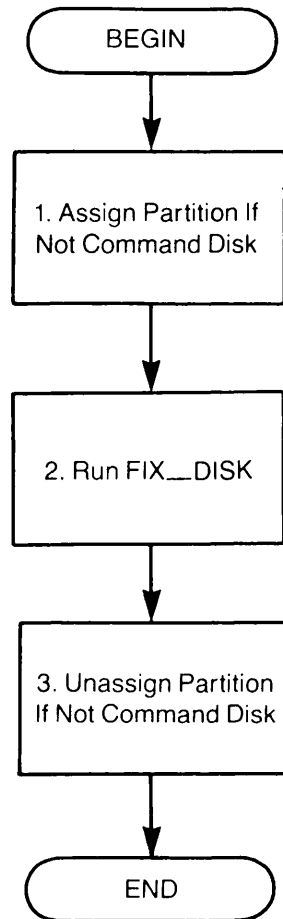
This message tells you that before you try to use the `ADDISK`, `STARTUP`, or `PHYSAV` commands on `pdev`, and before you try to copy `pdev` to another disk using `COPY_DISK`, you must run `FIX_DISK` on `pdev` using the `-FIX` option.

When you run `FIX_DISK` with the `-FIX` option on a disk that contains equivalence blocks in the `BADSPT` file, all records on the disk that point to the original (bad) records are updated so that they point to the corresponding alternate (good) records instead. The equivalence blocks are then deleted. When `FIX_DISK` is finished, the disk can be used.

WHAT TO DO BEFORE RUNNING `FIX_DISK`

Normally, `FIX_DISK` is run as part of backup procedures. When this is the case, follow the procedures described in Chapter 7 to use `FIX_DISK` in the correct manner.

Sometimes, `FIX_DISK` is run to repair a defective file structure. In this case, follow one of the system preparation procedures shown in Chapter 7. Then, perform the procedure illustrated in Figure 9-1 as part of the system preparation procedure being followed. Remember that `FIX_DISK` cannot be run under PRIMOS II.



Disk Repair Using FIX_DISK
Figure 9-1

THE PROCEDURE FOR RUNNING FIX_DISK

The procedure for running `FIX_DISK` as part of one of the system preparation procedures described in Chapter 7 is summarized in Figure 9-1. If you are repairing the command disk, perform this procedure at the supervisor terminal. Otherwise, this procedure may be performed at a user terminal.

1. If you are not repairing the command disk, assign the partition being repaired. For example:

OK, ASSIGN DISK 1060

2. Run `FIX_DISK`, as described below.

3. If you are not repairing the command disk, unassign the partition you assigned in step 1. For example:

OK, UNASSIGN DISK 1060

RUNNING FIX_DISK

`FIX_DISK` is invoked with a number of options on the command line that specify the following details of the disk repair operation:

- The physical disk (`-DISK`, `-COMMAND_DEVICE`).
- The extent of the repair operation (`-FIX`, `-UFD_COMPRESSION`, `-DUFE`, `-AUTO_TRUNCATION`, `-MAX_NESTED_LEVEL`, `-TRUNCATE`).
- The amount of terminal output (`-LEVEL`).
- The record numbers of any new badspots found (`-ADD_BADSPOT`).
- Whether badspots are to be listed (`-LIST_BADSPOTS`).
- Whether a pre-Rev. 19 format disk is to be converted to Rev. 19 format (`-CONVERT_19`).
- If a pre-Rev. 19 format disk is involved (`-NO_QUOTA`).
- If the user is to be queried, should the `DSKRAT` need repairing (`-INTERACTIVE`).

Study the following eight items and write down the options that you wish to specify, representing the desired actions. The `-DISK` option must be specified on the command line.

Then, invoke `FIX_DISK` with the list of options. `FIX_DISK` will perform its function and return you to the operating system prompt (OK,) when it is finished. As it executes, it will describe its progress and any disk problems encountered.

1. Specify the physical disk number: Use the `-DISK` option to specify on which physical disk `FIX_DISK` is to operate. (See Appendix A for information on determining physical disk numbers.) Follow the `-DISK` option with the physical disk number. The `-DISK` option and the physical disk number must be specified on the command line. For example:

```
OK, FIX_DISK -DISK 61260
```

If the specified disk is the command device (logical device 0 as indicated by a `STATUS DISKS` command), you must include the `-COMMAND_DEVICE` option on the command line, as follows:

```
OK, FIX_DISK -DISK 1060 -COMMAND_DEVICE
```

This will instruct `FIX_DISK` to automatically shut down the command device, assign it, repair it, unassign it, and start it up again. In addition, `FIX_DISK` will restore the event logging state, and re-establish any priority ACL on the command device. `FIX_DISK` does this because shutting down the command device will disable all event logging, and remove the priority ACL set on the command device.

When using `-COMMAND_DEVICE`, `FIX_DISK` must be run from the supervisor terminal. In addition, all users and phantoms should be logged out, as they rely on the availability of the command device.

If `-COMMAND_DEVICE` is not specified, `FIX_DISK` assumes that you have already shut down and assigned the disk. If you have not, the message:

```
DISK pdev NOT ASSIGNED
ER!
```

will be displayed.

Note

Do not forget to include the `-DISK` option, preceding the physical disk number on the command line. If the `-DISK` option is inadvertently omitted, or entered after the physical disk number, unpredictable behavior will occur. Should this happen, type a control-P to stop the program. If you have only assigned one disk (the one to be repaired) to your terminal, there is little chance that any harm has occurred. (The chances are greater if you have assigned two or more disks to your terminal.) Issue the `"RLS -ALL"` command, then invoke `FIX_DISK` again. Include the `-DISK` option on the command line in the correct place.

2. Specify the extent of the repair operation: If you do not want `FIX_DISK` to attempt to effect repairs, do not specify the `-FIX` option. `FIX_DISK` will report inconsistencies, but will not attempt to rectify them.

Caution

Do not use the `-FIX` option if you suspect that the disk drive itself is faulty. It is good practice to run `FIX_DISK` once without using the `-FIX` option. If no disk errors are reported, you can then run `FIX_DISK` again using the `-FIX` option. However, if `FIX_DISK` encounters disk errors, it may delete files if `-FIX` is specified on the command line.

If you do want `FIX_DISK` to make repairs, you must include the `-FIX` option on the command line, and you probably want to include the `-UFD_COMPRESSION` and `-DUFE` options as well. For example:

OK, `FIX_DISK -DISK 61260 -FIX -UFD_COMPRESSION -DUFE`

This tells `FIX_DISK` to:

- Make all needed repairs (`-FIX`).
- Compress unused space in directory records (`-UFD_COMPRESSION`).
- Delete all unrecognized file entries (`-DUFE`).

You can use the option `-ADD_BADSPOT` to add new badspots to the badspot file. For example:

OK, `FIX_DISK -DISK 61260 -FIX -ADD_BADSPOT record_1 ...`

Enter record numbers (record_1) in octal.

During operation, `FIX_DISK` may encounter directories that are nested very deeply (more than 99 levels deep). Normally, when this happens, `FIX_DISK` will abort with an error message. This allows you to consult with the owner of the offending directory tree and take appropriate action. However, if you want `FIX_DISK` to automatically delete such directories at the 99th level, include the `-AUTO_TRUNCATION` option on the command line, as follows:

OK, `FIX_DISK -DISK 61260 -FIX -AUTO_TRUNCATION`

In addition, you may change the maximum allowable number of nested levels by including the `-MAX_NESTED_LEVEL` option on the command line. If `FIX_DISK` encounters a directory nesting greater than the value you specify, `FIX_DISK` will abort, or, if `-AUTO_TRUNCATION` was specified, `FIX_DISK` will delete the directories that exceed the maximum nesting level.

For example:

OK, `FIX_DISK -DISK 61260 -MAX_NESTED_LEVEL 20`

Here, the maximum level is 20. If `-MAX_NESTED_LEVEL` is not specified, the maximum level defaults to 99.

3. Specify the disposition of files containing badspots: When `FIX_DISK` finds a record it cannot read, it will create a null record (a record filled with zeros) on a good portion of the disk and append the remaining records of the original file to the null record. If more than one sequential record is missing, and the file is a SAM file, then two null records are created and an error message is output indicating that the correct number of records is not known. If you prefer to have `FIX_DISK` truncate a file when it encounters a badspot (as it did prior to Rev. 19.3), include the `-TRUNCATE` option on the command line as follows:

19.3

OK, `FIX_DISK -DISK 61260 -FIX -TRUNCATE`

4. Specify the amount of terminal output: Normally, while `FIX_DISK` is running, it states the name of the UFD being processed, as follows:

BEGIN CLEOPATRA

Then, when it finishes that UFD, it prints:

END CLEOPATRA recs

Here, recs is the total number of 2048-byte records in use by the UFD.

`FIX_DISK` starts the output with `BEGIN MFD` and `END MFD recs`, since all UFDs have the MFD as their parent directory. Because listing all of the UFDs on the disk may take time, you may wish to disable this feature. To do this, include the `-LEVEL 0` option on the command line, as follows:

OK, `FIX_DISK -DISK 61260 -LEVEL 0`

When you use this option as shown, `FIX_DISK` will output the `BEGIN MFD` and `END MFD` messages, but will not output any other `BEGIN UFD` or `END UFD` messages.

If `-LEVEL` is not included on the command line, the default is `-LEVEL 1`. This means to only output top-level directories in the MFD via `BEGIN` and `END` messages. Specifying a higher `-LEVEL` value than 1 causes `BEGIN` and `END` messages to be output for top-level UFDs, their sub-UFDs, and so on, to the specified level. (To assist you in understanding the UFD structure on a disk, `FIX_DISK` indents the `BEGIN` and `END` messages by two spaces for each level.)

Note

Do not confuse the `-LEVEL` option with the `-MAX_NESTED_LEVEL` option. `-LEVEL` affects how much information is printed at your terminal, but does not otherwise affect `FIX_DISK`.

5. List badspots: Normally, the only information `FIX_DISK` prints concerning badspots is the number of lost records they represent. If you would like to see more information on badspots, include the `-LIST_BADSPOTS` option on the command line, as follows:

OK, `FIX_DISK -DISK 1060 -LIST_BADSPOTS`

This will cause `FIX_DISK` to list all the known badspots on the disk, followed by any equivalence blocks. This list will be printed before the `BEGIN` MFD message. For each badspot, the record number is identified. For each equivalence block, the record number of both the badspot and the re-mapped (alternate) record are listed. In addition to the record number, the track, head, and sector numbers are also listed for badspots and equivalence blocks.

6. Convert to Rev. 19 format: To convert a pre-Rev. 19 format disk to Rev. 19 format, include the `-CONVERT_19` and `-FIX` options on the command line, as follows:

OK, `FIX_DISK -DISK 1060 -FIX -CONVERT_19`

The `BADSPT` file, if any, will be converted, and all quota information will be initialized.

7. Do not convert to Rev. 19 format: To run `FIX_DISK` on a pre-Rev. 19 format disk without converting it to Rev. 19 format, include the `-NO_QUOTA` option on the command line, as follows:

OK, `FIX_DISK -DISK 1060 -NO_QUOTA`

This disabling of quota checking is necessary because pre-Rev. 19 disks have no quota information.

8. Repair defective or missing DSKRAT: If you have already attempted to run `FIX_DISK`, it may have issued one of the following error messages:

- The file structure of DSKRAT is bad.
- The number of heads is different.
It should be YY is XX
- The physical record size is different.
It should be YY is XX
- The DSKRAT header has the wrong length.
It should be YY is XX
- The partition cannot be handled by this version of `FIX_DISK`.

19.3

If one of these messages appears, re-run `FIX_DISK`, and include the `-INTERACTIVE` and `-FIX` options on the command line, as follows:

OK, `FIX_DISK -DISK 61260 -FIX -INTERACTIVE`

This will cause `FIX_DISK` to ask you questions when it discovers a defective or missing DSKRAT, rather than abort. The answers you supply enable it to construct a correct DSKRAT.

For more information on `FIX_DISK` options and their abbreviations, see Chapter 13.

WHAT TO DO AFTER RUNNING `FIX_DISK`

After you run `FIX_DISK`, continue with the system preparation procedure.

Note

If `FIX_DISK` has been run with the `-COMMAND_DEVICE` option, it is advisable to cold-start the system. Until the system has been cold-started, the supervisor terminal will have no origin directory. In this situation, giving the `ORIGIN` command at the supervisor terminal will produce an error message instead of attaching the supervisor terminal to its origin directory, `CMDNC0`.

EXAMPLES OF FIX_DISK

An example of FIX_DISK follows. Here, the partition BEEBLE is specified by its physical device number (22660), 0 levels of directories are to be printed, necessary disk modifications are to be made, and UFDs are to be compressed.

```
OK, FIX_DISK -DISK 22660 -LEVEL 0 -UFD_COMPRESSION -FIX
***FIX_DISK*** <Rev 19.3> 11/12/83 18:29
```

```
Partition name is BEEBLE
```

```
BEGIN MFD
END MFD 29404
```

```
74070 records in partition
29404 records used
    0 records lost
44666 records left
    0 records compressed
DSKRAT UPDATED!
FIX_DISK finished
OK,
```

FIX_DISK has identified the partition name; summarized crucial data concerning record usage; checked DSKRAT and found it to be OK; and signaled the completion of its operation. No errors of any sort are indicated. No compression took place.

Note

The line "records lost" refers to record loss caused by badspots. It does not indicate file truncation or record compression.

It is quite possible, however, that FIX_DISK will find problems on the partition it examines. Such problems would be indicated by error messages. A complete listing of FIX_DISK's error messages is located in Appendix F.

The next two examples illustrate FIX_DISK's error handling technique. In the first example, 0 levels of directories will be printed.

```
OK, FIX_DISK -DISK 1060 LEVEL 0 -UFD_COMPRESSION
***FIX_DISK*** <Rev 19.3> 12/20/83 14:21
```

Partition name is ZAPHOD

BEGIN MFD

ACL at word 513 does not point at a file or access category!

ACL is deleted!

ACL at word 4044 does not point at a file or access category!

ACL is deleted!

Access Category BOOKS.ACAT does not reference an ACL!

Access Category is deleted!

File CMDNC0 does not reference an ACL or Access Category!

Changed to default ACL pointer

File CHAPTER does not reference an ACL or Access Category!

Changed to default ACL pointer

File LATE does not reference an ACL or Access Category!

Changed to default ACL pointer

File RUNOFF does not reference an ACL or Access Category!

Changed to default ACL pointer

MFD

The Directory Used count is bad. It should be 67 instead of 71.

MFD>RUNOFF

END MFD 29404

74070 records in partition

29404 records used

0 records lost

44666 records left

1 records compressed

DSKRAT UPDATED!

FIX_DISK finished

OK,

19.3

In the above example, disk 1060 is identified as ZAPHOD; 2 ACLs in the MFD were deleted; the access category BOOKS.ACAT was deleted; four files were changed to the default ACL category; the Directory Used count in RUNOFF was found to be in error and was corrected; one record was freed by compression of UFDs; DSKRAT was updated; operation was returned to PRIMOS.

In the next example, FIX_DISK explores the entire disk and prints to level 9 (the limit specified by -LEVEL). The use of the option -CONVERT_19 indicates that FIX_DISK is to change this disk to the Rev. 19 format.

OK, FIX_DISK -DISK 460 -LEVEL 9 -CONVERT 19 -UFD_COMPRESSION -FIX
 FIX_DISK <REV 19.3> 12/21/83 13:23

Partition name is UNICRN

The quota system may be incorrect.

BEGIN MFD

BEGIN CMDNC0

BEGIN RUN

END RUN 8

BEGIN SYS.SIGNOFF.SEG

END SYS.SIGNOFF.SEG 14

END CMDNC0 3536

BEGIN OEDPUS

The father pointer is bad.

It should be 5332 is 5335

Bad record address = 53340 BRA = 5340 Father = 5332 Type = 0

File is deleted!

MFD>OEDPUS>ATE>0

Zero record added to file.

MFD>OEDPUS>ATE

19.3

The Directory Used count is bad. It should be 14 instead of 147.

MFD>OEDPUS

END OEDPUS 85

BEGIN EUMEN

END EUMEN 1001

BEGIN COEPH

END COEPH 8907

BEGIN CLNNOS

BEGIN HYBRIS

BEGIN CTHRSYS

END CTHRSYS 16

END HYBRIS 99

END CLNNOS 650

END MFD 13527

14814 records in partition

13527 records used

0 records lost

1287 records left

1 records compressed

DSKRAT UPDATED!

FIX_DISK finished

OK,

In the example above, FIX_DISK has tried to examine the MFD and all the UFDs. Terminal output was set to level 9. An error was located in the UFD OEDPUS; the correction was made, adding a record of zeros to the file, and reported; a record was freed by compressing a UFD, thus removing empty space on the partition; record information was printed; and the disk's record availability table (DSKRAT) was updated. The disk was converted to the Rev. 19 format.

19.3

On a single-partition system, running FIX_DISK with the -COMDEV option would look like this:

```
OK, FIX_DISK -DISK 460 -UFD_COMPRESSION -FIX -COMMAND_DEVICE
***FIX_DISK*** <REV 19.0> 11/02/82 0:20
```

Partition name is UNICRN

BEGIN MFD

ACL at word 733 does not point at a file or access category!

ACL is deleted!

File SPOOLQ does not reference an ACL or Access Category!

Changed to default ACL pointer

BEGIN SAD

END SAD 19

BEGIN CMDNC0

END CMDNC0 1234

BEGIN DOS

END DOS 21

BEGIN LM

END LM 26

BEGIN SYSCOM

END SYSCOM 164

BEGIN LIB7

END LIB7 83

BEGIN FAM

END FAM 94

BEGIN XXX

END XXX 243

BEGIN SYSOVL

END SYSOVL 151

BEGIN BATCHQ

END BATCHQ 78

BEGIN PRIMENET*

END PRIMENET* 67

BEGIN SYSTEM

END SYSTEM 1083

BEGIN EMACS*

END EMACS* 1

BEGIN LOGREC*

END LOGREC* 74

BEGIN LIB

END LIB 634

BEGIN SEGRUN*

END SEGRUN* 252

BEGIN OPSYS

END OPSYS 667

```

BEGIN DIRECTV
END DIRECTV 33
BEGIN SPOOLQ
END SPOOLQ 130
END MFD 5157

```

```

14814 records in partition
5157 records used
0 records lost
9657 records used
0 records compressed
DSKRAT OK
FIX_DISK finished
Starting up revision 19 partition "UNICRN".
OK,

```

Note that the partition is automatically started at the conclusion of FIX_DISK.

In the next example, FIX_DISK is run with the -LIST_BADSPOTS option, to show the badspots on the disk. FIX_DISK is then run with the -ADD_BADSPOT and -FIX options to add a new badspot:

```

OK, ASSIGN DISK 1066
OK, FIX_DISK -DISK 1066 -LIST_BADSPOTS
***FIX_DISK*** <Rev 19.3> 12/20/83 16:49

```

Partition name is GOLD

The badspot section has 1 entries.

Track = 290 Head = 2 Sector = 3 Bad record = '24335.

```

BEGIN MFD
  BEGIN CMDNC0
  END CMDNC0 1
  BEGIN DOS
  END DOS 1
END MFD 8

```

```

29628 records in partition
8 records used
1 records lost
29619 records left
0 records compressed

```

```

DSKRAT MISMATCH!
FIX_DISK finished
OK, FIX_DISK -DISK 1066 -ADD_BADSPOT 22222 -FIX
***FIX_DISK*** <Rev 19.3> 12/20/83 16:51

```

Partition name is GOLD

The badspot section has 1 entries.

```

BEGIN MFD

```

19.3

```

BEGIN CMDNCO
END  CMDNCO  1
BEGIN DOS
END  DOS  1
Processing add badspot request for '22222
Added to BADSPT file!
END  MFD  8

```

```

29628 records in partition
      8 records used
      2 records lost
29618 records left
      0 records compressed
DSKRAT UPDATED!
FIX_DISK finished
OK,

```

In the above example, FIX_DISK first listed the badspot '24335 that already existed in the file BADSPT. It checked the DSKRAT file and found that it did not agree with the record allocation information generated by FIX_DISK. FIX_DISK was then invoked again, with the -FIX and -ADD_BADSPOT options specified. On this second pass the badspot information and the DSKRAT file were updated.

In the following example, FIX_DISK is invoked with the -TRUNCATE and -NUMBER_OF_RETRIES options. FIX_DISK will truncate any bad records found, but will try four times (rather than the default of two) to read any bad records found.

19.3

```

OK, ASSIGN DISK 1066
OK, FIX_DISK -DISK 1066 -TRUNCATE -NUMBER_OF_RETRIES 4 -FIX
***FIX_DISK*** <Rev 19.3> 12/20/83 17:20

```

Partition name is GOLD

The badspot section has 2 entries.

```

BEGIN MFD
  BEGIN CMDNCO
    END  CMDNCO  1
  BEGIN DOS
    END  DOS  1
END  MFD  8

29628 records in partition
      8 records used
      2 records lost
29618 records left
      0 records compressed
DSKRAT OK
FIX_DISK finished
OK, UNASSIGN DISK 1066
OK,

```

HOW DOES FIX_DISK WORK?

Locating DSKRAT Errors: In order to read each physical record currently in use, FIX_DISK must traverse the entire logical file structure of a partition. While doing this, it creates its own record availability table (RAT) which is updated and checked against the pre-existing disk record availability table (DSKRAT) for each record. If a discrepancy is noted in comparing FIX_DISK's RAT to the DSKRAT, an error message is displayed at the operator's terminal.

Locating Quota Errors: On Rev. 19 format disks (disks on which the quota information is maintained), FIX_DISK compares the directory- and tree-used counts against the actual number of records within the UFD. An error message is generated if there is an error in the UFD's record of the number of records that have been used.

Converting a Partition to the Rev. 19 Format: To make an old partition conform to Rev. 19 standards, FIX_DISK does three things: it initializes the quota information; it changes the current badspot file to the Rev. 19 format; it creates a Rev. 19 stamp. Such conversion is done through the use of FIX_DISK's -CONVERT_19 option.

19.3 |

Handling the BADSPT File: All badspots encountered are added to the badspot file if it exists. If it does not exist, a new badspot file will be created if there is room on the disk. If FIX_DISK encounters an equivalence section in a disk's BADSPT file, FIX_DISK maps bad records to their equivalence records and then sets the file system pointers to those records. When the file system structure has been completely traversed, the equivalence section is deleted.

10

Printers and Plotters

INTRODUCTION

This chapter focuses on the use of printers and plotters. The most important command associated with the control of printers is PROP. This utility allows the operator to control printer operations in two ways: by defining and modifying the printer environment; and by starting, stopping, and monitoring the processes that service the printer environments.

This chapter explains:

- Printer environments
- The PROP command
- How to deal with mechanical problems

PRINTER ENVIRONMENTS

When a user uses the SPOOL command to print a file, the spooler subsystem needs to determine which printer or collection of printers are acceptable for the file. To coordinate information about printers with information on user print requests, the spooler subsystem uses printer environments.

This section explains:

- What is a printer environment?
- What is a spooler phantom?
- What is the SPOOLQ directory?
- How do environments work?

What Is a Printer Environment?

A printer environment is a named collection of all the decisions made concerning a printer or plotter. For example:

- To what printer or plotter does the environment apply?
- What type of paper is mounted on the printer/plotter?
- What is the maximum file size that can be printed?
- How many lines per page should be printed?
- How many columns per line should be printed?
- Should a COMOUTPUT file be kept of printer/plotter activities?

The value of the printer environment is that such decisions may be made in advance, rather than being made each time a printer is started up.

Your installation may have only one printer, but you might have several different kinds of paper that can be mounted on it at various times. There may be times (such as working hours) when you want to limit the size of files that are printed, and other times (such as off hours) when you wish to impose no limits at all. An installation that has two printers may wish to send all large files to the faster printer. Because potentially complex situations arise, users do not spool files to specific printers or plotters. Instead, they specify the destination and desired form type for their print requests, using the -AT and -FORM options of the SPOOL command.

Several printers may recognize a particular combination of destination and form names. In this case, the printer first ready to print a particular file will start. Or, a user can spool a file with a combination of destination and form names that is not at that time recognized by any printer. In this case, the file cannot be printed until a printer is reconfigured to recognize that combination -- for example, by mounting a new form type on the printer.

Other information is needed by the system to send files to specific printers or plotters. To keep all of this information straight, and to inform printers of what files they may print or plot, printer

environments are used. Information on a printer environment is stored in a file named E.environment in the SPOOLQ UFD. Usually several printer environments are created for each printer, so that an environment will be on record for most of the possible printer configurations. Then, when a printer is started or when paper is changed, the appropriate environment can be invoked quickly and easily.

At any time, the operator (or any user) can obtain a list of all printer environments defined on the system, including which ones are being serviced by a spooler phantom. The command used is:

PROP -STATUS

In addition, the operator (or any user) can see how any printer environment is set up by using the command:

PROP environment -DISPLAY

Figure 10-1 shows the results of using the DISPLAY command when a default environment (that is, an environment composed entirely of default values) is constructed using the -CREATE option.

```
OK, PROP DEFAULT -CREATE
> DISPLAY
[PROP rev 19.0]

DEVICE: PRO
PAPER:
MESSAGE:

COMOUT: OFF
UPCASE: OFF
PRINT:  ON
PLOT:   OFF
EVFU:   OFF
TYPE:           0
LENGTH:        38
LARGE:         20
LIMIT          off
UPPER:         63
LOWER:         0
HEADER:        1
WIDTH:        108
LINES:         off

> QUIT
OK,
```

Illustration of the Default PROP Environment
Figure 10-1

The commands that create and modify these printer environments are explained later in this chapter.

To start up a printer so that it can begin printing users' files, a spooler phantom is started up by issuing the command:

```
PROP environment -START
```

This starts up a spooler phantom that uses the specified environment to determine what files to print, on which printer.

What Is a Spooler Phantom?

A phantom is a CPL or command input file that runs as a process without being attached to a terminal. A spooler phantom is one that controls the activities of a single printer or plotter. Each spooler phantom runs the program SPOOLQ>SPPHN.SEG based on the information in one particular printer environment. A spooler phantom is then said to be servicing a printer environment.

The environment tells the phantom which printing device to drive and what kinds of files the device may handle. Once started, the phantom repeatedly searches spool queues for files it can handle. When such files are located, the phantom prints or plots them.

The phantom program, the printer environments, the queue file, and the print files are all located in the SPOOLQ UFD.

What Is the SPOOLQ Directory?

The spooler uses the SPOOLQ UFD to store files in the print queue. If multiple systems are connected over a network, each system should have its own SPOOLQ directory, located on one of its own disk partitions.

The SPOOLQ UFD contains two types of files of specific interest at this point: spool queue files and printer environment-related files.

Spool Queue Files: A spool queue file holds the contents of a print request in the spool queue. One such file exists for each request in the queue. The file is placed in the directory by the spooler whenever a user inputs the SPOOL command to print a file. When the file has been printed, the spool queue file is eliminated, thus removing the file from the spool queue. The filename for such a file is PRTnnn, where nnn represents the file's position on the queue at the time the SPOOL request was invoked.

Here is a sample SPOOLQ directory. At the time of listing, it contains four spool queue files.

<SYSTPB>SPOOLQ (Owner)
229 records in this directory, 229 total records out of quota of 0.

20 Files.

CRDSP.L.SAVE	E.TBPR0	E.TP.DBL	E.TP.NPR
L.DEST	L.DFLT	L.FORM	PRT003
NEW_FMT2.CPL	O.TBPR0	O.TE.NPR	O.TP.DBL
O.TP.NPR	O_TBPR0	O_TP.DBL	O_TP.NPR
PRT005	PRT008	PRT009	Q.CTRL

1 Segment Directory.

SPPHN.SEG

Environment-related Files: Some environment-related files are created by the operator. Others are created by the spooler subsystem. The environment-related files themselves are of little concern to the operator, as they are created, modified, and deleted by the PROP command and the phantoms that service the environments. The exception is the O_environment file described below. This file should be archived regularly, as it tends to grow without bound.

Each running printer or plotter has its own phantom, and hence, its own environment file. In fact, a single device may have several potential environments stored as files in the SPOOLQ UFD.

Filenames that affect spooler operation will always contain the name of the environment plus a prefix that specifies the role of the file in the spooler's operation. Environment files are named according to the following pattern:

<u>Filename</u>	<u>Function</u>
O_environment	Contains a COMOUTPUT file for the phantom that services <u>environment</u> , listing all files spooled.
E.environment	Contains the environment definition for the phantom that services <u>environment</u> .
A.environment	Contains the last operator request received by the phantom that services <u>environment</u> . If <u>A.environment</u> is open, it means that the request has been received but not yet acted on.
O.environment	Contains the COMINPUT file created by PROP for the phantom that services <u>environment</u> . Indicates that <u>environment</u> is being serviced by a phantom.

R.environment Contains all operator requests that have not yet been noticed by the phantom that services environment.

Caution

Do not attempt to edit any of the above files except O_environment. They contain binary, not textual, data. If an attempt is made to edit them, PROP may be unable to perform any operations on the printer environment whose file you edited, and the phantom servicing that environment may log itself out. To remedy this situation should it occur, type the command:

```
DELETE SPOOLQ>+.environment -NO_QUERY -NO_VERIFY
```

Here, environment is the name of the printer environment whose file was accidentally edited. This will delete the environment even when PROP is unable to delete it. Then use PROP to recreate the environment:

```
PROP environment -CREATE
```

Specify the parameters of the environment as they were before the accident. The environment will now be ready for use again.

The sample listing of SPOOLQ earlier in this section contains two environments, named PRO and PRI. Both are running, and both have COMOUTPUT files.

Other Environment-related files: In addition to the files mentioned above, the System Administrator may create three files to define allowable form and destination names for the system. These files, which appear in the sample directory shown above, are as follows:

<u>Filename</u>	<u>Function</u>
L.DEST	This file specifies all allowable synonyms for the SPOOL -AT option (such as FLOOR1, LAB, LOBBY, etc.). It contains one destination name per line, each destination containing a maximum of 16 characters, uppercase only. If this file is absent, any destination name may be specified by the user.
L.DFLT	This file contains one destination name, with a maximum of 16 characters, that becomes the default destination for the SPOOL -AT option (such as LOBBY). If this file does not exist, the default destination is blank.

L.FORM This file specifies all allowable synonyms for the SPOOL -FORM option (such as RUSH, WHITE, WIDE, etc.). It contains one form type per line, each line having a maximum of six characters, in uppercase only. If this file is absent, any form name is legal.

The Q.CTRL File: A control file named Q.CTRL, used by the queuing mechanism, also is in this UFD. The queue structure permits a maximum of 200 entries. Any attempt to exceed this will yield an error message from the spool program. Because the SPOOL command always adds new entries in FIFO (First In First Out) order, the queue may become full even when fewer than 200 entries are in it. However, this is a rare occurrence.

Note

If users complain that they receive a "File in use" message whenever they try to use the SPOOL command to spool a file, it may be that the Q.CTRL file in SPOOLQ is unavailable. To remedy this situation, go to the supervisor terminal, and issue the following command:

```
CLOSE SPOOLQ>Q.CTRL
```

Other files: Other files may be added to the directory SPOOLQ as the System Administrator or operator desires. The file NETWORK_INFORMATION.SPOOL in the SPOOLQ listing shown above is an example of this. This is a standard text file, containing network information that will be referenced when this system's spool queue is addressed by another system in the network.

The file contains two lines. Line one indicates the name of the system on which the SPOOLQ UFD resides. Line two designates the version of the spooler running on this system. For example:

```
SYS.B
REV.19
```

This file and others like it are created with the editor (ED). If you are not familiar with ED, see the New User's Guide to EDITOR and RUNOFF.

How Environments Work: An Example

The required number of environments can be set up and kept on file in the SPOOLQ UFD for each printer. For example, printer PR0 might have four environments on file: environment PR0, which uses standard paper and a standard printout format; environment WHITE, which uses white paper; environment SHIP, which prints shipping labels on special forms; and environment RUSH, which uses standard paper and prints only very short jobs, thus acting as a quick-service queue.

The environment for a certain printer may be switched as needed. For example, after starting up the system, the operator may:

1. Power up printer PR0.
2. Mount white paper.
3. Give the command `PROP WHITE -START`.

The phantom servicing the environment named WHITE then controls printer PR0. Later on, seeing the need to use printer PR0 to print some "rush jobs", the operator may:

1. Give the command `PROP WHITE -STOP FINISH`.
2. When the printer has printed the message STOPPING, mount standard paper.
3. Give the command `PROP RUSH -START`.

The printer is now running under the control of the phantom servicing the environment named RUSH.

THE PROP COMMAND

The PROP command allows you to perform the following common operations:

- Determine what printer environments exist on the system, and which ones are currently being serviced by phantoms (`-STATUS`).
- Display the configuration of a printer environment (`-DISPLAY`).
- Modify an existing printer environment, even when the environment is being serviced by a phantom (`-MODIFY`).
- Start up a phantom to service a phantom environment (`-START`).
- Shut down a phantom that is servicing a phantom environment (`-STOP`).
- Temporarily suspend the activities of a phantom that is servicing a printer environment (`-HANG`).

- Continue a suspended phantom (-CONTINUE).
- Recover following a printer jam (or similar) by instructing the phantom that is servicing a printer environment to reprint part or all of a file (-RESTART, -BACK, -ABORT).
- Enlist the assistance of the phantom that is servicing a printer environment in ensuring correct alignment of forms (-LINEUP).

Except for -STATUS and -DISPLAY, all of the options accepted by PROP require the user to be logged in as SYSTEM, or as the user that originally created the specified environment. (User SYSTEM can perform operator functions on any environment.) See Chapter 13 for a complete list of PROP options.

Determining Printer Environment Information

As described earlier in this chapter, the -STATUS and -DISPLAY options of PROP allow both the operator and all users to determine what environments are defined, and how they are configured.

PROP -STATUS

This will output a list of defined environments. For each listed environment name, "stopped" is output if no phantom is servicing the environment, or "started" is output if there is a phantom currently servicing the environment.

PROP environment -DISPLAY

This will output all of the configuration information on the specified environment. The first line of the information reads:

DEVICE: dev

Here, dev is the PRIMOS device name of the printer or plotter used by this environment. Example device names include PR0, PLOT, CENPR, AMLC10.

STATUS USERS

This will output a list of all users on the system, including all running phantoms. Phantoms that are servicing printer environments will be listed. You can recognize them by looking under the Devices column; a phantom servicing an environment will show DEVICE: in a PROP -DISPLAY of that environment.

Modifying an Environment

Spooler environments are modified by using the -MODIFY option. The -MODIFY option makes changes to an existing environment. The modified environment takes effect immediately if no phantom is servicing the environment, or later, depending on the modify option.

For example, to modify the environment named SAMPLE, enter:

```
PROP SAMPLE -MODIFY
```

PROP will read in the specified environment, and enter PROP's environment definition mode. PROP prints an angle prompt (>) and waits for subcommands. A complete description of all subcommands is given in Chapter 13.

All subcommands except DISPLAY, QUIT, and FILE define some parameter of the environment. DISPLAY shows the environment as it currently stands, but does not leave Environment Definition mode. QUIT leaves Environment Definition mode without updating the new environment.

To update the environment, enter the FILE subcommand. If no phantom is servicing the environment, the new environment will take effect immediately, and PROP will exit to PRIMOS. If a phantom is servicing the environment, PROP will send the updated environment to the phantom. While waiting for the phantom to acknowledge receipt of the new environment, PROP will type the message:

Wait...

When the phantom acknowledges receipt of the new environment, PROP will type the message:

Acknowledged.

If the phantom is printing a file, the new environment will not actually take effect until it finishes printing the file. (To cause the phantom to adopt the new environment immediately, or not until the phantom has no more files to print, see Chapter 13.) If the phantom is not printing a file, the new environment will take effect immediately. In either case, when the new environment takes effect, the following message will be sent to your terminal:

```
*** SYSTEM (user nn on sysname) at hh:mm
*SPOOL* environment    Starting
```

Occasionally, you may also need to create a new environment. See Chapter 13 for information on the -CREATE option. It is quite similar to the -MODIFY option.

Figure 10-2 shows a terminal session which creates an environment. See Figure 10-3 for an example of a terminal session which modifies an environment.

OK, PROP SAMPLE -CREATE
[PROP rev 19.0]

> PAPER WHITE
> LENGTH 54
> WIDTH 88
> LINES 66
> FORM WH
> FORM DOC
> FORM RREPORT
> DEST SALES
> DEST MARKETING
> MESSAGE This is a sample environment.
> DISPLAY

DEVICE: PRO

PAPER: WHITE

FORM:

WH

DOC

RREPORT

DEST:

SALES

MARKETING

MESSAGE:

This is a sample environment.

COMOUT: OFF

UPCASE: OFF

PRINT: ON

PLOT: OFF

EVFU: OFF

TYPE: 0

LENGTH: 54

LARGE: 20

LIMIT: off

UPPER: 63

LOWER: 0

HEADER: 1

WIDTH: 88

LINES: 66

> FILE

OK,

Creating a PROP Environment
Figure 10-2

OK, PROP SAMPLE -MODIFY
[PROP rev 19.0]

> UNFORM RIPORT
 > FORM REPORT
 > FORM WIDE
 > UNFORM WH
 > FORM WHITE
 > LARGE 35
 > HEADER 2
 > DEST FRONT.OFFICE
 > EVFU -ON
 > COMO ON
 > MESS This is a modified sample environment.
 > DISPLAY

DEVICE: PRO

PAPER: WHITE

FORM:

WHITE

DOC

REPORT

WIDE

DEST:

SALES

MKT

FRONT.OFFICE

MESSAGE:

This is a modified sample environment.

COMOUT: ON

UPCASE: OFF

PRINT: ON

PLOT: OFF

EVFU: ON

TYPE: 0

LENGTH: 54

LARGE: 35

LIMIT: off

UPPER: 63

LOWER: 0

HEADER: 2

WIDTH: 88

LINES: 66

> FILE

OK,

Modifying a PROP Environment
 Figure 10-3

Synonyms: The phantom environment may contain synonyms for both the form type (the PAPER and FORM subcommands) and the printer name (the DEST subcommand). Both map to the same name at the phantom level. Thus, when the SPOOL command is issued by a user, SPOOL's -FORM and -AT options may specify parameters that are not identical to the values currently expected by the printer environment. In this case, the phantom servicing the environment will not attempt to print a file with those parameters.

The phantom matches SPOOL's -FORM name with the environment's PAPER and FORM names, and matches SPOOL's -AT name with the environment's DEST names and the environment's own name. For example, to define an environment that runs on WHITE forms, and accepts the form synonyms REPORT, DOC, and 8_X_11, you would enter the following subcommands:

```
PAPER WHITE
FORM REPORT
FORM DOC
FORM 8_X_11
```

Then, any request with any of those names after the spooler -FORM option is eligible for printing by the phantom servicing the environment. For example:

```
OK, SPOOL TESSERACT -FORM 8_X_11
```

Assuming the default -AT destination name is recognized by the environment, the file TESSERACT will print on the printer with WHITE forms mounted. The only noticeable difference between specifying the above and specifying -FORM WHITE is that the word 8_X_11 will be found somewhere on the header page.

Limiting File Size: During the day, you may wish to limit the size of files printed. After the peak hours, you would then remove any such limits. To place a limit on the size of files that can be printed by a phantom servicing the environment you are modifying, enter the subcommand:

```
LIMIT recs
```

Specify the maximum number of 2048-byte records allowed in recs. Later, to remove the limit, you would again modify the environment, and enter the subcommand:

```
LIMIT OFF
```

While the limit is imposed, users may complain that their files are not being printed. If this happens, check the size of their print files using SPOOL -LIST — if the size exceeds the limit, then the phantom will not print the file until you increase or remove the limit.

The Electronic Vertical Format Unit: Some 300 lpm printer/plotters and band printers now define a form's length with the Electronic Vertical Format Unit rather than the forms-length switch and paper tape loop. The procedure for adding an EVFU to the phantom environment is as follows:

- When creating a PROP environment for 300 lpm printer/plotters, set form length with the following subcommands:

```
EVFU -ON
TYPE 0
```

- When creating a PROP environment for band printers, set form length with the following subcommands:

```
EVFU -ON
TYPE 1
DEVICE PR0 (or PR1, PR2, PR3)
```

- In cases where an EVFU file has been established in advance, two steps are necessary: make sure the LINES parameter is set equal to the number of lines in the EVFU file, and issue the subcommand:

```
EVFU -NAME filename
```

In most cases, all necessary EVFU files will have been established in advance by the System Administrator. If not, a special EVFU file must be constructed. Instructions for building such a file are located in the System Administrator's Guide.

Notes

If the printer loses power, the EVFU must be reloaded by stopping and restarting the printer using PROP.

Align the top of the form at the top of the paper before the environment is started with PROP.

It is particularly important to use correct paper forms when using EVFU. Accompany the SPOOL command with the correct -FORM option.

Prime's Office Automation System and CAD/CAM System use a modified Spooler subsystem. If you are configuring an environment for a Letter Quality Printer on which Office Automation documents will be spooled, see the Office Automation System Administrators Guide discussion of spooler environments. See the Prime Plotter Software Guide if you are configuring a printer for CAD/CAM usage.

Starting Up a Spooler Phantom

To start up a phantom that will service a printer environment, issue the command:

```
PROP environment -START
```

After a short while, you will receive the message:

```
*** SYSTEM (user nn on sysname) at hh:mm  
*SPOOL* environment    Starting
```

Shutting Down a Spooler Phantom

To shut down a phantom that is servicing a particular printer environment, issue the command:

```
PROP environment -STOP
```

After the phantom finishes printing whatever file it may be printing, you will receive the message:

```
*** SYSTEM (user nn on sysname) at hh:mm  
*SPOOL* environment    — STOPPING
```

If you don't want the phantom to finish printing the current file before shutting down, issue this command:

```
PROP environment -STOP NOW
```

The currently printing file will be left in the queue, so it will be printed later.

Suspending and Continuing a Spooler Phantom

To temporarily suspend a phantom that is servicing a particular printer environment, issue the command:

```
PROP environment -HANG
```

After the phantom finishes printing whatever file it may be printing, you will receive the message:

```
*** SYSTEM (user nn on sysname) at hh:mm  
*SPOOL* environment    Hanging
```

Then, to cause the phantom to continue normal operation, issue the command:

```
PROP environment -CONTINUE
```

The phantom will immediately resume scanning the queue, and you will receive no further message.

Recovering After a Printer Jam

If the printer runs out of paper or ribbon, you may be able to remedy the situation without causing part of the file being printed to be lost or rendered unreadable. Then you may simply fix the problem and allow the printer to continue, without using the PROP command; the phantom will automatically continue printing where it left off.

This procedure is undesirable if part of the printed output has been lost or is unreadable. In this latter case, perform the following steps:

1. Remedy the problem with the printer, but do not start up the printer again.

2. Issue the command:

PROP environment -RESTART

PROP will display:

Wait...

3. Within 30 seconds, start up the printer again.

4. PROP should display:

Wait... Acknowledged.

At almost the same time, the message:

*** SYSTEM (user nn on sysname) at hh:mm
SPOOL environment --RESTARTING

will be displayed. (It may even come out before the "Acknowledged" message.)

PROP may instead print:

Wait... the request timed out and has been deleted.

Check to make sure that the printer is powered-on and set online. Then, reissue the PROP -RESTART command shown in step 2.

Instead of -RESTART, you may use -ABORT. With this command, the current file will be reprinted, but perhaps not right away. The phantom may decide to print other (perhaps smaller) files. Another

option is -BACK. It is useful when a problem occurs after a large amount of the file has been already printed. By specifying -BACK instead of -RESTART, you cause the phantom to restart the file anywhere from 128 to 256 lines before the jam in the file. The file will probably not be restarted on a page boundary, so use of the -BACK option is only recommended when the file is large and a fair amount of it has already been printed.

Aligning Forms

PROP has an option that helps you align the paper to the top of the page. To use it, first make sure there is a file being printed by the printer. (You may have to spool one yourself.) Follow this procedure:

1. Issue the command:

PROP environment -LINEUP

2. Wait until you receive the message:

*** SYSTEM (user nn on sysname) at hh:mm
SPOOL environment Hanging

If it doesn't appear after a while, spool a small file to the printer.

3. The printer will have printed four lines at the top of a page. Inspect the paper alignment in relation to the printed lines.
4. If the alignment is correct, go to step 7. Otherwise, proceed with step 5.
5. Align the paper, approaching the correct alignment.
6. Check the alignment by repeating step 1. This will cause the printer to reprint the same four lines at the top of the next page. Inspect the alignment again. Continue this procedure until you have properly aligned the paper.
7. Issue the command:

PROP environment -CONTINUE

Changing Printer Environments: An Example

Spooler phantoms are generally started up at system startup time. In this example, the spoolers have already been started.

Initializing a new environment for a printer is a two-step process: the old phantom must be stopped, and the new phantom must be started. In the example below, the phantom servicing environment PRO will be stopped and a new phantom that will service environment WIDE will be started. If there were a need to change paper or to make any physical adjustments to the printer, these procedures would be done during the time between the stop and the start instructions.

```
OK, PROP PRO -STOP FINISH
[PROP REV 19.0]
```

```
Wait... Acknowledged.
*** SYSTEM (user 92) at 16:08
*SPOOL* PRO      — STOPPING
```

```
OK, PROP WIDE -START
[PROP REV 19.0]
```

```
Wait... Acknowledged.

*** SYSTEM (user 92) at 16:08
*SPOOL* WIDE      — Starting
```

```
OK, PROP -STATUS
[PROP rev 19.0]
```

```
PRO           stopped
PR1           started
RUSH          started
WIDE          started
OK,
```

When the -START option is given, PROP starts up a new phantom to service the specified environment. There may be a brief time delay between the time the stop or start instruction is given and the actual action of the phantom. The phantom has actually stopped when the screen display shows:

```
*SPOOL* environment — STOPPING
```

Similarly, the phantom has actually started when the screen displays:

```
*SPOOL* environment — Starting
```

As illustrated above, the command line PROP -STATUS may be used to make sure that the phantom is operative.

DEALING WITH MECHANICAL PROBLEMS

If there are problems with the operation of the line printer, the order of corrective operations is:

1. Check that the printer is not jammed or out of paper.
2. Check that the printer is powered up and ready to print.
3. Check that the spool phantom is running by using the PROP -STATUS command. If the spooler is not running, go to the supervisor terminal and restart the spooler phantom by typing:

PROP environment -START

4. Make sure that the paper mounted on the printer matches that called for in the printer's environment. Check the specified paper by typing:

PROP environment -DISPLAY

5. If your system uses FAM I, make sure that user FAM has a priority setting of 2. Check this with STATUS USERS. If user FAM is not running at priority 2, set it using the CHAP command (see Chapter 13). If your system does not use FAM I, go to step 6.
6. See the System Administrator if the printer still does not function.

Note

If a printer has been running and then powers itself off (because of a blown fuse or power supply problem) it will jam other printers on the system. A cold start must be performed. Make sure that the faulty printer is not restarted during system startup.

11

Working With Batch

INTRODUCTION

The System Administrator sets up the Batch subsystem and defines the queues and their properties. The operator's responsibilities for the Batch subsystem generally consist of:

- Monitoring the Batch subsystem.
- Starting, "pausing", and stopping the Batch monitor.
- Controlling batch jobs when requested to do so.

The System Administrator may also request that the operator handle:

- Blocking and unblocking queues, using the BATGEN utility.
- Dealing with any problems that might affect the database, using the FIXBAT utility.

MONITORING BATCH

The operator monitors the status of the Batch subsystem by two methods: explicitly requesting information, and reading messages spontaneously displayed by Batch.

Requesting General Status Information

To determine the general status of the Batch system and monitor, give the command:

BATCH -DISPLAY

This will produce the following output:

- The number of waiting and held jobs per queue
- The filename, user-id, and queue name for each currently executing job
- The number of queues which contain waiting or held jobs

For example:

OK, BATCH -DISPLAY
[BATCH rev 19.0]

Number of waiting and held jobs:

Queue	Jobs
Normal-1	1
Normal-2	3

Total= 4 (2 queues)

2 currently running jobs:

User	Jobid#	#	Queue
CLOTHO	#10032	114	Normal-2
CLIO	#00172	117	Normal-1

For a brief summary of information, give the command:

BATCH -STATUS

BATCH -STATUS prints one line of information which describes the number of waiting and held jobs, and the number of executing jobs. In addition, either the total number of active jobs or the message "No batch jobs" is displayed. For example:

OK, BATCH -STATUS
[BATCH rev 19.0]
6 batch jobs; 4 waiting or held jobs in 2 queues; 2 executing jobs.

Requesting Information on User Jobs

Two options to the JOB command are useful for obtaining information on user jobs:

JOB -STATUS

and

JOB -DISPLAY

When given by a user logged in as SYSTEM or BATCH_SERVICE, the JOB -STATUS command displays only the user-id, job-id, status, external name, and queue of all active (that is, executing, held, or waiting) jobs. The JOB -DISPLAY command returns all of the information on all active jobs in the system.

For example, a JOB -STATUS command might return the following display:

OK, JOB -STATUS
[JOB rev 19.0]

User	Jobid#	State	External name	Queue
ORC	#00002	executing	AAAXXX	Normal-1
BALROG	#00003	waiting	MORIA.MAP	
BALROG	#00004	waiting	BRIDGE	
ENT	#10001	held	TREEWALK	Normal-2
SHELOB	#10003	waiting	\$WEB	
SAM	#20008	executing	ROPE	Express
SAM	#20009	waiting	SEARCH.MASTER	
GLLM	#30003	waiting	SMEAGOL	Background-1
GLLM	#30004	waiting	RING.0	
GLLM	#30005	waiting	SSSSSSSSSS	
SMAUG	#40012	held	\$PILE	Background-2
FRODO	#40013	held	RED.BOOK	
FRODO	#40016	waiting	SEARCH_ROUTINE	

The operator can monitor a specific active job, no matter what user submitted it, by giving the job-id in the JOB -STATUS or -DISPLAY command. For example:

JOB #10003 -DISPLAY

In addition, the operator can also monitor any of SYSTEM's jobs by its jobname. For example:

JOB SYS5 -DISPLAY

The following example illustrates the use of the JOB -DISPLAY command with a job-id:

```
OK, JOB #00003 -DISPLAY
[JOB rev 19.0]
```

```
Job MORIA.MAP(#00003), user BALROG waiting (queue Normal-1).
Submitted today at 1:39:24 p.m.
Funit=6, priority=5, cpu limit=None, elapsed limit=None.
Home ufd=<SYS.A>BALROG
```

19.2

The STATUS USERS command displays running batch jobs in its list of users. As of Rev. 19.2, batch jobs may be easily distinguished by the word batch in the Line column of the STATUS USERS display.

Understanding Spontaneous Messages

Whenever the Batch monitor initiates a user's job, the job sends a message to the supervisor terminal as follows:

```
*BATCH* Executing jobname for user username (job-id).
```

jobname is the filename of the job being run; username is the name of the user who submitted the job; and job-id is the number given the job by the Batch subsystem. When the job is completed (or aborted), the monitor sends the message:

```
Job jobname for user username (job-id) completed.
```

or

```
Job jobname for user username (job-id) aborted.
```

These messages help the operator monitor Batch usage and load without having to make explicit inquiries using BATCH -DISPLAY.

The System Administrator can prevent the printing of these messages by changing the command RESUME MONITOR in the file BATCHQ>START_BATCH_MONITOR.COM1 to read RESUME MONITOR -HUSH. When the monitor is next started, it will no longer send messages on job execution, completion, and abortion to the supervisor terminal.

Error Reporting by the Monitor: If a condition occurs that prevents the Batch subsystem from functioning correctly (for example, a full disk or a damaged database), the Batch monitor sends a warning message to the supervisor terminal. (Often these messages also ring the bell of the terminal.) Appendix G lists and explains these messages.

STARTING THE BATCH MONITOR

The Batch monitor is normally started up during system coldstart. Therefore, the operator rarely needs to explicitly start up the Batch monitor. However, an understanding of how the Batch monitor is started is important.

The command file CMDNC0>CONFIG or CMDNC0>PRIMOS.COMI usually contains a BATCH -START command as follows:

```
BATCH -START
```

(There may be two options, -RLEVEL and -TIMESLICE, on the command line. See Chapter 13 for more information on these options.) If Batch was not started when the system was started up, you can issue the identical command from the supervisor terminal to start it up.

Note

The startup procedure explained here differs from the startup procedure used at Revs. 17 and 18. The earlier procedure is not supported at Rev. 19.

How the BATCH -START Command Works

The BATCH -START command creates a phantom named BATCH_SERVICE. This phantom serves as the Batch monitor.

The monitor cannot begin work until the system time and date have been set. Therefore, if the BATCH -START command is given before time and date are set (as can happen when the BATCH -START command is included in the C_PRMO or PRIMOS.COMI file), the monitor does nothing until the SETIME command is given.

Once the system time and date have been set, the monitor runs a program called FIXBAT. FIXBAT ensures that a valid database exists for the processing of user jobs. The monitor then sends the following message to the supervisor terminal, notifying the operator that it is ready to process Batch jobs:

Monitor in operation.

If you give the BATCH -START command while the monitor is running, the Batch subsystem will ignore the command and send you the message:

Monitor already started.

If users submit Batch jobs when the monitor is not running, they receive the warning message:

Warning: jobs are not being processed at this time.

The jobs can be submitted despite the message. However, they will not be executed until the monitor has begun to process jobs.

PAUSING AND CONTINUING THE BATCH MONITOR

The operator may "pause" the monitor, thus keeping it from starting execution of any new jobs, while allowing it to:

- Complete currently executing jobs.
- Signal the completion or abortion of executing jobs.
- Delete queues.

To pause the batch monitor, give the command:

BATCH -PAUSE

When the monitor is paused, users who use the JOB or BATCH commands receive the message:

Note: the batch monitor is currently not starting up jobs.

To direct the monitor to start executing jobs again, the operator gives the command:

BATCH -CONTINUE

The operator can give the -PAUSE and -CONTINUE commands whether the monitor is started or stopped. Thus, the operator can pause the monitor an hour or two before stopping the system; stop and restart the system (still in its paused state); and then continue the monitor, allowing new jobs to be executed.

When the monitor is paused (or when it is started up in a paused state), the supervisor terminal receives the message:

Monitor paused.

When the monitor is continued, it sends the supervisor terminal the message:

Monitor continued.

If the operator attempts to pause a monitor that is already paused, the supervisor terminal receives the message:

Monitor already paused.

If the operator attempts to continue a monitor which is not paused, the supervisor terminal receives the message:

Monitor not paused.

STOPPING THE BATCH MONITOR

To stop the Batch monitor, give the command:

BATCH -STOP

When the monitor sees the BATCH -STOP command, it sends the following message to the supervisor terminal, and then logs itself out:

Operator stop.

The following message indicates that the monitor is not running or is just starting up:

Process not started.

Note

It is not advisable to stop and restart the monitor while jobs are running. When this happens, jobs that were running at the time of the BATCH -STOP command have a very slow turnaround — sometimes up to 10 minutes after their actual completion. (When the monitor is restarted, the phantom process that is running the job is logged out without signalling the monitor that the job is completed. Thus, the restarted monitor notices nothing until it does its periodic check of the queues. This period can be up to 10 minutes in duration.)

Forced Logouts

At Rev. 19, the operator can log out the monitor with the LOGOUT ALL or LOGOUT -nn command. The monitor will not log out immediately; instead, it will log itself out gracefully and send the message:

Force logout by operator.

The Batch database remains intact.

If the message is not sent within a short period of time, you can repeat the LOGOUT command. This second force logout will cause the monitor to log itself out immediately. However, it will leave the database in an unknown state. Since the BATCH -START command runs FIXBAT, it may repair the database. If not, either FIXBAT must be run interactively, or INIT must be run.

Example of Stopping the Batch Monitor

Following is an example of stopping and restarting the Batch monitor, showing how the commands and messages would appear at the supervisor terminal. A BATCH -DISPLAY command has been included to show the warning message sent when Batch is not running. At the time the command was given, there were no jobs in the Batch system.

```

OK, BATCH -STOP
[BATCH rev 19.0]
Stop request issued.
OK,
*** BATCH_SERVICE (user 101 on SYA.A) at 15:45
Operator stop.

Phantom 101: Normal logout at 15:45
Time used: -307h 18m connect 12m 06s CPU, 18m 21 I/O
BATCH -START
[BATCH rev 19.0]
Monitor started up.
OK,
*** BATCH_SERVICE (user 111 on SYA.A) at 15:46
Monitor in operation.
OK, BATCH -PAUSE
[BATCH rev 19.0]
Monitor paused.
OK,
*** BATCH_SERVICE (user 111 on SYA.A) at 15:47
Monitor paused.
BATCH -PAUSE
[BATCH rev 19.0]
Monitor already paused.
OK, BATCH -CONTINUE
[BATCH rev 19.0]
Monitor continued.
OK,
*** BATCH_SERVICE (user 111 on SYA.A) at 15:47
Monitor continued.
BATCH -CONTINUE
[BATCH rev 19.0]
Monitor not paused.
OK,
BATCH -DISPLAY
[BATCH rev 19.0]
No queues have waiting or held jobs.
No running jobs.

```

CONTROLLING BATCH JOBS

The operator has almost complete control over all jobs in the Batch subsystem. While logged in under SYSTEM or BATCH_SERVICE, the operator can perform any operation on a user job that the user could perform, with the following restrictions:

- The operator must refer to all user jobs by their job-id (instead of jobname).
- The operator cannot abort or restart jobs belonging to other users except from the supervisor terminal.

If the operator attempts an abort from a terminal other than the supervisor terminal, the abort will fail. If the operator attempts a restart under the same circumstances, the job will be flagged for restarting (assuming it is a restartable job), but the force logout which usually precedes restarting will fail because of insufficient access rights. The job will be restarted when it completes or aborts.

Here is an example of aborting a job. As this example shows, there is a brief interval between the time the JOB command acknowledges the ABORT command and the time when it informs the supervisor terminal that the job has been aborted. The message that the next waiting job in the queue (if any) has begun executing follows immediately:

```
OK, JOB #00003 -ABORT
[JOB rev 19.0]
Job MORIA.MAP(#00003) aborted.
```

```
OK, DATE
27 May 82 14:02:20 Thursday
```

```
***BATCH_SERVICE (user 104 on SYS.A) AT 14:02
Job MORIA.MAP for BALROG(#00003) aborted.
```

```
***BATCH_SERVICE (user 104 on SYS.A) AT 14:02
*BATCH* Executing BRIDGE for BALROG(#00004).
```

HANDLING BATCH QUEUES

Like any user, the operator can use the BATGEN -STATUS and BATGEN -DISPLAY commands to check the status or parameters of all currently defined queues. The operator can also use the BATGEN command to block a queue (thus temporarily closing it to new jobs) and to unblock a queue (opening it to jobs again).

Monitoring Batch Queues

To determine the status of all currently defined queues, issue the command:

BATGEN -STATUS

The currently defined queues and their statuses are displayed in tabular format. The status can be one of the following:

<u>Status</u>	<u>Meaning</u>
unblocked	The queue is available for job submission. This is the typical state for a queue.
blocked	The System Administrator or operator has blocked the queue. Users cannot submit new jobs to the queue. However, jobs that are already in the queue will continue to be processed by the Batch monitor.
flagged for deletion	The System Administrator has deleted the queue. The queue is in the process of being deleted, but there are still jobs in it that are waiting, held, or executing. When there are no more jobs in the queue, the Batch monitor will remove the queue from the list of currently defined queues, and will send a message to the supervisor terminal notifying you that this has happened.

To list the parameters of all currently defined queues, issue the command:

BATGEN -DISPLAY

See the System Administrator's Guide for complete information on the BATGEN -DISPLAY command and its output.

Blocking and Unblocking Queues

To block or unblock a queue, issue the command:

BATGEN BATCHQ>BATDEF

Once the file BATCHQ>BATDEF has been read and validated, BATGEN types a prompt character and waits for a BATGEN command. For example:

```
OK, BATGEN BATCHQ>BATDEF
[BATGEN rev 19.0]
>
```

Available commands are:

BLOCK { queueName }
 { ALL }

UNBLOCK { queueName }
 { ALL }

DISPLAY { queueName }
 { ALL }

STATUS

FILE

QUIT

Find out what the list of queues is by typing:

STATUS

Determine which queue is to be blocked or unblocked, and type:

BLOCK queueName

or:

UNBLOCK queueName

Issue another STATUS command to verify that you changed the right queue. Then, type:

FILE

At this point, the changes are made to the Batch subsystem. For more information on these commands, see Chapter 13 or the System Administrator's Guide.

USING FIXBAT

FIXBAT is a utility designed to:

- Handle the startup protocol for the Batch monitor, making sure that the database is valid before starting the monitor.
- Fix any broken pointers within the queue files.
- Reclaim disk space by deleting from the Batch queues all inactive jobs of a given age or older.

FIXBAT is run automatically every time the Batch monitor is started up by the BATCH -START command. The System Administrator decides whether FIXBAT merely checks for a valid database during this procedure (cleaning up the database, if necessary), or whether it also reclaims disk space by removing old jobs from the queues.

FIXBAT also may be run interactively. (If the Batch database becomes invalid, for instance, you would run FIXBAT interactively to repair it.)

Running FIXBAT at Startup Time

FIXBAT is run automatically by the Batch monitor whenever it is started up by the BATCH -START command. The command which runs FIXBAT is found in the command file BATCHQ>START_BATCH_MONITOR.COM1. As released, the command is:

```
RESUME BATCHQ>FIXBAT -STARTUP SAVE
```

This command checks to see that the database is valid before beginning the monitor, but it does not clean old jobs out of the database. Since most administrators do want this cleanup done on a frequent basis to conserve disk space, your System Administrator probably has added the -DAYS option to the command line. The -DAYS option takes a numeric argument. The most commonly used values are 0, 1, and 2. The argument 2 cleans out jobs run 2 or more days ago. The argument 1 cleans out jobs run at least 1 day ago. The argument 0 cleans out all finished jobs. See Chapter 13 for more information on FIXBAT options.

If FIXBAT aborts, the cause generally can be found by looking at the log file. Usually, deleting the offending file and restarting the Batch monitor (and therefore using FIXBAT) is the fastest way to fix any problems.

If FIXBAT has been run by the BATCH -START command, then FIXBAT has been running as the Batch monitor. In this case, when FIXBAT has finished, the BATCH -START command will resume monitoring and the monitor revision number will be typed out, followed by a log trail of its activities.

Invoking FIXBAT Interactively

FIXBAT resides as a program, FIXBAT.SAVE, in the BATCHQ UFD. To run FIXBAT:

1. Log out the Batch monitor (if it is running), using the command BATCH -STOP.
2. Log in as SYSTEM, or as a Batch administrator.
3. Enter the command:

RESUME BATCHQ>FIXBAT.

If you try to start FIXBAT while the Batch monitor is running, FIXBAT returns with the error message:

Batch monitor is running, do BATCH -STOP. (FIXBAT)

When invoked in this manner, FIXBAT will try to make all necessary repairs to the Batch database. If it returns an error message, try deleting the offending file named in the error message, and then run FIXBAT again. If this doesn't work, enter the command:

RESUME BATCHQ>INIT

This will recreate the Batch database, destroying all data on user's jobs, but leaving the queue configuration intact. Then try running FIXBAT again. If it still doesn't work, enter the command:

RESUME BATCHQ>INIT -RESET_QUEUES

Now, running FIXBAT should work. If it doesn't, ask your System Administrator or Customer Service Representative for assistance.

Using FIXBAT for Cleanup Operations

When FIXBAT is run interactively (without the -STARTUP option), it automatically fixes the database. When FIXBAT is run with the -STARTUP option (as with BATCH -START) however, it fixes the database only if one (or more) of three conditions is met:

- If -DAYS n has been specified in order to remove old jobs from the queue.
- If it cannot find the file BATCHQ>OTHER>VALID. (The absence of this file indicates an invalid database.)
- If it cannot find the MON.ST file in the BATCHQ UFD. The absence of this file indicates that the monitor was not logged out gracefully.

This is the case if the monitor encountered a fatal error, was forcibly logged out, or was halted by a premature system shutdown.

Deleting the Old Batch Job Entries: When FIXBAT deletes old Batch job entries from the queue files, it physically removes a job entry from the queue and writes the next job entry over the deleted one, repeating this procedure until the end of the queue file is reached.

It will perform this operation only if a -DAYS argument was specified on the command line.

The procedure for determining whether or not a job should be deleted is as follows:

1. The job must not be an active job; that is, it must be in a cancelled, aborted, or completed state.
2. Unless -DAYS 0 was specified, the job must have been completed, aborted, or cancelled in the current year or the previous year.
3. Unless -DAYS 0 was specified, the job must have been completed on a date that allows at least n full days between that date and the current date, not including those dates. (n is the argument supplied to the -DAYS option.) This means that if a job was completed on April 10, 1982, and the current date is April 12, 1982, the only way that job can be deleted is if n is 1. If n is 2, the job will not be deleted until the next day.

When FIXBAT deletes a job, it presents the final information on that job in a format similar to that of the information returned by a JOB -DISPLAY command (unless the -QUIET option was specified on the command line, in which case no information is displayed).

Note

If a deleted job is displayed, the queue name may be blank. This occurs if the user did not explicitly specify a queue. Also, the queue name may not resemble the uppercase/lowercase mapping of the queue name as defined in BATGEN. For example, output for queue COBOL might appear as (queue COBOL), (queue cobol), or (queue).

FIXBAT Error Messages and Responses

While FIXBAT is running, it may present certain messages describing what it is doing, or it may abort with a particular error message.

In general, if FIXBAT aborts, it means that certain parts of the database are irretrievably lost. This will usually be Batch job data. While deleting the offending file and rerunning FIXBAT may help, this procedure does not guarantee that FIXBAT won't abort on a different file.

If FIXBAT does not seem to be able to fix the database, the INIT program should be invoked.

Cleaning Up Queues

Each Batch queue numbers its jobs from 0000 to 9999. When number 9999 is reached, the queue is considered full, whether it still contains jobs or not.

When full queues exist, the following things happen:

- When users submit jobs to the full queue (using the JOB command's -QUEUE option), they receive the error message "queue full".
- When users submit jobs without specifying queues, the JOB program conducts its usual search for queues. However, it ignores the "full" queue, treating it as if it were blocked. If the full queue is the only queue that meets a user's requirements, that user receives the error message "No queue available for job." If some other queue is acceptable, the monitor simply submits the jobs to that queue.

Therefore, when a queue becomes full, the System Administrator must first delete the queue and then redefine it, so that new jobs may be submitted to it.

12

Working with FTS

INTRODUCTION

This chapter explains the commands the operator uses in day-to-day maintenance of the File Transfer Service (FTS). For full information on FTS, refer to the PRIMENET Guide.

As an operator, your responsibilities with regard to FTS are:

- Monitoring FTS.
- Managing file transfer requests.
- Controlling the file transfer servers.

The two commands used to perform these tasks are FTR and FTOP.

MONITORING THE FILE TRANSFER SERVICE

There are four areas of the File Transfer Service (FTS) that you should periodically monitor. These areas are:

- User requests, using FTR -STATUS and FTR -DISPLAY.
- Servers, using FTOP and STATUS USERS.
- Log files.
- The FTSQ* directory.

Monitoring User Requests

While it is the responsibility of the submitting user to see that a file transfer is successful, the operator should watch for requests that have been repeated many times or have been put on hold for a long period. The cause of such problems is identified by examining the server log file and with the FTR -STATUS or FTR -DISPLAY command, as illustrated in the examples below.

OK, FTR -STATUS

[FTR rev 2.0]

83-11-09.10:03:12 SYSTEM LOG.COMO (1) (FIS\$1) Status - waiting

83-11-09.10:03:51 SHELOB ACRONYMS (2) (FIS\$1) Status - waiting

OK, FTR -DISPLAY

[FTR rev 2.0]

Request - LOG.COMO (1)
 User - SYSTEM
 Queue - FIS\$1
 Queued - 83-11-09.10:03:12 Status - waiting
 Last attempt - 00-00-00.00:00:00 Attempts - 0
 Current time - 83-11-09.10:04:08
 Source file - <ZAPHOD>GOLD>LOG.COMO
 Source file size - 1226 bytes.
 Destination file - <SYSA>PREFECT>LOG.COMO
 Source site - ZAPHOD
 Destination site - SYSA
 Source user - SYSTEM
 Destination user - FORD
 Source file type - SAM
 Destination file type - SAM
 Options :-
 BINARY, COPY, NO DELETE, SOURCE NOTIFY, NO DESTINATION NOTIFY.
 More ?YES
 Request - ACRONYMS (2)
 User - SHELOB
 Queue - FIS\$1
 Queued - 83-11-09.10:03:51 Status - waiting
 Last attempt - 00-00-00.00:00:00 Attempts - 0
 Current time - 83-11-09.10:04:10
 Source file - <ZAPHOD>SHELOB>ACRONYMS
 Source file size - 1226 bytes.
 Destination file - <SYSB>INFO>ACRON
 Source site - ZAPHOD
 Destination site - SYSB
 Source user - SHELOB
 Destination user - ARTHUR
 Source file type - SAM
 Destination file type - SAM
 Options :-
 BINARY, COPY, NO DELETE, SOURCE NOTIFY, NO DESTINATION NOTIFY.
 OK,

19.3

Typical causes for problems with file transfers are: the network is congested; the computer at the remote site is not running; the remote FTS server or manager has not been started. When the cause of the difficulty has been identified, the operator should take appropriate action to rectify the difficulty.

Monitoring Servers

The file transfer servers and the FTS manager are phantom processes. Server names are assigned when the server is added to the FTS configuration. The manager phantom is named YTSMAN. These processes are controlled with the FTOP command (described below). Additionally, they should be periodically checked with the STATUS USERS command (described in Chapter 3) to make sure they are in operation.

Both the FTS server phantoms and the FTS manager phantom maintain command output files while they are running. Server files have the pathname FTSQ*>COMO.FTS>servername. The command output file generated by the manager has the pathname FTSQ*>YTSMAN.COMO. More information on servers is provided below.

Monitoring and Archiving FTS System Log Files

FTS system log files are maintained in the UFD FTSQ*. The file names are specified by the System Administrator as part of the FTS configuration. Server log files record all events for incoming and outgoing file transfers, and can be useful in providing a record of FTS usage when tracking the progress of a particular request. The server log should be examined daily to check the smooth running of the FTS system. This is done by using ED, locating the current date in the file, and then locating RESULT. The operator can use EDITOR's X command to repeat the LOCATE RESULT command line. For example:

```
OK, ED FTSQ*>FTP.LOG
EDIT
L June 10
00.00.18: [1.1] Request GNZO.9JUNE (8553233) started Thursday, June
10, 1982
L RESULT
00.00.19: [1.1] RESULT: Transfer Aborted : Out of order.
X
14.31.59: [2.1] RESULT: Transfer Rejected: File not available.
X
14.32.50: [2.1] RESULT: Transfer Rejected: Problem with remote
file.
X
14.38.08: [4.1] RESULT: Transfer Terminated: Satisfactory and
Complete.
```

Log files are not limited in size, and should thus be regularly archived so that the FTSQ* directory does not become full.

Monitoring the UFD FTSQ*

In addition to log files, the FTSQ* directory holds copies of user files that are to be transferred. The operator should make sure there is adequate disk space available to accommodate these copies.

MANAGING FILE TRANSFER REQUESTS (FTR)

The FTR command, by which users submit and monitor their file transfer requests, is described generally in the Prime User's Guide, and in detail in the PRIMENET Guide. Users other than the operator may use FTR only to control their own file transfer requests. The operator, who has gained special privileges by logging in as SYSTEM, uses this command to manage all users' file transfer requests.

For example, to abort a request that is already in progress, enter the command:

FTR -ABORT request-name
or
FTR -ABORT request-number

The specified request will be put on hold. Later, when you are ready to release the request and allow the transfer to take place, enter the command:

FTR -RELEASE request-name
or
FTR -RELEASE request-number

There are other FTR options that can be used to control user's jobs. These options are fully described in Chapter 13.

CONTROLLING THE FILE TRANSFER SERVERS (FTOP)

The FTOP command is available only to the operator at the supervisor terminal. With FTOP, the operator can start, stop, and monitor the operation of the file transfer manager (YTSMAN), and the file transfer servers, which are the phantoms that handle file transfer requests. The System Administrator can configure up to eight file transfer servers, each of which takes requests from its own queue of file transfer requests. (For information on the System Administrator's tasks with regard to FTS, see the System Administrator's Guide.)

Each file transfer server can handle up to eight file transfer requests simultaneously. For each transfer, the server creates a separate virtual circuit, or communications link, between the source and destination nodes.

The circuits that a server can potentially create are called links. Thus, a server might have five of its eight links active at a given moment.

Server processes must be started from the supervisor terminal. Once started, a server continues to run even when there are no requests to handle. However, a server may be stopped and restarted by the operator if necessary.

Note

The FTOP server should be stopped before it is logged out. This ensures that all transfers currently in progress will be successfully completed before the logout. The FTP server will print a message on the supervisor terminal when it has stopped.

In addition to the file transfer servers, FTS employs a phantom manager process called YTSMAN. YTSMAN receives file transfer requests from remote nodes and passes them to appropriate local servers. Like the servers, YTSMAN must be started from the supervisor terminal.

Note

When the local FTS server and YTSMAN are not running, local users may still queue requests by using FTR.

To start up the FTS manager, named YTSMAN, enter the following command at the supervisor terminal:

```
FTOP -START_MNGR
```

To start up an FTS server process named server_name, enter the following command at the supervisor terminal:

```
FTOP -START_SRVR server_name
```

Both of the above commands are normally included in the C_PRMO or PRIMOS.COMI file on your system. In a STATUS USERS display, the FTS manager has the username YTSMAN, and each FTS server has the username specified when it was started up (server_name).

To obtain a list of all configured servers, including the status and user number of each server, enter the command:

```
FTOP -LIST_SRVR_STS
```

A few minutes before shutting down the system, the FTS servers should be told to shut down as soon as they complete any transfers in progress. To do this, enter the following command for each running FTS server:

```
F*TOP -STOP_SRVR server_name
```

(You must be logged in as SYSTEM to use this command.) As each server shuts down, it will send a message to the supervisor terminal.

Sometimes, it may be necessary to immediately shut down the FTS servers, even if they are currently transferring files. This is known as "abandoning" the FTS servers. When an FTS server is abandoned, it places any file transfers it is currently processing on hold in the queue, so that they can be started up again later. It then logs itself out. To abandon an FTS server, enter the following command while logged in as SYSTEM:

```
F*TOP -ABND_SRVR server_name
```

If you want to stop remote file transfer requests, but continue to allow local requests, you can use the following command:

19.3

```
F*TOP -STOP_MNGR
```

This command tells YTSMAN (the file transfer manager) to complete current requests and log out. If the manager is not running, a message is printed.

For more information on the F*TOP command options, see Chapter 13.

PART III

Reference

13

PRIMOS

Commands Reference

INTRODUCTION

This section describes PRIMOS commands that are normally used at the supervisor terminal. Some of these commands may also be available for use under PRIMOS II; if so, this is indicated in the command description.

Generally, the supervisor terminal is used only to:

- Start up and shut down disks.
- Configure devices.
- Check status.
- Collect a record of LOGIN, LOGOUT, and other messages.

External commands and the internal commands RESTOR, RESUME, and START may also be invoked from the supervisor terminal under PRIMOS.

PRIMOS COMMANDS► ADDISK

ADDISK searches the table of logical disks for an available table entry location; if one is found, ADDISK starts up each specified disk partition.

There is a limit of 63 disks that may be added to a system. This limit applies to both local and remote disks. If there are 63 disks added to a system, the logical disk numbers will range from 0 to 75 octal.

Starting Up Local Disk Partitions: Before a disk can be used by users, it must be started up on the local system.

Starting up a local disk partition does not make the partition available to remote systems unless (or until) the remote systems add the partition as a remote partition, as described below.

The command format for starting up local disk partitions is:

$$\text{ADDISK } \left\{ \begin{array}{l} [\text{PROTECT}] \text{ pdev-1 } [\text{pdev-2}] \dots [\text{pdev-n}] \\ \text{pdev -RENAME packname} \end{array} \right\}$$

pdev is the physical device number of the partition being added.

ADDISK takes either the PROTECT argument or the -RENAME option, but not both at the same time:

PROTECT pdev ... Assigns write-protection for pdev. PROTECT may be specified only for disks that are added locally via the ADDISK command.

The status of the write-protection assignments may be changed for a running disk by shutting down the disk and then respecifying the ADDISK command with or without the PROTECT option. This will change its protection. (Whenever possible, a disk that is write-protected via ADDISK PROTECT should also have write protection physically enabled, to ensure that the disk is not inadvertently written upon.)

pdev -RENAME packname Allows the operator to specify a new name for a disk when adding it to the system. This is required when the disk being added has the same name as a disk already on the system.

If the operator adds such a disk without renaming it, the system will print the error message:

Name "packname" of disk pdev not unique
(conflicts with LDEV n).

Only one device at a time can be added when using the -RENAME option. For example:

```
ADDISK 460 -RENAME ATON
```

The SHUTDN command also uses the -RENAME option. Thus, the operator may restore the disk's original name (or give it a new name) when removing it from the system.

Caution

Do not use ADDISK to specify a device that is already started, or unpredictable results may occur.

Starting Up Remote Disk Partitions: If a local computer system is part of a network, disks on other systems on the network may be made available to users on the local system by the command:

```
ADDISK packname-1 [packname-2...packname-9] -ON nodename
```

nodename is the network name for a valid RFA (formerly FAM II) system. The node must be RFA-enabled (see CONFIG_NET). packname-n is the name of the remote partition. The remote disk does not have to be started, nor does the remote system have to be up.

19.3

Remotely added devices have write-protection status assigned them at their local system. The write-protection status of a device cannot be changed remotely.

Notes

Disks to be added remotely are specified by packname rather than by physical device number.

All disknames must be unique. It is not possible to add a new disk if its diskname is the same as that of a disk already started.

19.3

ADDISK Messages: ADDISK prints the messages listed below. Warning messages are indicated by (Warning). After a warning message has been printed, ADDISK will continue if more than one device has been specified.

- *** Cannot read DSKRAT of disk pdev.

(Warning) The disk specified as pdev has not been formatted by Rev. 19 MAKE. For details on MAKE, see Chapter 6.

- Conflicting PDEVs "pdev1" and "pdev2". (addisk)

Two of the PDEVs given in the list specify overlapping partitions of the same disk.

- *** Disk pdev is not a PRIMOS partition.

(Warning) The disk specified as pdev has not been formatted by Rev. 19 MAKE. For details on MAKE, see Chapter 6.

- *** Disk pdev: Not ready.

(Warning) A nonexistent device, or one that is not ready, has been specified as pdev.

- *** Disk pdev: Old partitions not supported.

(Warning) The disk specified as pdev has not been formatted by Rev. 19 MAKE. For details on MAKE, see Chapter 6.

- Disk table overflow: n entries required, only x free. (addisk)

The number of free entries (x) is smaller than the number of requested additions (n). Logical device numbers may be freed by using SHUTDN to remove devices from the table. PRIMOS supports 63 logical devices.

- Duplicate partition name "packname". (addisk)

Partition name packname was given more than once in this remote ADDISK command.

- Duplicate PDEV "pdev". (addisk)

PDEV pdev was specified more than once in this ADDISK command.

- *** Format of disk pdev not supported by this revision of PRIMOS.

(Warning) The disk specified as pdev has been formatted by a post-Rev. 19 MAKE. For details on MAKE, see Chapter 6.

- Must specify at least one PDEV. (addisk)

The ADDISK command requires at least one argument.

- Must supply at least one partition name. (addisk)

The ADDISK command requires at least one argument.

- *** Name "disk" of disk pdev not unique (conflicts with LDEV n).

(Warning) Partition disk already exists on this system's disk list. This message is given when "-ON nodename" is not used.

- Node "nodename" not configured in network. (addisk)

The node specified as nodename is not currently RFA enabled.

19.3

- Only one disk may be RENAMEd at a time. (addisk)

Only one PDEV may be specified when the -RENAME option is used.

- "packname" is not a valid partition name. (addisk)

The specified packname does not conform to partition name syntax. Either it contains an illegal character, or it is more than six characters long.

- "pdev" is not a valid PDEV. (addisk)

The pdev specified is not a legal physical device number. Either it contains decimal numbers or nondigits, or it is simply an illegal PDEV.

- PDEV pdev conflicts with assigned or paging device. (addisk)

The pdev is currently being used as, or overlaps with, the primary or alternate paging partition, or is an assignable disk.

- PDEV pdev conflicts with existing LDEV n. (addisk)

The pdev overlaps with logical disk n.

- Remote partitions may not be RENAMED. (addisk)

The -RENAME option may be used only for local disks.

- Starting up revision n partition "packname".

Disk packname is being added. n is either 18 or 19. On a Rev. 19 partition, this may be followed by the message: (Quota system may be incorrect; run FIX_DISK.)

- System console command only. (addisk)

The ADDISK command may be issued only from the supervisor terminal.

- "system" is not a valid system name. (addisk)

The specified remote system name is illegal. Either it contains illegal characters, or it is more than six characters long.

- System name must be specified with -ON option. (addisk)

The -ON option was given, but was not followed by a remote system name.

- Write-protected disks may not be RENAMED. (addisk)

The -RENAME option may not be specified along with the PROTECT option.

► AMLC

19.3 |

The AMLC command configures an asynchronous line connected to an AMLC, ICS1, or ICS2 controller. The command format is:

AMLC [protocol] line [configuration] [lword]

protocol is one of the following:

TTY Default terminal protocol

TRAN Transparent (no character conversion)

TTYNOP All traffic ignored

TTYUPC Lowercase alphabetic characters translated to uppercase for output; uses normal terminal protocol for input

The following are obsolete protocols which are used only with DMT AMLCs (Model 505x). These protocols will slow down performance if used with any other controller:

TTYHS Terminal with per-character interrupt

TRANHS TRAN with per-character interrupt

TTYHUP Lowercase alphabetic characters translated to uppercase for high-speed output; uses normal terminal protocol for input

line is the asynchronous line number (in octal). The maximum value is '177 octal (127 decimal).

configuration is a 16-bit halfword (in octal) used to set the line configuration. Three common configuration values, and the baud rates they represent, are:

<u>configuration</u>	<u>baud rate</u>
2213	300
2313	1200 (default)
2413	9600 (typically)

lword is a 16-bit halfword (in octal) which specifies configuration attributes of asynchronous lines.

The AMLC command is described in detail in the System Administrator's Guide.

► ASSIGN

The operator's primary use of the ASSIGN command is to assign disk partitions and tape drives as part of system maintenance (such as during backup procedures, when formatting disks, and when repairing file system partitions). Although the ASSIGN command is fully described in the PRIMOS Commands Reference Guide, the description of how to assign disk partitions and tape drives is included here for convenience.

The format of the ASSIGN command for disks is:

ASSIGN DISK pdev.

The specified disk, pdev, must have already been entered in the Assignable Disks Table via the DISKS command, described later in this chapter. See Appendix A for information on determining pdev.

The format of the ASSIGN command for magnetic tapes is:

```

|  ASSIGN { MTpdn  [-ALIAS MTldn] } [options]
|  { MTX -ALIAS MTldn }

```

Explanations of these arguments are as follows:

<u>Argument</u>	<u>Description</u>
MTpdn	Assigns a specific tape drive, <u>pdn</u> . Magnetic tape (MT) units are numbered from 0 to 7, inclusive.
MTX	Assigns an available drive. Must be accompanied by -ALIAS <u>MTldn</u> , which assigns a number (alias) to the drive for reference purposes. Legitimate values for <u>ldn</u> are 0-7. The actual drive assigned depends on any other options that appear on the command line.

Note

This argument cannot be specified from the supervisor terminal.

Users may also request certain special options with their tape drive assignments. The options to the ASSIGN command are:

<u>Option</u>	<u>Description</u>
-ALIAS <u>MTldn</u>	Allows the user to specify tape drives with logical device numbers, from 0 to 7 inclusive. <u>Any physical device may be used as long as the drive is specified by the requested ldn.</u> This feature is particularly handy for use in programs that run from CPL programs or command files. It allows the programs to refer to logical device numbers, which remain constant, and ignore physical device numbers, which may change from run to run, depending on the availability of particular drives.
-WAIT	Indicates that the user is willing to wait until the requested drive is available.

Caution

This option should not be used at the supervisor terminal. Doing so will cause the terminal to hang until the tape drive can be assigned by PRIMOS.

- TPID *id* Operator intervention required. Requests the operator to locate and mount a particular reel of tape, identified by a *tape id*. An *id* is a tape identifier (argument) describing a particular reel of tape, and/or type of tape drive (name, number, and so on). Identifiers may not begin with a hyphen (-), which is a reserved character indicating the next control argument on the ASSIGN statement line.
- { -RINGON } Operator intervention required. Specifies protection
 { -RINGOFF } rights by:
- RINGON Read- and write-permitted.
 or
 -RINGOFF Read only; write-protection in effect.
- Operator must remove or replace write-ring.
- DENSITY *n* Operator intervention may be required. Specifies tape density in bpi (bits per inch). Legal values for *n* are 800, 1600, 3200, and 6250. The operator must check the tape drive to make sure it is capable of functioning at the desired density setting.
- MOUNT Operator intervention required. Indicates a new tape is to be placed on a previously assigned drive.
- { -7TRK } Operator intervention required. Specifies 7- or
 { -9TRK } 9-track tape drive. Usually used in conjunction with the MTX argument. The operator must choose the correct tape drive.
- RETENSION Causes the tape to be fast-forwarded to end-of-tape and then rewound to beginning-of-tape. This stabilizes the tape-to-head pressure and stacks the tape evenly on the reel. Applies to cartridge tape drives (version 5 controller) only. If used on any other tape drive, this option is ignored.
- SPEED { 25 } Selects the speed for streamer tape drive running at
 { 100 } 1600 bpi. Ignored by other drives and by streamer running at 3200 bpi.

► BATCH

The BATCH command starts and stops the Batch monitor. It also allows operators and users to monitor usage of the Batch subsystem. BATCH is fully described in Chapter 11. The command format is:

BATCH options

The options are:

<u>Option</u>	<u>Function</u>
-CONTINUE	Takes the monitor out of a paused state allowing it to begin processing jobs again.
-DISPLAY	Lists waiting, held, and executing job information by queue.
-PAUSE	Tells the monitor not to start up jobs, but to finish processing jobs that are currently executing.
-START [options]	Starts the monitor.
-STATUS	Summarizes waiting, held, and executing job information for all queues.
-STOP	Stops the monitor.

The -START option works only from the supervisor terminal. The other options may be used from any terminal, but they require that the user have ALL access to the BATCHQ UFD. This normally includes user SYSTEM, user BATCH_SERVICE, and the System Administrator; however, your System Administrator may have defined other users as having ALL access to BATCHQ and its contents. When using the -START option to start the Batch monitor, the command format is:

```
BATCH -START [-RLEVEL rlv] [-TIMESLICE ts]
```

This command usually is included in the C_PRMO command file (see Volume I of this book), and thus is normally part of the system startup routine. However, it may be given by the operator if the Batch monitor is stopped and restarted while the system is running.

The value for RLEVEL must be between 0 and 3. Its default is 1. The value for TIMESLICE must be between 1 and 99. Its default is 20. (Both numbers are specified in decimal.) The options -START, -RLEVEL, and -TIMESLICE may be given in any order.

► BATGEN

The System Administrator uses the BATGEN command to configure, add, or delete Batch queues. The operator uses it to monitor, block, and unblock queues. If users have Read access to the BATDEF file, users can invoke the BATGEN command to monitor queue availability and parameters. BATGEN is described in Chapter 11. The command format is:

BATGEN [pathname] [option]

Normally, pathname is not specified, and defaults to the "live" queue configuration file, BATCHQ>BATDEF.

The options are as follows:

<u>Option</u>	<u>Function</u>
<u>-STATUS</u>	Displays a tabular list of defined queues and the status of each queue (blocked or unblocked).
<u>-DISPLAY</u> [queue]	Displays complete information on the queue named <u>queue</u> . If <u>queue</u> is not specified, all queues are displayed.

Only one option may be present on the command line. If no options are specified, BATGEN enters BATGEN command mode. The operator may enter commands that modify the queue configuration, then enter the FILE command to write the queue configuration file. If the file is BATCHQ>BATDEF, the changes take effect immediately. Only certain users may do this, as it requires RW access to BATCHQ>BATDEF. Normally, users SYSTEM, BATCH_SERVICE, and the System Administrator have the required access, and all other users have R access.

The BATGEN commands of interest to the operator are defined on the following pages.

<u>Command</u>	<u>Function</u>
<u>BLOCK</u> { queue } ALL }	Sets flag in status control block of an existing queue (or of all queues) to disallow submission of further jobs to the queue.
<u>UNBLOCK</u> { queue } ALL }	Resets flag to allow submission of jobs to a previously blocked queue (or to all queues). Default status for queues is "unblocked".
<u>DISPLAY</u> [queue] ALL }	Displays name, status, and characteristics of the named queue (or of all queues). Omitting the optional argument displays information for all queues.

<u>STATUS</u>	Shows name and status of all queues in tabular form.
<u>FILE</u> [pathname]	Modifies file named <u>pathname</u> to include commands given during this session. If <u>pathname</u> is not given, current file is modified (the usual situation).
<u>QUIT</u>	Terminates session without changing file. If anything was modified during the session, BATGEN will ask, "Environment modified, OK to quit?" A "yes" answer (or a carriage return) is then needed to execute QUIT. (BATGEN may be restarted with the PRIMOS START command after a QUIT, with no loss of information.)

► BOOT_CREATE

The BOOT_CREATE utility allows you to make a boot tape from which you can boot your system in the event of a system failure.

To make the system boot tape you must first create a List-File that contains the pathnames of all the directories and files necessary to restore your system to normal working conditions. BOOT_CREATE then uses the information contained in the List-File to generate the system boot tape.

When the BOOT_CREATE program is run, the List-File is read twice; the first pass checks that the files listed are present and accessible, the second pass calls the magnetic tape save utility MAGSAV which writes the files to the tape.

If either of the checks in the first pass of the List-File is unsuccessful, an error message is displayed.

When you use the boot tape to restore your disk, any UFDs that were originally ACL-protected are restored as unprotected UFDs.

The command line format for BOOT_CREATE is;

BOOT_CREATE [options]

Four command line options can be used with BOOT_CREATE:

List-File_name This is the name of the List-File.

{ -HELP }	This option displays a Help text that shows you the calling sequence of BOOT_CREATE and explains the options available.
{ -H }	

{ -NO_QUERY } This option suppresses a prompt for the
 { -NQ } List-File password.

If your List-File is contained in a password protected UFD and you haven't specified the password in the command line, or the password you have specified is wrong, an error message is displayed and BOOT_CREATE aborts.

-MT [n] This option suppresses the magnetic tape drive number prompt. If [n] is omitted, the system default is Drive 0.

Operator Prompts: BOOT_CREATE asks you to supply answers to the following prompts:

Does <DISKNAME>UFDNAME>SUBUFDNAME have a password ?

You are asked this question when you give the BOOT_CREATE command. If you answer YES to this prompt BOOT_CREATE asks you for the password of the UFD that contains the List-File. If, after answering YES, you fail to give the password or give an incorrect one BOOT_CREATE aborts. If the UFD is not password protected answer NO to the prompt and BOOT_CREATE will continue.

Enter List-File name:

If you didn't specify a name for the List-File in the BOOT_CREATE command line you are asked to provide one.

Which drive are you using ? (0 to7):

If you didn't specify the -MT option in the BOOT_CREATE command line you are asked to provide the number of the tape drive you have assigned.

Have you mounted the tape on drive <device number> ?

BOOT_CREATE asks this question after you have entered a tape drive number in response to the previous prompt. If you answer YES to this prompt, BOOT_CREATE continues. If you answer NO, BOOT_CREATE asks you to mount the correct tape on the tape drive.

If you don't want the prompts displayed at your terminal you can suppress them by using the command line options -MT and -NO_QUERY. If you run BOOT_CREATE on a phantom process, you may find it useful to make COMO files of the terminal output.

Creating the List-File: The List-File can reside anywhere in the system and can be created using the Prime line editor ED or the Prime screen editor EMACS.

The List-File must contain pathnames for PRIMOS II and MAGRST; other utilities and PRIMOS related files can be added as required. When you are creating the List-File you can specify the pathname of the directory or file that you want to save in several ways:

<DISKNAME>UFDNAME>DIRECTORY_NAME

<DISKNAME>UFDNAME>FILENAME

UFDNAME>DIRECTORY_NAME

UFDNAME>FILENAME

DIRECTORY_NAME

FILENAME

You can only use the last two formats if the directories or files you are specifying are contained in the UFD that you are currently working in.

List-File Arguments: The following arguments can be used by appending them to the pathnames of the specified directories and files; only one argument can be used with each pathname.

- YES This causes BOOT_CREATE to display an error message if the specified file or directory can't be located or accessed and then to terminate the program.
- NO This causes BOOT_CREATE to display a warning if the specified file or directory can't be located or accessed, and then continue writing the remaining objects to the tape.
- CHECK This allows you to check for a file's existence in a directory without having to save it. If the file exists, a blank line is submitted to BOOT_CREATE instead of the pathname.

If the file isn't located, an error message is displayed and the BOOT_CREATE aborts.

A recommended format for the boot tape is to have the PRIMOS II and PRIMOS files contained in two separate logical tapes; this is achieved by using an -EOT line in the List-File. The tape can then be structured so that all the files required by PRIMOS II are contained in logical tape 1 and the objects required to boot PRIMOS are contained in subsequent logical tapes.

Sample List-File: The following example gives the suggested contents of the List-File and shows how it is created using ED, the Prime line editor.

For the purposes of this example the List-File is being created in a UFD called TEST>MFD which is protected with the password FINE.

The information you would enter is shown underlined:

```
OK, ATTACH 'TEST FINE>MFD'
OK, ED
INPUT
DOS>*DOS64
CMDNC0>MAGRST
CMDNC0>MAKE
CMDNC0>FIXRAT
CMDNC0>COPY_DISK
CMDNC0>NSED
CMDNC0>FUTIL
CMDNC0>PHYRST
CMDNC0>C_PRMO -CHECK
CMDNC0>CONFIG -CHECK
-EOT
CMDNC0 -NO
PRIRUN -NO
SYSTEM -NO
SAD -NO
EDIT
FILE BOOT_LIST
OK,
```

Invoking BOOT_CREATE: When the List-File has been created, the BOOT_CREATE program is run to generate a boot tape. If you created the List-File in a password-protected UFD and want to use the -NO_QUERY option, you must specify the password in the pathname when you give the BOOT_CREATE command. The complete pathname must be enclosed within apostrophes, and must be written in upper case with the password separated from the UFD name by one space.

If you run the BOOT_CREATE program without the -NO_QUERY option, you will be prompted for the password if one is required.

Sample BOOT_CREATE Session: This example runs the BOOT_CREATE program with the command line option -NO_QUERY, and uses the List-File BOOT_LIST created above. As the -CHECK option was used with two of the pathnames in the List-File, two blank lines appear near the end of logical tape one.

Here is an example of using BOOT_CREATE:

OK, BOOT_CREATE 'TEST FINE>MFD>BOOT_LIST' -MT 0 -NO_QUERY
[Boot-create Rev 19.1]

No warnings

Assigning Tape drive 0 to your job

Device MT0 assigned.

Calling MAGSAV

[MAGSAV Rev. 19.2]

Tape unit (9 Trk): 0

Enter logical tape number: 1

Tape name: BOOT1

Date (MM DD YY):

Rev no:

Name or Command: \$A DOS

Name or Command: *DOS64

Name or Command: \$A CMDNC0

Name or Command: MAGRST

Name or Command: \$A CMDNC0

Name or Command: MAKE

Name or Command: \$A CMDNC0

Name or Command: FIXRAT

Name or Command: \$A CMDNC0

Name or Command: COPY_DISK

Name or Command: \$A CMDNC0

Name or Command: NSED

Name or Command: \$A CMDNC0

Name or Command: FUTIL

Name or Command: \$A CMDNC0

Name or Command: PHYRST

Name or Command:

Name or Command:

Name or Command: \$R

[MAGSAV Rev. 19.2]

Tape unit (9 Trk): 0

Enter logical tape number: 2

Tape name: BOOT2

Date (MM DD YY):

Rev no:

Name or Command: \$A MFD

Name or Command: CMDNC0

Name or Command: \$A MFD

Name or Command: PRIRUN

Name or Command: \$A MFD

Name or Command: SYSTEM

Name or Command: \$A MFD

Name or Command: SAD

Name or Command: \$R

Device released.

► CHAP

CHAP is an internal operator command that changes a user's time slice and priority level in the ready list.

The command format is:

```
CHAP { -userno } [priority [timeslice]]
      { ALL }
```

-userno The number of the user to be modified, in the form -nn.
(The priority and time slice of user 1 cannot be modified.)

ALL Specifies that changes are made to all users.

priority The priority level to be assigned to the user, in the
form of an integer from 3 (highest priority) to 0
(lowest priority).

timeslice New time slice value in tenths of a second. Must be
specified in octal.

If specified as 0, the time slice is reset to the standard value. If omitted, the time slice is left unchanged. If both priority and timeslice are omitted, the user's priority is reset to 1 (the level on which users normally run), and the time slice is reset to the standard value. The default value is '24 (20 decimal), or a two-second time slice.

A special time slice value, 177777, provides support for time-critical processes that require small amounts of CPU time separated by long idle periods, such as transaction processing.

When a time slice is set to 177777, the scheduler will allow the process to continue running as long as it likes. The process remains on the ready list until it waits for some occurrence, such as terminal input. The process can only be interrupted by a higher priority process that is eligible to run.

Setting any time slice to 177777 automatically sets the time slice for user 1 to the same value. This guarantees that the supervisor terminal will always get service.

Note

Setting the time slice to 177777 should never be used in an ordinary computing environment, since this facility gives a process uninterrupted and unlimited time. Such a process, if it performs many calculations, could produce an effect not unlike that of a system hang.

► CLOSE

The CLOSE command, when issued from the supervisor terminal, closes the specified file for all users. The format of the CLOSE command is:

CLOSE pathname

The pathname specified must reside on a partition of the local disk.

► CONFIG

The CONFIG command defines system parameters and defaults that are specified once per system session. The CONFIG command is disabled after its first use during a session. The CONFIG command in all its forms is fully described in the System Administrator's Guide.

► CONFIG_NET

19.3

CONFIG_NET builds the network configuration file. CONFIG_NET is described in detail in the Network Planning and Administration Guide.

► COPY_DISK

COPY_DISK is an external command that copies one disk to another and verifies the copy. It may be used from any terminal. COPY_DISK is fully described in Chapter 7.

COPY_DISK is invoked by the command:

COPY_DISK [options]

The options are:

OptionFunction

-NOVERIFY	Omits the verification of the steps of the COPY_DISK procedure. Decreases COPY_DISK time by approximately 60 percent. (This is the default.)
-----------	--

- DO_VERIFY Turns on verification of the steps of the COPY_DISK procedure.
- NO_BADS Turns off badspot handling, thus making partitions on the backup disk an exact copy of those on the source disk. Use this option if the source disk is full.
- TTY Causes all prompts issued by COPY_DISK that require a YES or NO answer to input the answer from the terminal, even if other input is coming from a command or CPL file.
- NOCHECKSUM Turns off checking of checksums. Allows copying between a disk written with a 4000 controller (an early model) and a disk written with a 4002 controller. To be used only with equipment having a 4000 controller.

Note

The early model 4000 controller generates different checksums than the 4002 model. If any equipment at your facility has the 4000 controller, use COPY_DISK's -NOCHECKSUM option when copying between disks written with the 4000 controller and disks written with a 4002 controller. This will enable the copying of disks without checking checksums. During this procedure, there is a small risk that bad information will be copied without detection.

- LOWEND Speeds up performance on machines smaller than the 750.

Caution

Use of the -LOWEND option with the Prime 750, 850, and 9950 will slow down the disk copying operation.

► DISKS

The DISKS command adds or removes the specified physical disk(s) to or from the Assignable Disks Table. The DISKS command may be given only from the supervisor terminal.

The command format is:

DISKS [NOT] pdev-0 [pdev-1] ... [pdev-7]

pdev-0 ... pdev-7 are physical device numbers. No more than ten disks may be entered into the Assignable Disks Table. A physical disk number must be specified in this table before a user can invoke the ASSIGN command to assign that disk.

When the optional argument NOT is specified in the DISKS command line, the specified physical disks are removed from the Assignable Disks Table. Removing a physical disk number from the table does not cause the disk to be unassigned; the operator must give the UNASSIGN command to unassign a disk from a user. For example, to add physical device numbers 20260, 50260, 60260, 70260, 10020, 110260, and 20262 to the Assignable Disks Table, type:

OK, DISKS 20260 50260 60260 70260 10020 110260 20262

These partitions may now be assigned by the users or operators.

To remove partition 20260 from the Assignable Disks Table and unassign the partition, type:

OK, UNASSIGN DISK 20260
OK, DISKS NOT 20260

► DPTCFG

The DPTCFG command compiles the configuration file for a DPTX system. (For details, see the Distributed Processing Terminal Executive Guide.)

► DPTX

The System Administrator uses the DPTX command to enable the Distributed Processing Terminal Executive System. DPTX allows the use of IBM3271/3277 terminals as Prime terminals and/or the use as IBM host terminals of 3271/3277 terminals or OWL 1200 terminals attached to Prime. For details, see the Distributed Processing Terminal Executive Guide.

► DROPDTR

This command applies to the DTR (Data Terminal Ready) signal associated with an asynchronous line. The DROPDTR command is useful only with a user line connected into a "port selector" or dialup modem. Issuing DROPDTR is allowed only while logged out. It will force the dropping of the DTR.

The DTR is normally dropped at the end of a grace time specified in the AMLTIM config directive. The grace time may be set as high as 10 minutes. Without using DROPDTR, a user connected to a port selector,

for example, who wished to disconnect and reconnect to another port, might have to wait as long as 10 minutes before the line became disconnected.

The System Administrator may force all DTR signals to be dropped at logout by including the DTRDRP directive in the CONFIG file. See the System Administrator's Guide for details.

► ELIGTS

ELIGTS is an internal command which modifies the eligibility time slice for system users. The command format is:

ELIGTS tenths

tenths is the time, in tenths of a second (in octal), that a user will run before being placed on the eligibility scheduler queue. The default value is 3 (about 1/3 second).

The default user time slice is 2 seconds, but a user will not remain on the ready list for this interval. When the ready list slice (default=1/3 second) of CPU time has been used, a user is moved from the ready list to the eligibility scheduler queue, and the user time slice is decremented by the ready time slice. The eligibility queue is checked by the scheduler after checking for interactive users (on the high priority queue) and before checking the low priority queues. Users cycle between the ready list and the eligibility queue until the user time slice is exhausted, at which time the user is entered in the low priority queues.

► EVENT_LOG

The command EVENT_LOG is used to turn system or network event logging off or on. The command format is:

```
EVENT_LOG [-NET] [ -ON
                  -OFF ]
```

If the -NET option is present, network logging is affected; otherwise, system logging is affected. If both -ON and -OFF are omitted, -ON is assumed.

EVENT_LOG is discussed in Chapter 5.

19.3

► FIND_RING_BREAK

FIND_RING_BREAK is a utility that allows you to determine where, between two active nodes of the RINGNET, a break has occurred. See the PRIMENET Guide for more information.

► FIX_DISK

FIX_DISK is an external command that checks (and optionally restores) the PRIMOS file integrity on any disk or partition. FIX_DISK is described fully in Chapter 9. The command format is:

FIX_DISK -DISK p_{dn} [options]

Note that -DISK p_{dn} must be included in the command line. p_{dn} is the physical device number of the partition on which FIX_DISK is to be run. See Appendix A for information on physical device numbers.

The options used with FIX_DISK fall into four categories:

- Options that specify repair tasks
- Options that modify terminal display
- An option that specifies the command device as the target
- Options for use with pre-Rev. 19 format disks

To Specify Repair Tasks: To specify repair tasks, use the following options:

<u>Option</u>	<u>Description</u>
-FIX	Directs FIX_DISK to perform the following modifications to the disk: correction of quota information, truncation or deletion of defective files, generation of a corrected DSKRAT if the current one is bad, and mapping of the badspot records to the BADSPT file. If omitted, no disk modifications are performed. Use this option whenever repair or conversion operations are to be performed.
-ADD_BADSPOT record-number ...	Allows you to specify record numbers to be added to the badspot file. FIX_DISK will attempt to map the bad record to a new location. Abbreviation: -ADBADS.

19.3

-AUTO_TRUNCATION Truncates directories nested too deeply in a directory tree. If omitted, `FIX_DISK` aborts if the maximum depth is reached. Maximum depth is set with the `-MAX` option (default is 99). Abbreviation: `-AT`.

Use this option to delete directories nested too deeply.

-DUFE Eliminates all inconsistent file entries or entries of unknown type. (The option `-DUFE` stands for Delete Unknown File Entries.) If omitted, no unknown file entries are touched, no UFDs containing unknown file entries are compressed, and `DSKRAT` is altered only to indicate which records are actually in use. This option is omitted to avoid the accidental deletion of valid file entries caused by running the wrong version of `FIX_DISK`.

Use this option to remove all file types unknown to this version of `FIX_DISK`. This option is particularly useful if `FIX_DISK` seems unable to finish an operation on a particular partition, as this inability to finish may indicate an unrecognizable file entry.

-INTERACTIVE Asks questions leading to construction of a consistent `DSKRAT` if the current `DSKRAT` is defective or missing. If omitted and the current `DSKRAT` is bad/missing, `FIX_DISK` will abort. The `-FIX` option must also be specified. Abbreviation: `-INT`.

Use this option if `FIX_DISK` has previously aborted and printed one of the following error messages:

The file structure of `DSKRAT` is bad.

The number of heads is different.

The physical record size is different.

The `DSKRAT` header has the wrong length.

-MAX_NESTED_LEVEL [n] Sets `n` (decimal) as the maximum depth to which directories can be nested. If omitted, the default maximum depth is 99 levels. Abbreviation: `-MAX`.

Use this option to set the maximum number of levels that directories may be nested.

19.3

- NUMBER_OF_RETRIES** number
Specifies the number of times FIX_DISK is to try to read a failing record. Each retry includes all nine different ways of reading a record. The default is two retries. Abbreviation: -NUMRTY.
- TRUNCATE**
This option can be specified if you prefer that FIX_DISK truncate a file when it encounters an uncorrectable badspot (as it did prior to Rev. 19.3). Normally, when FIX_DISK encounters an uncorrectable badspot, it will create a null record (a record filled with zeros) on a good portion of the disk and append the remaining records of the original file to the null record. Abbreviation: -TRU.
- UFD_COMPRESSION** Compresses UFDs by eliminating all entries for files/directories flagged as being deleted. Use of this option results in a decrease in the search time for UFDs. The -FIX option must also be specified. Abbreviation: -CMPR.
- Use this option to maximize the number of free records available on a partition.

To Modify Terminal Display: To specify terminal display modifications, use the following options:

<u>Option</u>	<u>Function</u>
-LEVEL [n]	Sets <u>n</u> (decimal) as the lowest level in the tree structure to be printed. When this option is omitted, the default value is level 1 (the MFD).
-LIST_BADSPOTS	Prints badspots and remapping records. Abbreviation: -LB.
-LIST_FILE	Prints filenames in all directories. Abbreviation: -LF.

To Specify the Command Device as the Target: To perform operations on the command device partition, use the following option:

<u>Option</u>	<u>Function</u>
-COMMAND_DEVICE	Indicates that FIX_DISK is to operate on the command disk. This option automatically closes all files in the command device, and causes all running phantoms to abort.

When this option is used, FIX_DISK must be invoked from the supervisor terminal. All users should be warned and logged out. Abbreviation: -COMDEV. |

When Operating on Pre-Rev. 19 Format Disks: When the disk you are running FIX_DISK on is a pre-Rev. 19 format disk, use one of the following options:

<u>Option</u>	<u>Function</u>
-NO_QUOTA	Assumes partition is not a quota partition; disables quota checking. Abbreviation: -NQ. Use this option when performing operations on a nonquota partition. This option must be used on all pre-Rev. 19 partitions, as these cannot be quota partitions.
-CONVERT_19	Converts the current partition to a PRIMOS Rev. 19 style disk: converts the BADSPT file to Rev. 19 format; initializes quota information; disables printing of warning/error messages related to quotas; creates a new rev stamp. The -FIX option must be used with -CONVERT_19.

► FIXBAT

FIXBAT is a utility for checking the Batch queue database integrity. It is normally supplied in UFD BATCHQ as FIXBAT.SAVE. See Chapter 11 for a complete description of FIXBAT.

The command format is:

```
RESUME BATCHQ>FIXBAT [options]
```

<u>Option</u>	<u>Meaning</u>
-DAYS n	Removes all cancelled, completed, or aborted jobs which are <u>n</u> or more days old from the Batch queues; sends a message to the terminal when a job has been removed. (<u>n</u> must be an integer between 0 and 60.) If <u>n</u> is 0, all nonactive jobs are removed from the queues.
-QUIET	Used with the -DAYS option to indicate that a message should not be sent to the terminal when FIXBAT removes a job from the queue.

-STARTUP argument Tells FIXBAT to start the BATCH monitor. When this option is used, FIXBAT assumes that it is being run by the BATCH -START command. That is, it assumes it is being run as a phantom from the supervisor terminal. The phantom that runs FIXBAT becomes the Batch monitor when FIXBAT is done.

The -STARTUP option takes one of four arguments: SAVE, SPOOL, DELETE, or NOLOG. These arguments tell FIXBAT what to do with the Batch comoutput file.

SAVE Renames the current comoutput log "OLDLOG" (deleting any existing "OLDLOG"). Creates a new comoutput file named O_LOG.

SPOOL Spools the current comoutput file, calling it BATCH.LOG. Creates and opens a new O_LOG file.

DELETE Opens O_LOG as a comoutput file. (The file is truncated when it is opened, destroying the existing contents.)

NOLOG Takes no action with regard to comoutput files.

► FTGEN

FTGEN is the File Transfer Service (FTS) command for the System Administrator. It allows the Administrator to configure the FTS system at a particular site, to initialize and validate the FTS database, and to display and modify the configuration as needed. FTGEN is described in the PRIMENET Guide.

► FTOP

The FTOP command is the operator's interface to the File Transfer Service (FTS). It allows the operator to start, stop, control, and monitor the status of FTS phantoms. It can only be invoked by users logged in as SYSTEM; two options, -START_MNGR and -START_SRVR, require invocation from the supervisor terminal in particular. The command format is:

FTOP option

19.3 |

The options -START_MNGR and -STOP_MNGR apply to the FTS manager process YTSMAN. All the other options apply to ordinary server processes.

The FTOP options are:

<u>Option</u>	<u>Function</u>
-ABND_SRVR	<p>Abandons an FTS server process. Causes the file transfer server <u>server_name</u> immediately to place all current file transfers on hold in the queue, and to log out. The format is:</p> <p style="text-align: center;">FTOP -ABND_SRVR <u>server_name</u></p> <p>If <u>server_name</u> is not running, an error message results. (Abbreviation: -ASV)</p>
	<p style="text-align: center;"><u>Note</u></p> <p>Forced logout of a server is not recommended. The recommended way to stop a server is to use the -STOP_SRVR option (see below).</p>
-ABRT_SRVR_LINK	<p>Aborts an FTS server link. Causes the file transfer server <u>server_name</u> to place the current file transfer on link <u>link_number</u> on hold in the queue. The server continues to run; it does not log out. The format is:</p> <p style="text-align: center;">FTOP -ABRT_SRVR_LINK <u>server_name</u> <u>link_number</u></p> <p>To find the link number of an active transfer, use the command option -LIST_SRVR_STS <u>server_name</u> (described below).</p> <p>If <u>server_name</u> is not running or <u>link_number</u> is not active, an error message results. (Abbreviation: -ASVL)</p>
-LIST_SRVR_STS	<p>Lists server status. Lists the status of the server <u>server_name</u>. The display indicates whether the server is currently active or not, as well as the status (for example, queued or transferring) of each of the eight possible file transfers that the server might be running. Each transfer is identified by a link number, in the range of 1 to 8. The format is:</p> <p style="text-align: center;">FTOP -LIST_SRVR_STS [<u>server_name</u>]</p>

If no server_name is specified, the status of all the configured servers is displayed. (Abbreviation: -LSVS)

-START_MNGR

Starts (phantoms) the FTS manager process YTSMAN. If manager_name is specified it overrides the default name YTSMAN.

Note

The command FTOP -START_MNGR should be invoked only from the supervisor terminal. This ensures that the manager is phantomized with the specified manager_name or with the default name of YTSMAN.

Invoking this command from a terminal other than the supervisor terminal results in an error message.

The format is:

FTOP -START_MNGR [manager_name]

The command to start up the manager may be added to the PRIMOS cold start C_PRMO file. (See Volume I of this book.) (Abbreviation: -STRMG)

-START_SRVR

Starts (phantoms) file transfer server server_name. If that server is already running, an error message is displayed.

Note

The command FTOP -START_SRVR should be invoked only from the supervisor terminal. This ensures that the server is phantomized with a user-id of server_name, and that the process priority and time slice parameters are automatically set in accordance with the configuration of the server (as configured in FTGEN).

If this option is invoked from a terminal other than the supervisor terminal, the system uses standard default values for server name, priority, and time slice, instead of using the values configured in FTGEN.

The command format is:

FTOP -START_SRVR server_name

The commands to start up the required file transfer servers may be added to the PRIMOS cold start C_PRMO file. (See Volume I of this book.)
Abbreviation: -STRSV

-STOP_MNGR

Logs out the File Server Manager after it has completed any transfer currently in progress. The format is:

FTOP -STOP_MNGR

If the manager is not running, an error message is displayed. A message is printed at the supervisor terminal when the manager is logged out. Abbreviation: -STPMG

Note

Closing down the FTS manager does not prevent locally queued transfer requests from being processed by any local servers that are running. Any remotely initiated transfers to and from the local node will not succeed. Such requests will be retried by the remote file transfer server at 30-minute intervals.

19.3

-STOP_SRVR

Stops an FTS server process. Causes the file transfer server server_name to log out after completing the file transfers it has in progress. The command format is:

FTOP -STOP_SRVR server_name

If the server is not running, an error message is displayed. A message is printed at the supervisor terminal when the server has shut down. Abbreviation: -STPSV

► FTR

The FTR command invokes the user interface to the File Transfer Service (FTS). It allows users to interactively request file transfers from local to remote sites. Files may be transferred either to or from the remote site. Requests may be made even when a communications link or remote computer is down because all requests are queued on the local computer. A user can display, modify, suspend, abort, or cancel a request once it is made. The use of FTR for submitting requests is described in the PRIMENET Guide.

The FTR command line format for managing requests is as follows:

FTR option [request-name] [control-argument...]

Control arguments are used only with the -MODIFY option, where they are required. The request-name identifies the particular file transfer request. Each file transfer request has associated with it a name and a number, either of which can be used to identify a particular request. The name is the name of the file to be transferred or a name specifically assigned by the submitting user (using the -NAME control arguments at the time of request submission). The number is assigned to the request by FTS to uniquely identify a request. The number is used to distinguish between two requests with the same name. You can find out the number of a request by means of the -STATUS option of the FTR command discussed below.

The operator may use any of the following FTR Request Management options on any submitted file transfer request when logged in as SYSTEM:

<u>Option</u>	<u>Description</u>
-ABORT	<p>Aborts a file transfer request. This option takes effect even if the transfer is currently in progress. An aborted request is placed on hold in the request queue. If the request is already in a request queue awaiting initiation, -ABORT is equivalent to -HOLD.</p> <p>If the request is already aborting, an error message results.</p> <p>An operator may abort any request.</p>
-CANCEL	<p>Deletes a request from a file transfer request queue. If the transfer is currently in progress, the request will <u>not</u> be cancelled.</p> <p>An operator may cancel any request.</p>

-DISPLAY Displays detailed information about a request. The display includes all the information given by the **-STATUS** option discussed below, in addition to all the information that is included in the request itself.

If request-name is not specified, all the requests owned by the user invoking the command are displayed.

If request-name is a name, all requests with that name belonging to the user are displayed.

An operator may invoke the command for any request-name. If request-name is not specified, all requests of all users are displayed.

-HOLD Delays file transfer initiation. This option applies to the specified request in the queue of requests waiting to be initiated. The request is not initiated until it is released via the **-RELEASE** option. If the specified request is already in progress, the command has no effect.

An operator may hold any request.

-MODIFY Modifies the characteristics of a submitted request. Once a request has been submitted (but before it has been initiated), most of its characteristics can be modified.

control-argument can be any of the FTR Request Submittal arguments except:

-QUEUE

-NO_COPY

-COPY

-DSTN_SITE

-SRC_SITE

-HOLD

(For a complete list of the FTR Request Submittal options and their meanings, see the PRIMENET Guide.)

Modifying the characteristics of a request is similar to canceling a request and resubmitting it. However, a modified request remains in the same position in the queue, whereas a cancelled and resubmitted request is deleted from the queue and then reentered in a new position.

An operator may modify any request.

-RELEASE Releases a file transfer request (request-name) that was previously held using the **-HOLD** option. The released request is made eligible for initiation. If the request is not being held, an error message will result.

An operator may release any request.

-STATUS Displays information about the current status of the request request-name. The following information is returned by **-STATUS** for each request:

- Date and time the request was queued
- User-id of the submitting user
- Name and number of the request
- Current status of the request

An operator may issue the FTR **-STATUS** command for any request. If the operator does not specify request-name, FTR displays all requests of all users.

19.3 | **-STATUS_ALL** Displays information about the current status of requests of all users.

► JOB

The **JOB** command allows the operator to monitor, hold, release, change, cancel, or abort a user's Batch jobs. **JOB** is discussed in Chapter 11. The command format is:

JOB [job-id] option

The options are as follows:

<u>Option</u>	<u>Function</u>
<u>-CANCEL</u>	Cancels a held or waiting job.
<u>-ABORT</u>	Aborts a running, held, or waiting job.

<u>-RESTART</u>	Terminates, then restarts a job.
<u>-HOLD</u>	Holds a job in the queue.
<u>-RELEASE</u>	Releases a held job so that it can run.
<u>-STATUS</u>	Displays the status of a job.
<u>-DISPLAY</u>	Displays status and parameters of a job.

The -HOLD and -RELEASE options are available to the operator only. When a job is held, it is still considered an active job, and it is counted in the list of waiting and held jobs given by the BATCH -DISPLAY command. However, it cannot run until it is released by the operator.

Holding a job is useful when it is known that a needed resource (such as magnetic tape, disk space, or the line printer) is not available. When the resource becomes available, the job can be released by the operator with the command:

```
JOB job-id -RELEASE
```

For the Batch subsystem, an operator is user SYSTEM or user BATCH_SERVICE. However, to use the -ABORT and -DISPLAY options of the JOB command, the supervisor terminal must be used if the job does not belong to the user issuing the command.

► { LIST_ACCESS }
 LAC

Lists the access rights for any object. The command format is:

```
LIST_ACCESS [objectname]
```

LIST_ACCESS is discussed in Chapter 2.

► { LIST_GROUP }
 LG

Lists the ACL groups to which the user belongs. LIST_GROUP is discussed in Chapter 2.

► { LIST_PRIORITY_ACCESS }
LPAC

Reads the contents of a priority ACL on a disk partition. LIST_PRIORITY_ACCESS is discussed in Chapter 2.

► LOGOUT

This command logs out a specified process. Files are closed, devices are unassigned, attach points (initial, home, and current) are cleared, and assigned segments are returned to the supervisor. The command format is:

LOGOUT { -usernumber }
ALL

-usernumber is the decimal number of the terminal being disconnected. If the user is a local terminal using a remote process, the terminal is reconnected to its local process. If the user is a remote terminal using a local process, the process is logged out and returned to the pool of free remote login processes.

ALL represents all currently logged-in remote, local, and phantom users, except for the supervisor terminal, NETMAN, and the FAM, if it is a running process. In addition to logging out all users, this command automatically sets MAXUSR to 0, preventing any subsequent logins until MAXUSR is reset. LOGOUT ALL can be issued just prior to a SHUTDOWN ALL command to allow a more orderly shutdown of PRIMOS. LOGOUT ALL can be issued only from the supervisor terminal. (See also the MAXUSR command.)

► LOOK

LOOK is an internal operator command that provides access to any user segment in the system. The LOOK command can be issued only by the system operator and must be preceded by an OPRPRI 1 command and followed by an OPRPRI 0 command. The command format is:

LOOK [-userno [segno [access [mapseg]]]]

-userno Number of the user owning the segment. Default is user 1.

segno Number of the segment to be examined. Default is '6000 (the Ring 0 stack segment for the user).

access Access rights to be granted (as in the SHARE command).
Default is '200 (read-only).

mapseg Segment of user 1's address space into which the specified
segment is to be mapped. Default is '4001.

Caution

This command is intended mainly as a debugging tool for the use of systems engineers and field analysts. The operator and administrator will normally have no use for it.

If the LOOK command involves an attempt to examine a segment that does not exist, an attempt to write to a segment that does exist, or attempts to map either shared or stack segments with write permission, the command is considered risky or dangerous to system integrity. The REALLY? prompt is issued for any LOOK command whose request is considered to be risky or dangerous to system integrity. A YES response allows the operation to proceed. To undo a LOOK command, specify 0 for segno.

See also the SHARE command for system segment access.

► MAGRST

Restores a disk file, directory tree, or partition from a magnetic tape written by MAGSAV. MAGRST is fully described in Chapter 8 and in the Magnetic Tape User's Guide. The command format is:

MAGRST [options]

After you give options (if any) on the command line, MAGRST responds with a series of questions. The MAGRST dialog (questions and appropriate user replies) are discussed in Chapter 8 and in the Magnetic Tape User's Guide.

<u>Option</u>	<u>Function</u>
-7TRK	Specifies 7-track format. The default is 9-track.
-TTY	Takes the tape unit number from your terminal. All other information is taken from the current input stream. You use this option with CPL files and command input files.

Note

There is no option that restores a tape without restoring the ACL information saved on the tape.

After MAGRST is invoked, and you have supplied the tape unit and logical tape numbers, the tape label will be read and displayed. MAGRST will then issue the prompt:

Ready to restore:

Allowable responses to this prompt are as follows:

<u>Response</u>	<u>Meaning</u>
YES	Causes MAGRST to restore the entire tape. When it is finished, it will return you to PRIMOS.
NO	Prevents MAGRST from restoring the tape, and causes MAGRST to prompt you once again for the tape unit and logical tape number.
\$I [pathname] [n]	Causes MAGRST to produce an index. The Ready to Restore: prompt is then repeated. The index is sent to the file specified by <u>pathname</u> , or to the user terminal if <u>pathname</u> is not specified. <u>n</u> specifies the number of directory levels (in octal) to be included in the index — the default for <u>n</u> is 2.
	When MAGRST has finished restoring the tape, it will exit to PRIMOS.
\$A pathname	Changes your current directory to the directory specified by <u>pathname</u> . Similar to the ATTACH command of PRIMOS.
NW [filename] [n]	Causes MAGRST to read the entire tape and produce an index, <u>without</u> restoring any files to disk. This is useful when you just want to see what is on the tape. The index is sent to the file <u>filename</u> in the current directory, or to the user terminal if <u>filename</u> is not specified. <u>n</u> specifies the number of directory levels to be included in the index — the default for <u>n</u> is 2.

The NW command does not accept pathnames. To write an index to a file in a different directory than the current one, use the \$A command to change the current directory, then issue the NW command.

When the tape has been read, MAGRST will exit to PRIMOS.

PA

Causes MAGRST to restore only certain files and directories — a partial restore. MAGRST will ask for pathnames of files and directories to be restored, using the prompt:

Tree name:

In response to this prompt, enter the pathname of the file or directory you wish to restore. For example: MFD>JOHN>LOST_FILE. Each time you enter a pathname, MAGRST will prompt for another one, until you enter a blank pathname by hitting the carriage return (CR) key. A maximum of ten pathnames per restore is permitted.

After you enter the blank pathname to terminate the list of files and directories you wish restored, MAGRST will read the entire tape, and restore the files and directories present in the list. When MAGRST has finished reading the entire tape, it will exit to PRIMOS.

► MAGSAV

Writes a disk file, directory tree, or partition to magnetic tape. MAGSAV is fully described in Chapter 7, and in the Magnetic Tape User's Guide. The command format is:

MAGSAV [options]

The options are shown on the following pages.

<u>Option</u>	<u>Function</u>
-7TRK	Specifies 7-track tape format. The default is 9-track.
-LONG	Specifies a 2048-byte record size. The default is 4096-byte variable length records.
-P300	Specifies 1024-byte records. Also suppresses ACLs.
-VAR	Writes variable-length records up to 4096 bytes. (This is the default.) -VAR is useful for large files, as it decreases the amount of tape used for record headers.
-UPDT	Specifies an update. The DUMPED switch in the UFD entry will be set for files and directories that are saved from disk onto tape. If you do not specify this option, the DUMPED switch is not set.
-INC	Specifies an incremental save. Only files and directories with a reset (=0) DUMPED switch are saved. Otherwise, all files and directories are saved.
-SAVE_UFD	Tells MAGSAV always to save directories, whether or not they have been modified. This option is only used with the -INC option. The -SAVE_UFD option causes directory headers to be saved with the individual files. Abbreviation: -SUFD.
-TTY	Takes the tape unit number from your terminal. All other information is taken from the current input stream. You use this option with CPL files and command input files.
-NO_ACL	Specifies that MAGSAV is not to save any ACLs or ACL references. Tapes saved with the -NO_ACL option can be restored by Rev. 18 MAGRST onto a Rev. 18 system. If this option is not specified, ACL information is saved to tape, as explained below. Abbreviation: -NOA.

After MAGSAV is invoked, and you have supplied the tape unit, logical tape numbers, and tape label information, MAGSAV will then issue the prompt:

Name or command:

Allowable responses to this prompt are as follows:

<u>Response</u>	<u>Meaning</u>
filename	Causes MAGSAV to save the specified file or directory residing in the current directory.

*	Causes MAGSAV to save all the files and directories in the current directory.
MFD	Causes MAGSAV to save the entire partition. To use this response, you must be attached to the MFD of a partition.
\$I [pathname] [n]	Causes MAGSAV to produce an index as it saves subsequent files and directories. The index is sent to the file specified by <u>pathname</u> , or to the user terminal if <u>pathname</u> is not specified. <u>n</u> specifies the number of directory levels to be included in the index — the default for <u>n</u> is 2.
\$Q	Terminates the logical tape and returns you to PRIMOS. Does not rewind the tape. If you subsequently invoke MAGSAV for the same tape unit, specify 0 as the logical tape number.
\$R	Terminates the logical tape, rewinds the tape, and returns you to PRIMOS. If you subsequently invoke MAGSAV for the same tape unit, specify a logical tape number of one more than the logical tape number of the session you just completed.
\$UPDT ON	Turns on update mode. After this command is issued, files that are saved will have the DUMPED switch set. This command is the same as the -UPDT command line option.
\$UPDT OFF	Turns off update mode. This overrides any specification of -UPDT on the command line. After this command is issued, files that are saved will not have the DUMPED switch set.
\$INC ON	Turns on incremental dump mode. If a directory is saved after this command is issued, each file inside it will be saved only if its DUMPED switch is not set. This command is the same as the -INC command line option.
\$INC OFF	Turns off incremental dump mode. This overrides any specification of -INC on the command line. After this command is issued, all files inside specified

directories will be saved, independent of the setting of their DUMPED switches.

\$A pathname

Changes your current directory to the directory specified by pathname. Similar to the ATTACH command of PRIMOS.

\$TTY

Allows you to enter a new tape unit number from the terminal during the course of a save. Especially helpful if an unexpected End-of-Tape occurs from a command file or CPL program.

19.3

► MAKE

MAKE creates a structure for any PRIMOS-supported disk pack or partition. MAKE is described fully in Chapter 6. The command format is:

MAKE -DISK pdev -PARTITION name [options]

The -DISK option is required. If pdev or name are not specified, MAKE will prompt for them. pdev is the physical device number of the disk to be created, and name is the name to be assigned to the disk. See Appendix A for information on constructing physical device numbers.

Caution

Do not forget to include the -DISK option on the command line, preceding the pdev argument. Otherwise, PRIMOS (or PRIMOS II) will interpret pdev as the starting address of the MAKE program, and will not start at location '1000 as expected by MAKE. The results almost certainly will be unpleasant, and potentially harmful to existing data if you are running MAKE under PRIMOS II, which provides no protection against writing on the wrong disk.

When using MAKE under PRIMOS II, you should physically write-protect all disk drives except the one on which MAKE is to operate.

The options are shown on the following pages.

<u>Option</u>	<u>Description</u>	
-DISK_TYPE [type]	Specifies the type of disk. Legal types are SMD, CMD, 68MB, 158MB, 160MB, 600MB, MODEL_4475, and FLOPPY. If <u>type</u> is not specified, the user will be prompted for it. The default is SMD.	19.3
-MAP_UNCORR	Specifies that records with correctable errors are not to be considered badspots. The default is to consider records with correctable errors to be badspots, in addition to those records having uncorrectable errors. (This does not apply to floppy disks.)	
-SPLIT [recs]	Specifies that the partition is to be split -- some of it is to be used for file storage, some is for paging storage. <u>recs</u> specifies the number of records to be reserved for paging storage. If <u>recs</u> is not specified, the total number of available records will be displayed, and the user will be prompted for <u>recs</u> . The default is to use all of the available storage for file storage and none for paging storage. (This does not apply to floppy disks.)	
-NEW_DISK	Specifies that the disk partition is being made for the first time. This prevents MAKE from searching for an existing badspot file. Abbreviation: -NEWDSK	
-COPY_BADSPOTS_BY_NAME partition	Specifies that MAKE is to copy badspot information from the named partition. Useful when each badspot file on a disk pack contains information on all badspots in the pack, not merely on badspots in its own partition. The named partition must reside on the same pack as the partition being made, and must be a started-up disk (ADDISK). The default is for MAKE to read the badspot file only on the partition being made. Abbreviation: -CPYNAM.	19.3
-COPY_BADSPORT_BY_DEVICE pdev	Specifies that MAKE is to copy the badspot file from the designated physical device on the same disk pack as the partition being made. Useful when every badspot file on the pack carries information about badspots on all partitions on the pack. The default is for MAKE to read the badspot file only on the	

19.3

partition being made. The device must be ASSIGNED if this option is used. Abbreviation: -CPYDEV.

- FORMAT Specifies that formatting must be performed on the disk before any other operations are attempted. This must be specified if the disk has never before been used on a Prime system. (This does not apply to floppy disks.) The default is to perform no formatting.
- QUERY_BADSPOTS Causes MAKE to prompt the user for known badspots on the disk. The default is to not prompt the user for this information. (This does not apply to floppy disks.)
- PRE_REV19 Required if the partition to be made is to be used on Prime systems earlier than Rev. 19. The default is to make a Rev. 19-format disk. (This does not apply to floppy disks.)
- BADSPOT_LEVEL [n] Specifies the level of verification to be employed in checking for badspots. n can be from 0 to 4, inclusive. If n is 0, no verification will take place. The larger n is, the more verification is performed, and the longer MAKE takes to run. If n is not specified, the user will be prompted for this information. The default for n is 1 if the disk type is an SMD or CMD, and 4 for FMD disks (all other types). (This does not apply to floppy disks.)
- BAUD_RATE [bps] Sets the initial baud rate of the supervisor terminal for the disk. The baud rate takes effect when the disk being made is used to boot the system. bps can be 110, 300, 1200, or 9600. If bps is not specified, the user will be prompted for this information. The default is 300 bits per second.
- NO_INIT This rarely-used option specifies that MAKE is not to initialize the file system portion of the disk. The default is to initialize.

► MAXSCH

MAXSCH controls the amount of overlapped processing performed by the system by controlling the number of processes on the ready queue and the fast I/O wait queues. The command format is:

MAXSCH n

n is the value of the variable MAXSCH in SUPCOM; the default value depends upon factors such as the amount of main memory on the system and the type of CPU.

► MAXUSR

MAXUSR is an internal command that is issued during coldstart to allow users to log in. If it is issued before the system date and time have been set, it will be ignored.

The first time the MAXUSR command is issued after the system date and time have been set, the "Login please" message will be issued at all terminals connected to the system via eligible asynchronous lines. (Terminals connected via assignable asynchronous lines will receive no such message.) Once this happens, the system will respond to commands issued from those terminals.

Until the first MAXUSR command is issued, PRIMOS may echo characters typed on those terminals, but will not respond to them until the MAXUSR command is issued. The MAXUSR command also controls the number of users that are allowed to be logged into PRIMOS at any time.

The command format for MAXUSR is:

MAXUSR [n]

n is the maximum number of allowable users (octal). If not specified, it defaults to the number of configured users.

Even if MAXUSR 0 is issued, the "Login please" message will be sent to user's terminals; however, they will be unable to log in until a higher value of n is specified in a MAXUSR command.

When it is necessary to limit the number of users on the system, MAXUSR can be used to specify that no more than n users will be allowed to log in. If the number of users is already above n, no user will be forcibly logged out, but no new users will be allowed to log in and no phantoms will be started except those started from the supervisor terminal (User 1). If other users attempt to log in or start a phantom they will receive the error message "Max number of users exceeded."

This facility will be useful when the system is to be shut down, setting MAXUSR to 0 will prevent new logins just before the shutdown. LOGOUT ALL logs out all users and also sets MAXUSR to 0.

► MESSAGE

MESSAGE is an internal command used to send messages to users or to other operators. Both users and operators may send messages.

Messages may be sent from:

- Any user terminal to any user terminal.
- Any user terminal to the supervisor terminal.
- The supervisor terminal to all users.
- The supervisor terminal to a specified user.
- The supervisor terminal to another supervisor terminal on a different node on the network.

Operator-to-User Messages: The format of an operator-to-user message is:

$$\text{MESSAGE} \left\{ \begin{array}{l} \text{username} \\ \text{-user number} \\ \text{ALL} \end{array} \right\} [\text{option}]$$

text of message

username is a user-id. All users logged in as username receive the message.

usernumber is the number of a specific terminal line. Only the terminal specified as usernumber receives the message. To determine the user numbers for the various terminals, issue the STATUS USER command.

ALL represents all users on the system. All users receive the message.

text of message is a single line to be sent. Sending a message produces two lines of information on the receiver's terminal. The top line identifies the sender and the time the message was sent; the second contains the text of the message. For example:

```
*** SYSTEM at 09:28
White forms will be printed at 12:00 today.
```

Options of the MESSAGE command are:

-NOW Prints the message immediately. This is undesirable if the user is in the middle of a sensitive operation.

If **-NOW** is not specified, the message is stored in a broadcast buffer (ALL) or a single user buffer. The message is printed at the user's terminal when that user returns to PRIMOS command level. A message that is in the broadcast buffer is also printed after a

user issues the LOGIN command. To cancel a broadcast message, issue another broadcast message with a null line as text-of-message.

When -NOW is specified, stored messages are not affected.

-FORCE Allows the operator to override a user's receive state, sending the message no matter what setting of the receive state. See the section on receive states below.

If -FORCE is not used, reception of the message may be blocked by some or all of the intended recipients. If this happens when ALL (sending to all users) was specified, the user numbers of all users who are blocking receipt of the message will be printed. If this happens when sending to a userid or username, the operator will be asked if the message should be forcibly sent to each user who is blocking receipt of the message.

-ON nodename The message will be sent to the specified user or users on the system nodename rather than the local system.

Operator-to-Operator Messages: The format of an operator-to-operator message is:

```
MESSAGE -l -ON nodename [-NOW] [-FORCE]
text of message
```

When this format is used, text of message is printed at the supervisor terminal of the node specified by nodename.

If the nodename given is not connected to the local node, then the error message "Bad parameter. (MSG\$)" will be printed after the operator enters the message to be sent.

Note

The nodename of the originating system is not printed at the receiving system. The sending operator should include the originating nodename as part of the message text.

Receive States: Users may set the receive state of their terminal with the -ACCEPT, -DEFER, and -REJECT arguments of the MESSAGE command. The arguments control the flow of messages according to the following pattern:

MESSAGE option

The options are:

<u>Option</u>	<u>Function</u>
<u>-ACCEPT</u>	Enables reception of all messages
<u>-DEFER</u>	Inhibits immediate messages; accepts messages sent without the <u>-NOW</u> option
<u>-REJECT</u>	Inhibits all messages

The receive state of each user is determined using the -STATUS option of the MESSAGE command. For example:

OK, MESSAGE -STATUS

User	No	State
SYSTEM	1	Accept
EEYORE	2	Reject
POOH	8	Defer
ROO	10	Accept
BATCH_SERVICE	102	Accept
YTSMAN	105	Accept
FTP	106	Accept
SYSTEM	109	Accept

Either of the following command lines may be used to determine the message status of an individual user:

MESSAGE -STATUS username
MESSAGE -STATUS usernumber

To determine the receive state of the terminal you are using, the command format is:

MESSAGE -STATUS ME

► MONITOR_RING

19.3

MONITOR_RING allows you to check the throughput and status of the RINGNET network from the point of view of this node. See the PRIMENET Guide for information on MONITOR_RING.

► NET

19.3

This is a half-duplex command and is not supported at PRIMOS Revision 19.3.

NET controls half-duplex PRIMENET on previously configured SMLC/MDLC lines. This command can be executed only from the supervisor terminal. NET is discussed in Chapter 4. For more details on NET see the PRIMENET Guide. The command format is:

NET option

Four options allow you to control the state of the half-duplex (HDX) connection. These options are summarized below:

<u>Option</u>	<u>Function</u>
-ASSIGN SMLCnn	<p>Reserves a previously configured HDX synchronous line for HDX operations. Assigning a line does not initiate a connection to the remote system. The -START option, which is issued after the -ASSIGN option, is used to initiate a connection.</p> <p><u>nn</u> is the number of the HDX line, as configured in NETCFG.</p> <p>Once a line is assigned, it belongs to PRIMENET and cannot be used by any other process until it is unassigned by the NET -UNASSIGN command.</p>
-START SMLCnn [-SITE node]	<p>Activates an HDX line. The line must be assigned by the -ASSIGN option before the -START option can be used.</p> <p><u>nn</u> is the number of the assigned HDX line to be activated. <u>node</u> is an HDX node (as configured in NETCFG).</p> <p>The -SITE option determines whether the line being started will be used to make or to receive calls. If -SITE is included, the line will be used to originate calls to <u>node</u>. If -SITE is omitted, the line will be used to receive calls from any HDX nodes that have been configured.</p>

Only the operator sending a call may use the -SITE option. Thus, the operators on the two systems involved must determine ahead of time which one will initiate calls. Unless the operator on one system uses -START with -SITE and the other uses -START without -SITE, the communications link will fail. Should this occur, the operator(s) in error must issue the NET -STOP command and then reissue the correct NET -START command.

-STOP { SMLCnn
 nodename }

Disconnects an HDX line. A line that is disconnected using the -STOP option enters one of two states, depending on how the line was started. If the -START command was issued without the -SITE option, the line returns to a started state, ready to accept another incoming call from any legal HDX remote node. If the -START option was issued with the -SITE option, the line returns to an assigned state, still reserved for HDX use. The line must then be started again before another call can be made.

nn is the line number of the HDX line to be disconnected. nodename is the name of a currently active node whose line is to be disconnected. Either line numbers or nodenames may be used to specify a line, regardless of how the line was started.

-UNASSIGN SMLCnn

Unassigns an assigned line, removing it from its reserved state. The unassigned line is no longer reserved for HDX PRIMENET's use, and may be used by other processes (for example, RJE or DPTX). To use the line for HDX again, you must reassign it.

A line must not be in a started state when the -UNASSIGN option is issued. You must issue the NET -STOP command before unassigning the line with NET -UNASSIGN.

► NETCFG

19.3 | At Revision 19.3, NETCFG has been replaced by CONFIG_NET.

► OPRPRI

OPRPRI is an internal command issued from the supervisor terminal that is implemented as a check against inadvertent or unauthorized use of SHARE and other commands that might adversely affect the system. The command format is:

OPRPRI n

n is either 1 or 0.

- 1 Removes safeguard and allows the SHARE command to be given.
- 0 Resets safeguard against issuing SHARE command inadvertently. This is the default.

► PHYRST

Restores to disk, partitions that have been saved by PHYSAV on magnetic tape. PHYRST is described fully in Chapter 8, and in the Magnetic Tape User's Guide. The command format is:

PHYRST [-TTY] [-NO_BADS] [-UNMOD]

The options are:

<u>Option</u>	<u>Function</u>
-TTY	Causes PHYRST to request its magnetic tape user number from the terminal, even if the utility is being run from a CPL or COMINPUT file.
-NO_BADS	Disables badspot handling. If this option is not given, PHYRST does handle badspots, as explained in the section on badspot handling in Chapter 8. (Badspot handling is available on Rev. 19 disks only.)
-UNMOD	Prevents system hangs due to incorrect recovery from DMX overruns. Use this option only if your equipment has one of the following early model controllers: wire wrap disk controller boards without ECR 3748, or etched boards without ECRs 3062 and 3342.

► PHYSAV

Writes to magnetic tape the contents of one or more assigned disk partitions. The copy is made track by track, in physical order. PHYSAV is described fully in Chapter 7, and in the Magnetic Tape User's Guide. The command format is:

PHYSAV [options]

The options are:

<u>Option</u>	<u>Function</u>
-TTY	PHYSAV asks for the magnetic tape unit number at the terminal, even if it is running from a CPL or COMINPUT file. Do <u>not</u> use the -TTY option if you are planning to run the command file as a Batch job or phantom, or if you want PHYSAV to take the magnetic tape unit number from the command file.
-LOWEND	Use with machines other than a Prime 750, 850, or 9950.
-UNMOD	Prevents system hangs due to incorrect recovery from DMX overruns. Use this option only if your equipment has one of the following early model controllers: wire wrap disk controller boards without ECR 3748, or etched boards without ECRs 3062 and 3342.
-SPEED { 25 } { 100 }	Used only when writing to a streamer tape drive. With these drives, speed automatically defaults to 100 ips, unless you give the -SPEED 25 option or assign the drive at a density of 3200 bpi. (In the latter case, speed is set to 50 ips.)

19.3

► PRIMOS

PRIMOS is the command used by the operator to boot PRIMOS. See Volume I of this book for information on how to start up PRIMOS. For a description of the PRIMOS command, which may only be invoked under PRIMOS II, see Chapter 14.

► PRINT_NETLOG

The PRINT_NETLOG utility analyzes a network event log file and produces a formatted output file that chronicles the network events represented in the file. PRINT_NETLOG is discussed in Chapter 5. Appendix I contains a list of network event log messages output by PRINT_NETLOG. The command line to invoke PRINT_NETLOG is:

```
PRINT_NETLOG { [output-file] } [options]
               TTY
```

If TTY is specified, output will be written to the user's terminal. If neither TTY nor output-file is specified, output will be written to the file NETLST in the current directory. Otherwise, output-file is the name of the file to which the output is to be written.

The input file, containing a binary-encoded log of network events, is specified using the -INPUT option, described below. If the -INPUT option is not included on the PRINT_NETLOG command line, PRINT_NETLOG will use the most recently created log file in the UFD named PRIMENET*. Network event log filenames are of the form NET_LOG.mm/dd/yy. If PRINT_NETLOG is unable to find a network event log file, it will prompt for an input file name.

The following options are available:

<u>Option</u>	<u>Description</u>
<u>-HELP</u>	Print a list of PRINT_NETLOG options. The PRINT_NETLOG command must be retyped after the options are printed.
<u>-INPUT</u> treename	Specifies the treename of the input log file to be processed. If this option is not present on the command line, PRINT_NETLOG will attempt to use the most recently created network event log file, as described above.
<u>-FROM</u> { mmddyy [hhmm] } TODAY	Only entries from the specified date to the latest entry are processed. Specify TODAY instead of mmddyy to refer to today's date. Following the date specification, an optional time specification of the form hhmm (hours, minutes) may be entered. A time entry may be between 0000 and 2359. Omitting the time specification is equivalent to specifying '0000'. PRINT_NETLOG checks <u>each</u> entry individually to see if its date/time stamp indicates

that it should be formatted. An out-of-sequence entry (for example, the wrong date entered by the operator) will not turn on entry formatting prematurely.

-TYPE type type ...

Process entries only of the indicated types. The types can be any of the following:

Network Event Types

COLD	Cold starts
WARM	Warm starts
TIMDAT	Time/date entries
RESET	Circuit resets
BADSEQ	Packets out of sequence
OVERFL	NETBUF overflow entries
SHUTDN	Operator shutdowns
LPE	Local procedure errors
RING1	Tokens inserted into the ring
RING2	Ring DIM out of receive blocks
RING3	Ring nodes not accepting transmits
NETDMP	NETDMP calls
SMLC1	SMLC status errors
SMLC2	SMLC - no STX preceding ETX
SMLC3	No system blocks for SMLC protocol messages
SMLC4	SMLC resets
HOSTDN	Level III protocols down
PWFAIL	Power fail checks
INCREQ	Incoming call request
OUCREQ	Outgoing call request
REMARK	Operator remark
NPXTHR	NPX throttled on transmit/receive
NPXRCV	NPX unexpected receiver status
NPXCLR	NPX master circuit was cleared
NPXSEQ	NPX message out of sequence
NPXCON	NPX unknown circuit status
NPXRLS	NPX bad virtual circuit clearing
RNGRCV	PNC spurious receive interrupt
RNGHRD	PNC hardware failure
RNGRES	Resource failure
SMLC5	Indicates a CMDR has been sent
SMLC6	Internal error has been encountered
RNDTMT	Ring receive timeout
DIAPKT	Diagnostic packet received
ICS1.1	Dconfigure code word was not queued
ICS1.2	Logical connection deleted

ICS1.3	Logical connection could not be deleted
ICS1.4	LCAD1_ was not found in the LCB
ICS1.5	IPQNM IE\$FRM error was received
ICS1.6	Flush Timeout
ICS1.7	Illegal flush was completed
ICS1.8	Line is not assigned
ICS1.9	Unidentified ICS error occurred
ICS1.0	Line is not defined

19.3

Note that the time/date stamps associated with the selected entries will not be processed unless TIMDAT is explicitly selected, for example, '-TYPE DISK TIMDAT' will process all disk errors and their associated time/date stamps. If TIMDAT alone is specified, all time/date stamps will be processed. If TIMDAT is specified in conjunction with one or more other types, only the time/dates of the selected types will be processed. If the -TYPE option is not specified, all entries will be processed.

-SPOOL

Spool the output file when done. PRINT_NETLOG will print the name of the output spool file.

-DELETE

Delete the output file when done. This option should only be specified when the -SPOOL option is also specified.

-PURGE

Empty, but do not delete, the event log input file when event log processing is complete. Write access is required on the input file.

-CENSUS

PRINT_NETLOG totals the entries for each event in the input file and writes the totals to the output file or terminal. Only non-zero totals are displayed.

-CONTINUE

Continue after a bad entry is found. PRINT_NETLOG will normally halt if an invalid entry is encountered. If this option is specified, PRINT_NETLOG will continue processing in an attempt to find the next valid entry.

-DEBUG

This option causes PRINT_NETLOG to read entries from the terminal and can be used for testing PRINT_NETLOG's formatting for entry types. Each entry should be entered as a series of tokens (using RDTK\$\$'s rules). Octal tokens are converted to binary; all others are taken as ASCII strings. PRINT_NETLOG leaves this mode of operation whenever a 'Q', 'q', or null line is entered.

-REMARK text

Enter an event of type REMARK directly into the input file. This can be used, for example, by an operator who wishes to record an observation on some event that might affect the subsequent operation of the network. All text after the -REMARK option is taken as the text to be entered into the input file. Consequently, the -REMARK option must be the last option specified on the command line. The message can be up to 80 characters in length and need not be surrounded by apostrophes. Write access is required on the input file.

-DUMP

In addition to its normal formatting, PRINT_NETLOG will dump each entry processed in octal. This option is provided as an additional aid to those who define their own event types. Only those entries that have been selected for processing are dumped.

PRINT_NETLOG Messages: The following messages may be displayed by PRINT_NETLOG.

- UNKNOWN CPU MODEL xx.

A CPU model number was encountered with which PRINT_NETLOG is not familiar. PRINT_NETLOG generates a warning message and continues processing, treating the CPU model number as 0.

- BAD ENTRY ENCOUNTERED IN FILE 'logfile'.

The event log file logfile contains an entry which is not defined by PRINT_NETLOG.

- DEFAULT INPUT FILE NAME NOT CONSTRUCTED.

A file of the type NET_LOG.mm/dd/yy could not be found in PRIMENET* or the top level directories did not exist or the user had insufficient access.

- INPUT LOGGING FILE filename NOT FOUND.
UFD <0>PRIMENET* CONTAINS LOGGING FILE FOR THE LATEST COLD START.
ENTER INPUT FILE NAME (ENTER CR TO QUIT):

The user did not explicitly specify an input event file and PRINT_NETLOG cannot find a file with a name of the format NET_LOG.mm/dd/yy. PRINT_NETLOG prompts the user for the input file name.

- OK TO DELETE FILE filename? ANSWER: 'Y' OR 'N'!

The output from PRINT_NETLOG may be directed to a file or to the terminal. If file output is desired, and PRINT_NETLOG finds that the output file already exists, this message is printed. The reply should be 'Y' to delete the file or 'N' to enter a new destination. If 'N' is entered, the message

NEW OUTPUT FILE NAME:

is printed. The user should enter a pathname; if not, PRINT_NETLOG will continue to query the user with the preceding message sequence asking if the existing output file is to be deleted or what will be the name of the new output file. Entering TTY will cause output to the terminal.

- OUTPUT HAS BEEN PLACED IN FILE 'filename'

PRINT_NETLOG has completed the processing of the event entries and all other file manipulation requested by the user. This message is generated provided that output was directed to a file.

► PRINT_SYSLOG

The PRINT_SYSLOG utility analyzes a system event log file and produces a formatted output file that chronicles the system events represented in the file. PRINT_SYSLOG is discussed in Chapter 5. Appendix H contains a list of system event log messages output by PRINT_SYSLOG. The command line to invoke PRINT_SYSLOG is:

```
PRINT_SYSLOG { [output-file] } [options]
               TTY
```

If TTY is specified, output will be written to the user's terminal. If neither TTY nor output-file is specified, output will be written to the file LOGLST in the current directory. Otherwise, output-file is the name of the file to which the output is to be written.

The input file, containing a binary-encoded log of system events, is specified using the -INPUT option, described below. If the -INPUT option is not included on the PRINT_SYSLOG command line, PRINT_SYSLOG will use the most recently created log file in the UFD named LOGREC*. System event log filenames are of the form LOG.mm/dd/yy. If PRINT_SYSLOG is unable to find a system event log file, it will prompt for an input file name.

The following options are available:

<u>Option</u>	<u>Description</u>
<u>-HELP</u>	Print a list of PRINT_SYSLOG options. The PRINT_SYSLOG command must be retyped after the options are printed.
<u>-INPUT</u> treename	Specifies the treename of the input log file to be processed. If this option is not present on the command line, PRINT_SYSLOG will attempt to use the most recently created system event log file, as described above.
<u>-FROM</u> { mmddyy [hhmm] } TODAY	Only entries from the specified date to the latest entry are processed. Specify TODAY instead of <u>mmddyy</u> to refer to today's date. Following the date specification, an optional time specification of the form <u>hhmm</u> (hours, minutes) may be entered. A time entry may be between 0000 and 2359. Omitting the time specification is equivalent to specifying '0000'. PRINT_SYSLOG checks <u>each</u> entry individually to see if its date/time stamp indicates

that it should be formatted. An out-of-sequence entry (for example, the wrong date entered by the operator) will not turn on entry formatting prematurely.

-TYPE type type ...

Process entries only of the indicated types. The types can be any of the following:

System Event Types

COLD	Cold starts
WARM	Warm starts
TIMDAT	Time/date entries
CHECKS	Machine checks (including memory parity)
POWERF	Power fail checks
DISK	Disk errors
DSKNAM	ADDISK entries
OVERFL	LOGBUF overflow entries
SHUTDN	Operator shutdowns
CHK300	P300 machine checks
PAR300	P300 memory parity checks
MOD300	P300 missing memory module checks
TYPE10	Entry for type 10
TYPE11	Entry for type 11
TYPE12	Entry for type 12
TYPE13	Entry for type 13
TYPE14	Entry for type 14
TYPE15	Entry for type 15
QUIET	Machine check mode now quiet
SETIME	Operator issued a SETIME command
REMARK	Operator remark
PACL	Priority ACL set
SENSOR	Sensor check

Note that the time/date stamps associated with the selected entries will not be processed unless TIMDAT is explicitly selected, for example, '-TYPE DISK TIMDAT' will process all disk errors and their associated time/date stamps. If TIMDAT alone is specified, all time/date stamps will be processed. If TIMDAT is specified in conjunction with one or more other types, only the time/dates of the selected types will be processed. If the -TYPE option is not specified, all entries will be processed.

- SPOOL Spool the output file when done. PRINT_SYSLOG will print the name of the output spool file.
- DELETE Delete the output file when done. This option should only be specified when the -SPOOL option is also specified.
- PURGE Empty, but do not delete, the event log input file when event log processing is complete. Write access is required on the input file.
- CENSUS PRINT_SYSLOG totals the entries for each event in the input file and writes the totals to the output file or terminal. Only non-zero totals are displayed.
- CONTINUE Continue after a bad entry is found. PRINT_SYSLOG will normally halt if an invalid entry is encountered. If this option is specified, PRINT_SYSLOG will continue processing in an attempt to find the next valid entry.
- DEBUG This option causes PRINT_SYSLOG to read entries from the terminal and can be used for testing PRINT_SYSLOG's formatting for entry types. Each entry should be entered as a series of tokens (using RDTK\$\$'s rules). Octal tokens are converted to binary; all others are taken as ASCII strings. PRINT_SYSLOG leaves this mode of operation whenever a 'Q', 'q', or null line is entered.
- REMARK text Enter an event of type REMARK directly into the input file. This can be used, for example, by an operator who wishes to record an observation on some event that might affect the subsequent operation of the system. All text after the -REMARK option is taken as the text to be entered into the input file. Consequently, the -REMARK option must be the last option specified on the command line. The message can be up to 80 characters in length and need not be surrounded by apostrophes. Write access is required on the input file.

-DUMP

In addition to its normal formatting, PRINT_SYSLOG will dump each entry processed in octal. This option is provided as an additional aid to those who define their own event types. Only those entries which have been selected for processing are dumped.

PRINT_SYSLOG Messages: The following messages may be displayed by PRINT_SYSLOG:

- UNKNOWN CPU MODEL xx.

A CPU model number was encountered with which PRINT_SYSLOG is not familiar. PRINT_SYSLOG generates a warning message and continues processing, treating the CPU model number as 0.

- BAD ENTRY ENCOUNTERED IN FILE 'logfile'.

The event log file logfile contains an entry which is not defined by PRINT_SYSLOG.

- DEFAULT INPUT FILE NAME NOT CONSTRUCTED.

A file of the type LOG.mm/dd/yy could not be found in LOGREC* or the top level directories did not exist or the user had insufficient access.

- INPUT LOGGING FILE filename NOT FOUND.
UFD <0>LOGREC* CONTAINS LOGGING FILE FOR THE LATEST COLD START.
ENTER INPUT FILE NAME (ENTER OR TO QUIT):

The user did not explicitly specify an input event file and PRINT_SYSLOG cannot find a file with a name of the format LOG.mm/dd/yy. PRINT_SYSLOG prompts the user for the input file name.

- OK TO DELETE FILE filename? ANSWER: 'Y' OR 'N'!

The output from PRINT_SYSLOG may be directed to a file or to the terminal. If file output is desired, and PRINT_SYSLOG finds that the output file already exists, this message is printed. The reply should be 'Y' to delete the file or 'N' to enter a new destination. If 'N' is entered, the message

NEW OUTPUT FILE NAME:

is printed. The user should enter a pathname; if not, PRINT_SYSLOG will continue to query the user with the preceding message sequence

asking if the existing output file is to be deleted or what will be the name of the new output file. Entering TTY will cause output to the terminal.

- OUTPUT HAS BEEN PLACED IN FILE 'filename'

PRINT_SYSLOG has completed the processing of the event entries and all other file manipulation requested by the user. This message is generated provided that output was directed to a file.

► PROP

PROP is a command that invokes the spool queue management utility for the system printer. PROP is described fully in Chapter 10. The command format is:

```
PROP { environment [option] }
      -STATUS
```

The options listed below define how PROP will be used:

<u>Option</u>	<u>Function</u>
<u>-ABORT</u>	Causes the phantom servicing <u>environment</u> to stop printing the current file; the file is left in the spool queue. This command allows the operator to force the printing of all short files, followed immediately by the restart of the aborted file. This option also forces a previously requested paper change, hang, or shutdown, using the FINISH keyword, to occur immediately.
<u>-BACK</u>	Causes the phantom servicing <u>environment</u> to restart printing of the current file 128-256 lines prior to the current line. Note that printing will not necessarily continue on a page boundary. This option is useful for very large files when a large portion of the file has been printed.
<u>-CONTINUE</u>	Takes the phantom servicing <u>environment</u> out of -HANG mode so it can continue printing or scanning. Printer operation continues where it left off.

<u>-CREATE</u>	Sets up a new <u>environment</u> . Subcommands (listed below) are used to define the environment. To delete an environment, use the -DELETE option.
<u>-DELETE</u>	Deletes <u>environment</u> . The environment must not be in service in order to use -DELETE.
<u>-DISPLAY</u>	Prints a detailed description of <u>environment</u> .
<u>-DROP</u>	Causes the phantom servicing <u>environment</u> to stop printing the current file and to delete this file from the spool queue. (The SPOOL option -CANCEL will delete a file from the spool queue if it has not started printing.)
<u>-HANG</u> $\left[\begin{array}{c} \text{NOW} \\ \text{FINISH} \\ \text{IDLE} \end{array} \right]$	Causes the phantom servicing <u>environment</u> to temporarily suspend printing and/or scanning the queue. NOW suspends the phantom immediately. FINISH suspends the phantom after it finishes the currently printing file. IDLE suspends the phantom when it has no more work to do. (The default is FINISH.)
<u>-LINEUP</u> [lines]	Used to check alignment of paper. Functions like -RESTART, except that the phantom servicing <u>environment</u> does an implied -HANG after printing <u>lines</u> number of lines. (The default for <u>lines</u> is 4.) If the alignment of the paper is correct, give the -CONTINUE. Otherwise, realign the paper and give another -LINEUP command.

Note

-LINEUP may be specified as part of the -START command. For example:

```
PROP PRL -START -LINEUP 5
```

In this case, the lineup will be done on the first file eligible for printing.

<u>-MODIFY</u> $\left[\begin{array}{c} \text{NOW} \\ \text{FINISH} \\ \text{IDLE} \end{array} \right]$	Modifies <u>environment</u> . If -MODIFY is used on a running environment, use the NOW, FINISH, or IDLE argument to determine when the changes will be made. NOW causes the modifications to be made immediately. FINISH causes the modifications to be made after the current file is finished printing. IDLE causes the
---	---

modifications to be made when the running environment has no more work to do. (The default is FINISH.) As with -CREATE, the subcommands shown below define the changes to be made.

<u>-RESTART</u>	Causes the phantom servicing <u>environment</u> to restart the currently printing file. This is often used after the printer has been halted because paper has run out, paper has jammed, or a similar cause. Printing of the file restarts from the beginning of the file.
<u>-START</u>	Starts up a phantom to service <u>environment</u> . Used to start the spooler initially or after -STOP.
<u>-STATUS</u>	Prints a list of the currently defined environments and indicates which ones are currently being serviced by a phantom. Cannot be specified with <u>environment</u> .
<u>-STOP</u> $\left[\begin{array}{c} \text{NOW} \\ \text{FINISH} \\ \text{IDLE} \end{array} \right]$	Causes the phantom servicing <u>environment</u> to log itself out. This is used to stop printing or scanning the queue. NOW shuts down the phantom immediately. FINISH shuts down the phantom after the current file is finished printing. IDLE shuts down the phantom when it has no more work to do. (The default is FINISH.) To start up another phantom to service <u>environment</u> , use the -START option.

The use of the -CREATE or -MODIFY option initiates PROP's environment definition mode. PROP prints an angle prompt (>) and waits for subcommands. The specified environment may now be defined through use of the subcommands shown below:

<u>Subcommand</u>	<u>Definition</u>
<u>COMOUT</u> $\left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}$	ON specifies that a COMOUTPUT file will record all phantom actions. The file, named SPOOLQ> <u>environment</u> , is created the first time a phantom is activated to service <u>environment</u> . On subsequent activations, material is appended to the file. OFF specifies that no COMOUTPUT file is made. (Default is OFF.)

DEST synonym Defines synonym as a logical destination recognized by this environment. Users specify logical destinations using the -AT option of the SPOOL command.

DISPLAY Displays all environment parameters with their latest values.

DEVICE { PR0
PR1
PR2
PR3
PLOT
AMLC n } Directs output to the selected device. If AMLC is selected, it must be followed by an octal line number. The asynchronous line configuration must have already been set on the supervisor terminal. (Default is PR0.)

EVFU { -ON
-OFF
-NAME file } Indicates whether or not the printer associated with environment utilizes an EVFU (Electronic Vertical Format Unit).

-ON indicates that the EVFU is used to define form length, and that no special channels have been defined. A typical use is on a 300 lpm printer/plotter that contains an EVFU (for example, a standard 300 lpm printer/plotter without a "forms length" switch).

-OFF is the default setting, and is used for a printer that does not contain an EVFU (for example, a standard 300 lpm printer/plotter with a "forms length" switch). With -OFF specified, the phantom servicing environment will treat the printer normally.

-NAME file is used when an EVFU file explicitly defining channels has been constructed. file names the specific EVFU file for a particular form. This file must reside in the SPOOLQ directory.

For further details, see Chapter 10.

FILE Exits the environment definition mode. If -CREATE was used to enter environment definition mode, the environment file E.environment is created and defaults are used wherever the user did not specify a value. If -MODIFY was used to enter environment definition mode, environment is updated. The modifications may not take effect until sometime later if environment is currently being serviced by a phantom.

FORM synonym

Defines synonym as a form name recognized by environment. Users specify form names using the -FORM option of the SPOOL command.

| HEADER $\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$

Sets the number of header pages. A setting of 2 will also cause a trailer page to be printed for each file. (Default is 1.)

LARGE n

Prints files shorter than n records before longer files. This may range from 0 to 32767. (Default is 30.)

LENGTH n

Prints n lines per page. This may range from 10 to 32767. (Default is 38.)

LIMIT n

Prevents printing of any files bigger than n disk records. This may range from 0 to 32767, or OFF. (Default is OFF, meaning no limit has been set.)

LINES n

Sets the number of physical lines per page. (Default is OFF, which equals 0.)

Note

If you are using a Prime Matrix Line Printer (Model 3173 or 3126) with EVFU enabled, n must be the exact number of lines per page to assure a correct EVFU (Electronic Vertical Format Unit) load. See the EVFU subcommand, in this list.

LOWER n

Looks for SPOOLQs starting at logical disk n, where n is a decimal number. This may range from 0 to 63. (Default is 0.)

MESSAGE text

Prints text on every header page. text is one line of up to 80 characters. If text is omitted, a null message will result.

PAPER [name]

Indicates that name is the form mounted; name cannot be more than six characters long. (Default is blank.)

Note

Use of this option deletes all currently existing FORM synonyms.

<u>PLOT</u> { ON } { OFF }	If ON, scans the queue for PLOT files. If OFF, ignores PLOT files in the queue. (Default is OFF.)
<u>PRINT</u> { ON } { OFF }	If ON, scans the queue for PRINT files. If OFF, ignores PRINT files in the queue. (Default is ON.)
<u>QUIT</u>	Exits environment definition mode. Any parameter changes are ignored; <u>environment</u> remains unchanged.
<u>TYPE</u> { 0 } { 1 }	Specify 0 to indicate a 300 lpm printer/plotter. Specify 1 to indicate a band printer.
<u>UNDEST</u> synonym	Removes the logical destination name <u>synonym</u> from the list of logical destination synonyms.
<u>UNFORM</u> synonym	Removes the form name <u>synonym</u> from the list of form name synonyms.
<u>UPCASE</u> { ON } { OFF }	If ON, converts all lowercase characters to uppercase before printing. (Default is OFF.)
<u>UPPER</u> n	Prevents search for SPOOLQs on logical disks with numbers higher than <u>n</u> . <u>n</u> is a decimal number which ranges from 0 to 63. (Default is 63.)
<u>WIDTH</u> n	Sets the number of physical columns on a page. Used for formatting header and trailer pages. The range of <u>n</u> may be from 10-140. (Default is 108.)

► { REMOVE_PRIORITY_ACCESS }
 { RPAC }

This command removes a priority ACL from a partition, undoing the effect of a previous SET_PRIORITY_ACCESS command. The format is:

{ REMOVE_PRIORITY_ACCESS } partition-name
 { RPAC }

partition-name is the name of the disk volume whose priority ACL is to be removed. This command may only be issued from the supervisor terminal or by the System Administrator.

► REPLY

The operator uses the REPLY command to reply to users' mag tape requests. See Chapter 4 for a discussion of REPLY. The command format is:

REPLY -usernum -TAPE { ABORT
 GO
 pdn
 RESEND }

The command must include the user number (-usernum) and the -TAPE option, or an error message will result. Only one of the following listed keywords can be specified at a time:

<u>Keyword</u>	<u>Description</u>
ABORT	The operator is unable to assign the requested drive (no drive available, tape not found, etc.).
GO	The operator approves the request. The message displayed at the user terminal indicates that the desired tape drive has been assigned. GO is used to answer all requests that did <u>not</u> specify the MTX -ALIAS MTldn option.
pdn	The pdn option is used in all cases where a user specifies the MTX -ALIAS MTldn option. The operator selects a suitable drive, performs any special requests, then uses this option to send the drive's physical device number to the user's terminal. The following message is then displayed: Device MTpdn Assigned.

RESEND The most recently sent assignment request is repeated at the supervisor terminal.

Four additional REPLY options are available for the operator's use:

<u>Option</u>	<u>Description</u>
-TAPE -RESEND	This repeats the most recent request on the operators terminal.
-ALL -RESEND	This repeats all outstanding (unanswered) requests on the operator's terminal.
-usernum -RESEND	This repeats all outstanding requests from the specified user number.
-REPEAT seconds	This allows the operator to set how frequently (in seconds) the RESEND message is repeated. The <u>seconds</u> parameter is a decimal number. The default repeat frequency is 180 seconds.

► { SET_PRIORITY_ACCESS }
SPAC

To specify a user's access to an entire disk volume, overriding actual ACL specifications, use the SET_PRIORITY_ACCESS command. This sets a priority ACL on the specified disk for the users specified. This command can only be issued from the supervisor terminal or by the System Administrator, and its form is:

{ SET_PRIORITY_ACCESS } partition-name access-control-list
SPAC

partition-name is the name of the disk volume to be affected, and access-control-list is the list of identifiers and access rights. If a user is specified in this list, whether it be via username, group name, or \$REST (all users), the access rights specified will be granted to that user whenever the user references the specified disk volume. If all users are to be affected by a priority ACL, the \$REST specifier must be present in access-control-list. There is no implied \$REST:NONE as with regular ACLs.

If a user is not specified in the ACL list, the access rights granted the user will be taken from the ACL and password information on the disk volume, as usual.

Because a \$REST specifier overrides access specified on the disk, only \$REST:NONE should be specified. If \$REST:LUR were specified, for example, all users would be able to read all files, even if they normally would be denied access by ACL information on the disk.

For security reasons, use of this command causes an event (event type PACL) to be logged in the system event logging file. In addition, a message is printed at the supervisor terminal. The message includes the date and time at which the priority ACL was added, the partition to which it was added, and the ID number and username of the user who added it. No event or message occurs when a priority ACL is removed.

To remove a priority ACL, use the REMOVE_PRIORITY_ACCESS command. To list a priority ACL, use the LIST_PRIORITY_ACCESS command.

► { SET_QUOTA }
SQ

The SET_QUOTA command is used to set the maximum storage quota on a directory or subdirectory. The command format is:

{ SET_QUOTA } pathname -MAX number
SQ

pathname is the pathname of the directory having its quota set. If a quota is to be placed on the current directory, the full pathname must be given. However, if the quota is to be placed on a subdirectory of the current directory, only the name of the subdirectory need be specified.

number is the maximum number of records the directory can use. If number is zero, the quota is removed, allowing the directory to use as much storage as it can, being only limited by the available disk space or a quota placed on a parent directory.

If an attempt is made to set a quota on a directory or subdirectory to which the user has insufficient access, an "Insufficient access rights" message will be generated. If this happens, the operator may use the SET_PRIORITY_ACCESS command at the supervisor terminal to acquire sufficient access on the disk volume containing pathname. The SET_QUOTA command can then be repeated.

If pathname has no current quota (quota = 0), and there are attached users or open files in the directory or its subtree, a "File in use" message will be generated. This can happen on system directories, such as CMDNC0, BATCHQ, SPOOLQ, and LOGREC*, since users and phantom processes frequently access them. To successfully set a quota on a directory that is in use, wait until all users and phantoms log out, or set it before they log in after system coldstart.

► SETIME

The SETIME command sets date and time. The command format is:

SETIME -mmddyy -hhmm

mmddyy are digits that represent the month, day, and last two digits of the year.

hhmm are digits that represent the time in hours and minutes.

The two arguments to SETIME must be separated by spaces, and must start with a hyphen as the first character. For example, to set the date and time as "November 2, 1982 4:30PM", type:

OK, SETIME -110282 -1630

Note

Following a system coldstart on systems without a battery clock, the SETIME command must be issued before the MAXUSR command can be used to allow users to log in.

► SETMOD

SETMOD sets the mode for magnetic tape assignments. It must be given by the operator from the supervisor terminal. See Chapter 4 for a discussion of SETMOD. The command format is:

SETMOD { -OPERATOR
-USER
-NOASSIGN }

The options and the assignment modes they establish are:

<u>Option</u>	<u>Mode Definition</u>
<u>-USER</u>	This mode permits user assignment of tape drives by physical device number (pdn) alone, or by pdn and the option -ALIAS ldn. (ldn is the logical device number.) All other options to ASSIGN (such as MTX and -RINGON) require operator intervention. Only commands requiring operator assistance are displayed on the supervisor terminal. This is the default mode.

-OPERATOR This mode requires operator intervention in all tape drive assignment operations. All user-issued ASSIGN commands are displayed with user numbers at the supervisor terminal. The operator answers each ASSIGN request with the REPLY command, discussed below.

-NOASSIGN This mode forbids the assignment of any tape drive unit from user terminals. Any attempt to assign a drive will result in the message:

No Magtape Assignment Permitted. (ASSIGN)
ER!

In environments that restrict user access to tape drives, this informs users that the operator is not available for request handling.

► SHARE

SHARE is an internal command used to install a command or library into a supervisor segment. The SHARE command can be issued only by the system operator and must be preceded by an OPR 1 command and followed by an OPR 0 command. The principal use of the SHARE command is to make shared procedures available to all users on the system. The command format is:

SHARE [pathname] segment-number [access-rights]

pathname An optional parameter naming a runfile in the current UFD. If pathname is specified, the named file is restored into segment-number. If omitted, the command is being used to change access-rights.

segment-number The number of the segment to be shared. Valid segment numbers are from '1 to '3777. However, only segments '1-'40 and '2000-'2377 should be specified. Specification of a segment number outside these ranges will cause unpredictable results.

Caution

The gate segment, segment 5, should not be specified in a SHARE command. If this segment is shared, direct-entrance calls from user space will cause ACCESS VIOLATION messages.

access-rights The access rights to be given segment-number.
Possible values are:

0	No access.
'200	Read access.
'600	Read and execute access. (Default).
'700	Read, write, and execute access.

Segments '2000 to '2377 are available to hold shared programs. Those shared programs can be executed by any user. For a list of shared segment assignments, see the System Administrator's Guide.

It is also possible to change the access rights of supervisor segments '1 to '40. By changing access rights of specific supervisor segments, it is possible to either monitor or patch the supervisor from a user terminal. This feature is primarily for PRIMOS development and debugging; for users, it is dangerous and its use is not recommended.

Caution

It is possible to overwrite the operating system and the shared utilities with this command. Do not share into segments 0 - '1777. Segments 0 to '1777 are reserved for PRIMOS. Other segments that may contain system utilities are described in the System Administrator's Guide.

► SHUTDOWN

The SHUTDOWN command is used to shut down the system, local disk partitions, or remote disk partitions. It must be issued from the supervisor terminal.

System Shutdown: To shut down the PRIMOS system in an orderly manner, the command format is:

SHUTDOWN ALL

This will perform a complete PRIMOS system shutdown. To verify that this is what the operator wanted, the command will issue the prompt:

REALLY?

The operator must answer YES for the shutdown to occur. All user files are closed, disk partitions are shut down, network communication with the system is halted, and input from user terminals is turned off. When PRIMOS completes this process, it halts the CPU.

Note

If the shutdown is a normal one, it is advisable to follow the procedure for system shutdown outlined in Volume I of this book.

Shutting Down Local Disk Partitions: Shutting down a local disk partition causes all files on the partition that are currently open to be closed, all users who are attached to directories residing on the partition to be detached, and all attempts to use the partition from either the local system or a remote system to be rejected. The specified disk partition will be unavailable until it is added back to the system in a subsequent ADDISK command. Until then, users whose initial attach points reside on the partition will be unable to log in. See Chapter 7 for a complete discussion of shutting down disk partitions.

To shut down one or more disk partitions, the command format is:

```
SHUTDN { pdev-1 [pdev-2 ... pdev-n] }
          { pdev -RENAME packname }
```

The arguments function in the following manner:

<u>Argument</u>	<u>Function</u>
pdev-1 ... pdev-n	Shuts down the listed disk partitions.
pdev -RENAME packname	Allows the operator to supply a new <u>packname</u> for a partition when shutting it down. Only one physical device number can be specified each time this option is used. For example:

OK, SH 461 -RENAME B3.BAK

Aside from renaming the partition, this command has the same effect as SHUTDN pdev without the -RENAME option.

Caution

Do not shut down the physical device associated with logical device 0, or PRIMOS will lose the command directory (from memory, not disk). To recover, use the ADDISK command and attach to CMDNC0.

Shutting Down Remote Disk Partitions: Shutting down a remote disk partition removes it from the list of known disk partitions on the local system (the system on which the command is issued). It does not affect access to the partition from any other system, including the system on which the partition resides. However, shutting down a remote disk will close all files on the partition that are currently in use by local users. Also, all local users who are attached to directories residing on the partition will be detached, and all further attempts to use the partition from the local system to be rejected. The specified disk partition will be unavailable to users logged into the local system until it is made available once again via a subsequent ADDISK command. Remote disk partitions connected to the system can be disconnected by the command:

SHUTDOWN packname-1 [packname-2...packname-9] -ON nodename

nodename Network name of the system on which the devices are physically mounted.

packname-n Remote disk partitions, made available to the local system, that are to be disconnected.

Note

If a device is shut down at its local system, it is no longer available for use on any other system in the network.

SHUTDOWN Messages: SHUTDOWN prints the messages listed below. Warning messages are indicated by (Warning). When a warning message is printed, SHUTDOWN will continue if more than one device has been specified.

- Conflicting PDEVs "pdev1" and "pdev2". (shutdn)

Two of the PDEVs given in the list specify overlapping partitions of the same disk.

- Disk "packname" is local. (shutdn)

(Warning) The partition named packname is a local device, and must be shut down by PDEV.

- Disk "packname" is not currently added. (shutdn)

(Warning) The specified remote partition is not in the local disk list.

- Disk "packname" not on system "system". (shutdn)

(Warning) The specified remote partition exists, but does not belong to the remote system specified in the -ON option.

- Disk "pdev" is not currently added. (shutdn)

(Warning) The specified physical device is not in the list of started devices.

- Duplicate partition name "packname". (shutdn)

Partition name packname was given more than once in this remote SHUTDN command.

- Duplicate PDEV "pdev". (shutdn)

PDEV pdev was specified more than once in this SHUTDN command.

19.3 |

- Must specify at least one PDEV. (shutdn)

The SHUTDN command requires at least one argument.

- Must supply at least one partition name. (shutdn)

The SHUTDN command requires at least one argument.

- Node "nodename" not configured in network. (shutdn)

19.3 | The node specified as nodename is not currently RFA-enabled.

- Only one disk may be RENAMEd at a time. (shutdn)

Only one PDEV may be specified at a time when the -RENAME option is used.

- "packname" is not a valid partition name. (shutdn)

The specified packname does not conform to partition name syntax. Either it contains an illegal character, or it is more than six characters long.

- "pdev" is not a valid PDEV. (shutdn)

The pdev specified is not a legal physical device number. Either it contains decimal numbers or nondigits, or it is simply an illegal PDEV.

- Remote partitions may not be RENAMEd. (shutdn)

The -RENAME option may be used only for local disks.

- "system" is not a valid system name. (shutdn)

The specified remote system name is illegal. Either it contains illegal characters, or it is more than six characters long.

- System console command only. (shutdn)

The SHUTDOWN command may be used only from the supervisor terminal.

- System name must be specified with -ON option. (shutdn)

The -ON option was given, but was not followed by a remote system name.

- Write-protected disks may not be RENAMEd. (shutdn)

The -RENAME option may not be used on a disk added with the PROTECT option.

► SPOOL

The SPOOL command allows you, as an operator (logged in as SYSTEM, or from the supervisor terminal), to monitor the spool queue and delete unwanted entries. The format is:

SPOOL [option]

<u>Option</u>	<u>Description</u>
<u>-LIST</u>	Displays a list of all files currently in the spool queue.
<u>-CANCEL</u> n	Removes requested number (n) from the spool queue.

For more information, refer to Chapter 10 and the PRIMOS Commands Reference Guide.

► START_NET

The START_NET command allows you to bring up PRIMENET on a system without interrupting local system activity. If your node has been configured as a gateway node, it will also activate the route-through server.

The START_NET command can be issued only from the supervisor terminal. It is normally issued at system start-up, though it can be issued at any time. START_NET must precede any remote ADDISK commands since ADDISK verifies that the remote node is configured before allowing you to add disks for that remote system. The command format is:

```
START_NET [config_pathname] -NODE node-name [options]
```

The arguments and options are:

19.3

config_pathname Pathname of the configuration file. If not specified, defaults to PRIMENET*>PRIMENET.CONFIG.

-NODE nodename Specifies the name of the local node. Required entry.

-HELP Describes syntax and options of START_NET.

-TRACING_NODE Enables the trace option when MONITOR_RING is used. Abbreviation: -TN.

Note

User SYSTEM requires read access to the configuration file.

START_NET Replaces the NET ON configuration directive at Rev. 19.3.

► STATUS

STATUS is an internal command for monitoring system usage. In its full form at the supervisor terminal, it prints network nodename, list of PRIMOS file units open, information on physical devices connected to the system (volumename, logical device number, physical device number, network node), status of network nodes, identification numbers of the paging and command devices, and description of logged-in users (login name, user number, line number, devices, and peripherals). See Chapter 3 for a complete description of the STATUS command. The command format is:

```
STATUS [argument]
```

The arguments and the descriptions of their display as seen from the supervisor terminal are:

<u>Argument</u>	<u>Description</u>
<u>ALL</u>	Prints network nodenames, main memory size, file units open, assigned magnetic tape devices, started disk partitions, semaphore information, status of network nodes, the paging and command devices, and logged-in users.
<u>COMM</u>	Prints information on communications controllers (excluding the Prime Node Controller) present in a system. For each controller, the information includes: the controller name, its type, its device address, the number of asynchronous lines on it, and the number of synchronous lines on it.
<u>DEVICES</u>	Prints physical device number, user-id, user number, and logical device number of all currently assigned magnetic tape devices.
<u>DISKS</u>	Prints partition name, logical and physical device numbers, and nodename of all currently started disk partitions.
<u>ME</u>	Prints information on all users. Identical to STATUS USERS when used from the supervisor terminal.
<u>NETWORK</u>	Prints information regarding the status of the full duplex, ring, and public data networks.
<u>PROJECTS</u>	Prints information regarding the project status and user number of all currently logged-in users.
<u>SEMAPHORE</u>	Prints all semaphores, their values, and their users.
<u>SYSTEM</u>	Prints the version of PRIMOS in operation and, if the command is given from the supervisor terminal, the amount of physical memory being used.
<u>UNITS</u>	Prints user-id, system name, and then prints file unit information for each currently open file unit.
<u>USERS</u>	Prints user number, line number, and all partitions and assigned devices in use by each terminal currently logged into the system.

► STOP_NET

The STOP_NET command allows you to shut down PRIMENET without interrupting local system activity. It can be issued only from the supervisor terminal. The following actions occur on the local node when the STOP_NET command is issued:

- Remote disks are shut down.
- Open virtual circuits are cleared.
- NPX slaves go to sleep.
- Route-through server logs out.
- NETMAN logs out.

19.3

Note

If FTS is running on the local node, you must stop the file manager and file servers using the commands FTOP -STOP_MNGR and FTOP -STOP_SRVR before issuing the STOP_NET command.

The format of the STOP_NET command is:

```
STOP_NET [-HELP]
```

-HELP describes the complete syntax and options of STOP_NET.

► UNASSIGN

The operator can use the UNASSIGN command to remove the assignment of a peripheral device assigned to any user. UNASSIGN is discussed in Chapter 4.

The operator's primary use of the UNASSIGN command is to unassign disk partitions and tape drives as part of system maintenance (such as during backup procedures, when formatting disks, and when repairing file system partitions). Although the UNASSIGN command is fully described in the PRIMOS Commands Reference Guide, the description of how to unassign disk partitions and tape drives is included here for convenience.

The format of the UNASSIGN command for disks is:

```
UNASSIGN DISK pdev
```

The format of the UNASSIGN command for tape drives is:

```
UNASSIGN { MTpdn
          -ALIAS MTlcn }
```

The -ALIAS option can only be specified if a logical device number was previously assigned to this particular drive.

► USAGE

This system metering tool allows operators and users to monitor several performance factors of PRIMOS. Both manual and automatic sampling modes are available. For a complete discussion of USAGE and its output, see Chapter 5. The command format is:

USAGE [options]

Options may be selected in any order from the list below:

<u>Option</u>	<u>Meaning</u>
-USER	Causes system and per-user metering information to be displayed at each sample time. This is the default mode of operation.
-DISK	Causes system and disk metering information to be displayed at each sample time.
-ALL	Causes system, per-user, and disk metering information to be displayed at each sample time.
-FREQ n	Selects automatic sampling every <u>n</u> seconds (<u>n</u> must be an integer in the range 1 to 32767). It is recommended that <u>n</u> be not less than 30. If -FREQ is not given, manual sampling is selected (see below).
-TIMES n	Specifies the total number of samples to be taken if automatic sampling is in effect. The command will terminate after <u>n</u> sets of data have been printed. <u>n</u> must be an integer in the range 1 to 32767. If -TIMES is not specified, sampling continues indefinitely.
-BRIEF	Specifies that a short form of output is to be produced. This form presents an overview of what processes and users are consuming system resources. The default long form produces additional information.

19.3

- | | |
|----------|--|
| -DEBUG | Displays a detailed USAGE version number. |
| -ON node | Displays information about a remote system. <u>node</u> is the name of the system you wish to monitor. |

► USERS

Prints the number of users currently logged into the system. There are no options or parameters for this command. The total number of system users is output by this command. However, this total does not include the supervisor terminal, nor does it include users logged in from the local system to a remote system (unless they are using NETLINK).

► USRASR

The USRASR command allows the supervisor terminal to act as a user terminal by associating it with a different process. After initiating USRASR, it is still possible to invoke some supervisor commands.

The command format is:

USRASR user-number

user-number is a user number. For example:

OK, USRASR 4

The USRASR command works correctly only if the associated communications line is not enabled on the asynchronous controller. The communications line can be disabled with the AMLC command:

OK, AMLC TTYNOP 2

OK, USRASR 4

This AMLC command disconnects the user normally attached to asynchronous line 2.

To return the terminal to operation as a normal supervisor terminal, type:

OK, USRASR 1

14

PRIMOS II Commands Reference

INTRODUCTION

PRIMOS II is Prime's single-user operating system. It is primarily used for bootstrapping PRIMOS and for running diagnostics and maintenance utilities. In special situations, it may be desirable to run the computer as a single-user system under PRIMOS II. Such situations, however, are somewhat rare. As a general rule all operations are conducted under PRIMOS.

Operation under PRIMOS II should be avoided whenever possible.

WARNING

PRIMOS II can not access Disk Controllers '22 and '23.

19.3

USING PRIMOS II

PRIMOS II and its supporting software allow the user a number of operations:

- Format disks and partitions using MAKE.
- Back up disks with COPY_DISK.
- Create files and programs using NSED (unshared editor).

- Perform magnetic tape operations with MAGSAV and MAGRST.
- Perform file manipulation with FUTIL (copying, deleting, etc.).
- Perform maintenance on pre-Rev. 19 format disks with FIXRAT.
- Perform operations with maximum security because of the inherent single-user configuration.

Operational aspects of PRIMOS II that differ from PRIMOS include:

- Only certain commands may be used. The commands that run under PRIMOS II are as follows:

ATTACH	LISTF	PHYRST
BINARY	LISTING	PM
CLOSE	LOGPRT	PRERR
CNAME	MAGRST	PRIMOS
COMINPUT	MAGSAV	RESTORE
COPY_DISK	MAKE	RESUME
CREATE	NETCFG	SAVE
DELETE	NSED	SHUTDN
FIXRAT	OPEN	STARTUP
FUTIL	PASSWD	STATUS
INPUT		

Thus, many commands available to the operator under PRIMOS are unavailable here. For information on the use of these commands, consult the PRIMOS Commands Reference Guide.

- Pathnames may not be used under PRIMOS II.
- It is not possible to interrupt a program with BREAK (CONTROL-P). Instead, use the VCP (or, on older machines, the control panel) to HALT. A restart is accomplished by starting at a specified starting location. See Volume I of this book for more information.
- PRIMOS II occupies the user's address space, that is, the same physical memory as do system programs. This places some restrictions on program loading to ensure that PRIMOS II is not overwritten. PRIMOS II will not restore a saved file if this operation would overwrite the operating system.
- PRIMOS II cannot run any V-mode programs, such as SEG, or any shared programs.
- PRIMOS II cannot start up more than four partitions.
- PRIMOS II must be shut down (using the SHUTDN command) before a removable disk pack is removed.
- PRIMOS II does not have any CPL capabilities; active functions and global variables are not available.

Caution

If files are added, extended or deleted under PRIMOS II, quota information is rendered invalid.

ACLs are not recognized under PRIMOS II, and quotas are not enforced.

TYPES OF COMMANDS USED UNDER PRIMOS II

Commands that are used when working with PRIMOS II fall into four classes:

- Commands that work the same under PRIMOS II and PRIMOS, and for which either operating system is appropriate.
- Commands that work under PRIMOS II as they do under PRIMOS, but that do not accept pathnames.
- Commands that are not available or not recommended under PRIMOS.
- Commands that behave differently under PRIMOS II than under PRIMOS.

Commands That Work Under PRIMOS and PRIMOS II

The following commands work the same under PRIMOS II and PRIMOS. They are not described in this chapter, but some of them are described in Chapter 13. The NETCFG command is described in the System Administrator's Guide. The LISTF, PASSWD, PM, and PRERR commands are documented in the PRIMOS Commands Reference Guide.

COPY_DISK	NETCFG
LISTF	PASSWD
MAGRST	PHYRST
MAGSAV	PM
MAKE	PRERR

Commands That Do Not Accept Pathnames Under PRIMOS II

The following commands work under PRIMOS II as they do under PRIMOS, but they do not accept pathnames. They are not described in this chapter, but they are all described in the PRIMOS Commands Reference Guide.

ATTACH	OPEN
BINARY	RESTORE
CNAME	RESUME
INPUT	SAVE
LISTING	

Commands Not Available or Not Recommended For Use Under PRIMOS

The following commands are not available or not recommended under PRIMOS:

- FIXRAT, described in Appendix B.
- FUTIL, described in Appendix C.
- LOGPRT, described in Appendix D.
- NSED, described in the Prime User's Guide and the New User's Guide to EDITOR and RUNOFF. This functions exactly as the ED command under PRIMOS.
- STARTUP, described in this chapter.
- PRIMOS, described in this chapter.

Commands That Behave Differently Under PRIMOS and PRIMOS II

The following commands behave differently under PRIMOS II than they do under PRIMOS. They are all described in this chapter.

CLOSE	DELETE
COMINPUT	SHUTDN
CREATE	STATUS

PRIMOS II COMMANDS

► CLOSE $\left\{ \begin{array}{l} \text{ALL} \\ [\text{filename}] [\text{unit } 1 \dots \text{unit } 8] \end{array} \right\}$

Closes all units when ALL specified, or closes the specified filename in the current directory and closes all specified unit numbers. unit numbers, specified in octal, range from 1 to 17 (15 decimal).

► COMINPUT $\left\{ \begin{array}{l} \text{CONTINUE} \\ \text{PAUSE} \\ \text{filename } [\text{ufd}] \\ \text{TTY} \end{array} \right\} [\text{unit}]$

Used to take input from command input files rather than from the terminal. If unit is not specified, the default is 6. unit is specified in octal.

COMINPUT CONTINUE: Specifies that subsequent commands are to be read in from the command file already opened on unit.

COMINPUT PAUSE: Specifies that subsequent commands are to be taken from the terminal, but that the command input file is not to be closed. unit is ignored.

COMINPUT filename [ufd]: Specifies that the file in the current directory (or in the UFD named ufd, if specified) is to be opened on unit, and that subsequent commands are to be read in from that file.

COMINPUT TTY: Specifies that subsequent commands are to be taken from the terminal, and that the command input file open on unit is to be closed.

► CREATE ufd

The specified ufd is created in the current directory as a password directory with null owner and non-owner passwords and with no quota.

► DELETE filename

Deletes the specified filename in the current directory. If filename is a directory or segment directory, it must not contain any files or directories. To delete directories that are not empty, use the TREDEL command of FUTIL.

► PRIMOS

If the automatic boot procedure was not used, PRIMOS may be started up from within PRIMOS II. Usually, this is done by issuing the command:

PRIMOS

This starts PRIMOS from its default directory, which is normally UFD PRIRUN. However, you can start up PRIMOS from another directory by specifying a pathname on the PRIMOS command line, as follows:

PRIMOS pathname

This will cause the version of PRIMOS in the directory specified by pathname to be run. Once this command is issued, pathname replaces PRIRUN as the default directory from which PRIMOS is run. The next issuance of the PRIMOS command without an argument will cause PRIMOS to be run from the directory pathname.

If it is necessary to run PRIMOS from a directory other than the default PRIMOS directory (PRIRUN, or pathname), but it is not desirable to change the default PRIMOS directory, a different procedure is used. First, attach to the directory from which PRIMOS is to be run. Since the ATTACH command in PRIMOS II does not take pathnames, the ATTACH subcommand of FUTIL may be used instead. Then, enter the command RESUME PRIMOS. For example:

```
OK: FUTIL
GO
[FUTIL rev 19.0]
> ATTACH OPSYS>REV19
> QUIT
```

OK: RESUME PRIMOS

► SHUTDN

Causes all started up disks to be shut down by PRIMOS II. After this command is issued, disks may be turned off and disk packs removed.

► STARTUP [pdev-0 ... pdev-3]

The STARTUP command defines a list of physical devices to be used by PRIMOS II. A device is considered started if it has been mentioned in a previous STARTUP command. Additional devices may be started if the new list in a subsequent STARTUP command does not conflict with the list in a previous STARTUP.

pdev-0...pdev-3 are items in a list of physical disk (device) numbers. The order of the list defines the logical number sequence of the devices (pdev-0 is logical 0, pdev-1 is logical 1, etc.) as in the following example:

```
STARTUP 460 10460 462
```

The above command defines that physical devices 460, 10460, and 462 are to be started up and associates the following logical device numbers with the physical device numbers specified: 460 is logical 0; 10460 is logical 1; and 462 is logical 2. Logical device numbers may also be specified as arguments to the STARTUP command. When used in this manner, they must be followed by a slash and the associated physical device number as shown in the examples below:

```
STARTUP 0/460 1/10460 2/462
```

```
STARTUP 3/1464
```

If a nonexistent or not ready device is specified, the message

```
DISK pdev NOT READY
```

will be printed.

Starting up a disk that was not originally formatted by MAKE will result in the error message:

```
DISK IS NON DOS
```

(See Chapter 6 for information on MAKE.)

Note

Only four logical devices can be connected by STARTUP under PRIMOS II.

► STATUS

Displays the current directory, the low address of PRIMOS II (DOSLO), a list of open file unit numbers (in decimal), and a list of started-up disks (logical disk number, physical disk number, and volume name).

APPENDIXES

A

Physical Device Numbers

INTRODUCTION

Each disk or disk partition has a physical device number that identifies the type of storage device, the drive unit on which it is mounted, the size of the partition, and the location of the partition on the disk. These physical device numbers are used in the following commands:

ADDISK	DISKS	PHYRST
ASSIGN DISK	FIX_DISK	PHYSAV
CONFIG	FIXRAT	SHUTDN
COPY_DISK	MAKE	UNASSIGN DISKS

This appendix describes how to determine a physical device number for a given disk.

DISKS

Types of Disks

PRIMOS fully supports three varieties of disks — storage module disks (SMDs), cartridge module devices (CMDs), and fixed-media disks (FMDs). These disk varieties can be used for file storage and paging area by PRIMOS. Each variety is available in two or more sizes. In addition, PRIMOS supports file access for floppy disks (diskettes). However, floppy disks cannot be used as paging storage by PRIMOS.

Sizing a Disk Partition

19.3 |

Disk partition size is measured in heads. The number of heads on a disk (and the amount of usable storage per head) depends on the type of disk involved. Table A-1 summarizes these values for all disks fully supported at Rev. 19.3.

Table A-1
Disk Size Data

19.3 |

Disk Type	#Heads	#Records per Head	Total # Records	Usable MB	Removable
80MB SMD	5	7407	37035	75.85	yes
300MB SMD	19	7407	140733	288.22	yes
32MB CMD	2	7407	14814	30.34	1 head
64MB CMD	4	7407	29628	60.68	1 head
96MB CMD	6	7407	44442	91.02	1 head
68MB FMD	3	10071	30213	61.88	no
158MB FMD	7	10071	70497	144.38	no
160MB FMD	10	7389	73890	151.33	no
300MB FMD (1)	19	7407	140733	288.22	no
675MB FMD (2)	40	7569	302760	620.05	no

|

Notes to Table A-1

19.3 |

(1) Model 4475

(2) Sometimes referred to as a 600MB disk, or a 630MB disk. Some 675MB disks are labeled "600MB". The official name is now 675MB disk.

For all of the disks in the above table, a record holds 2048 bytes of user data. (An additional 32 bytes per record are used by PRIMOS for housekeeping information.) The amount of usable data in megabytes (Usable MB) is based on the 2048 bytes per record figure. This is why it is always somewhat less than the rated size of the disk — approximately 5 to 8% of a disk's capacity is used for formatting and housekeeping information.

The "Removable" column in the table above specifies whether the disk drive supports removable disk packs. Storage modules (SMDs) allow the entire disk pack to be removed. Fixed-media disks (FMDs) allow no such removal. Cartridge module devices (CMDs) allow the removal of one platter, which has one surface (head) of usable information. The remaining surfaces (1, 3, or 5 of them) are not removable.

The size of a disk partition is specified as an integral number of heads. A disk partition may be from 1 to 31 heads in size. Since a partition cannot be larger than 31 heads, the 40 heads available on a 675MB disk cannot be contained by one partition; at least two partitions must be defined to utilize the disk fully.

The head offset is always an even number, ranging from 0 to 30.

Drive Unit Numbers

Individual disk drive units are identified by drive unit numbers. A drive unit number is selected on the disk drive unit itself. The operator supplies this number to PRIMOS (or PRIMOS II) during disk-related activities as part of the physical device number. PRIMOS uses this drive unit number to identify the particular drive unit being referenced. No two disk drive units connected to the same disk controller may have the same drive unit number.

It is important to keep a record in the system logbook of drive unit numbers and of the physical device numbers (including partitioning) for disks mounted on these drives.

Storage Modules (SMD) and Cartridge Modules (CMD): Drive unit numbers for storage modules and cartridge modules are indicated and set by the removable buttons on the front of the drive. These buttons have the drive unit number printed on them, ranging from 0 to 3.

Fixed-Media Disks (FMD): The drive unit number for the Model 4475 FMD is set by a removable button on the front of the drive. Drive unit numbers for other fixed-media disks cannot be set by the operator. They are set internally by field technicians. A drive unit number for a fixed-media disk ranges from 0 to 3.

19.3

Floppy Disks (Diskettes): Drive unit numbers for floppy disks are set by a thumbwheel switch on the drive unit. For floppy disks, a drive unit number ranges from 1 to 8.

PHYSICAL DEVICE NUMBERS

Physical device numbers are specified in octal. The method used to construct a physical device number is based on the type of disk.

Storage Module (SMD) and Fixed-Media (FMD) Disks

Table A-2 summarizes all of the legitimate physical device numbers for storage module and fixed media disks. It does not list numbers that, although legal, are sure to result in an unused surface.

Table A-2

Physical Device Numbers for Storage Modules
and Fixed Media Devices

		Starting Head Number							
		0	2	4	6	8	10	12	14
N u m b e r o f H e a d s i n P a r t i t i o n	1	-----	01006z	02006z	03006z	-----	-----	-----	-----
	2	00046y	01046y	02046y	03046y	04046y	05046y	06046y	07046y
	3	00046z	01046z	02046z	-----	-----	-----	-----	-----
	4	00106y	01106y	02106y	03106y	04106y	05106y	06106y	07106y
	5	00106z	01106z	-----	-----	-----	-----	-----	07106z
	6	00146y	01146y	02146y	03146y	04146y	05146y	06146y	07146y
	7	00146z	-----	-----	-----	-----	-----	06146z	-----
	8	00206y	01206y	02206y	03206y	04206y	05206y	06206y	07206y
	9	-----	-----	-----	-----	-----	05206z	-----	-----
	10	00246y	01246y	02246y	03246y	04246y	05246y	06246y	07246y
	11	-----	-----	-----	-----	04246z	-----	-----	-----
	12	00306y	01306y	02306y	03306y	04306y	05306y	06306y	07306y
	13	-----	-----	-----	03306z	-----	-----	-----	-----
	14	00346y	01346y	02346y	03346y	04346y	05346y	06346y	07346y
	15	-----	-----	02346z	-----	-----	-----	-----	-----
	16	00406y	01406y	02406y	03406y	04406y	05406y	06406y	07406y
	17	-----	01406z	-----	-----	-----	-----	-----	-----
	18	00446y	01446y	02446y	03446y	04446y	05446y	06446y	-----
	19	00446z	-----	-----	-----	-----	-----	-----	-----
	20	00506y	01506y	02506y	03506y	04506y	05506y	-----	-----
	21	-----	-----	-----	-----	-----	-----	-----	-----
	22	00546y	01546y	02546y	03546y	04546y	-----	-----	-----
	23	-----	-----	-----	-----	-----	-----	-----	-----
	24	00606y	01606y	02606y	03606y	-----	-----	-----	-----
	25	-----	-----	-----	-----	-----	-----	-----	-----
	26	00646y	01646y	02646y	-----	-----	-----	-----	07646y
	27	-----	-----	-----	-----	-----	-----	-----	-----
	28	00706y	01706y	-----	-----	-----	-----	06706y	-----
	29	-----	-----	-----	-----	-----	-----	-----	-----
	30	00746y	-----	-----	-----	-----	05746y	-----	-----

Table A-2 shows all the valid physical device numbers for the 68MB, 80MB, 158MB, 160MB, 300MB, and 675MB disks. Numbers marked with z should only be used as the last partition on 80MB or 300MB storage modules, on Model 4475 fixed-media disks, or on 68MB or 158MB fixed-media disks. Use of these numbers on other disks reduces the storage capacity of the disk by over 15M bytes per unused head. (Note also that no partitions can start beyond head 30.)

y is twice the unit number of the drive unit on which the disk is mounted. z is twice the drive unit number plus one.

Table A-2 (continued)

Physical Device Numbers for Storage Modules
and Fixed Media Devices

Starting Head Number									
16	18	20	22	24	26	28	30		
-----	11006z	-----	-----	-----	-----	-----	-----	1	
10046y	11046y	12046y	13046y	14046y	15046y	16046y	-----	2	N
10046z	-----	-----	-----	-----	-----	-----	-----	3	u
10106y	11106y	12106y	13106y	14106y	15106y	-----	-----	4	m
-----	-----	-----	-----	-----	-----	-----	-----	5	b
10146y	11146y	12146y	13146y	14146y	-----	-----	-----	6	e
-----	-----	-----	-----	-----	-----	-----	-----	7	r
10206y	11206y	12206y	13206y	-----	-----	-----	-----	8	
-----	-----	-----	-----	-----	-----	-----	-----	9	o
10246y	11246y	12246y	-----	-----	-----	-----	17246y	10	f
-----	-----	-----	-----	-----	-----	-----	-----	11	
10306y	11306y	-----	-----	-----	-----	16306y	-----	12	H
-----	-----	-----	-----	-----	-----	-----	-----	13	e
10346y	-----	-----	-----	-----	15346y	-----	-----	14	a
-----	-----	-----	-----	-----	-----	-----	-----	15	d
-----	-----	-----	-----	14406y	-----	-----	-----	16	s
-----	-----	-----	-----	-----	-----	-----	-----	17	
-----	-----	-----	13446y	-----	-----	-----	-----	18	i
-----	-----	-----	-----	-----	-----	-----	-----	19	n
-----	-----	12506y	-----	-----	-----	-----	-----	20	
-----	-----	-----	-----	-----	-----	-----	-----	21	P
-----	11546y	-----	-----	-----	-----	-----	-----	22	a
-----	-----	-----	-----	-----	-----	-----	-----	23	r
10606y	-----	-----	-----	-----	-----	-----	-----	24	t
-----	-----	-----	-----	-----	-----	-----	-----	25	i
-----	-----	-----	-----	-----	-----	-----	-----	26	t
-----	-----	-----	-----	-----	-----	-----	-----	27	i
-----	-----	-----	-----	-----	-----	-----	-----	28	o
-----	-----	-----	-----	-----	-----	-----	-----	29	n
-----	-----	-----	-----	-----	-----	-----	-----	30	

To use Table A-2:

1. Decide on the number of heads in the partition.
2. Decide on the head number of the first head in the partition.
3. Look up the physical device number in the table.
4. Add '100 to the number for controller address '22;
'200 for address '27; '300 for address '23.

19.3

Notes

If the partition defined is not in Table A-2, then it is not a legal partition.

All partitions must begin on an even head number.

You should define a disk partition with an odd number of heads only if it is the last partition on a pack with an odd number of heads (80MB and 300MB storage modules, Model 4475 fixed-media disks, or 68MB and 158MB fixed-media disks). Defining such partitions in other cases wastes disk space.

19.3 |

Example: A system contains three drive units; drives 0 and 1 have 300 MByte storage modules, and drive 2 has an 80 MByte storage module. (See Figure A-1.) The modules are to be partitioned as follows:

Drive 0	Partitions of 2, 2, 6, 2, 2, 2, and 3 heads
Drive 1	Partitions of 14 and 5 heads
Drive 2	Partitions of 2 and 3 heads

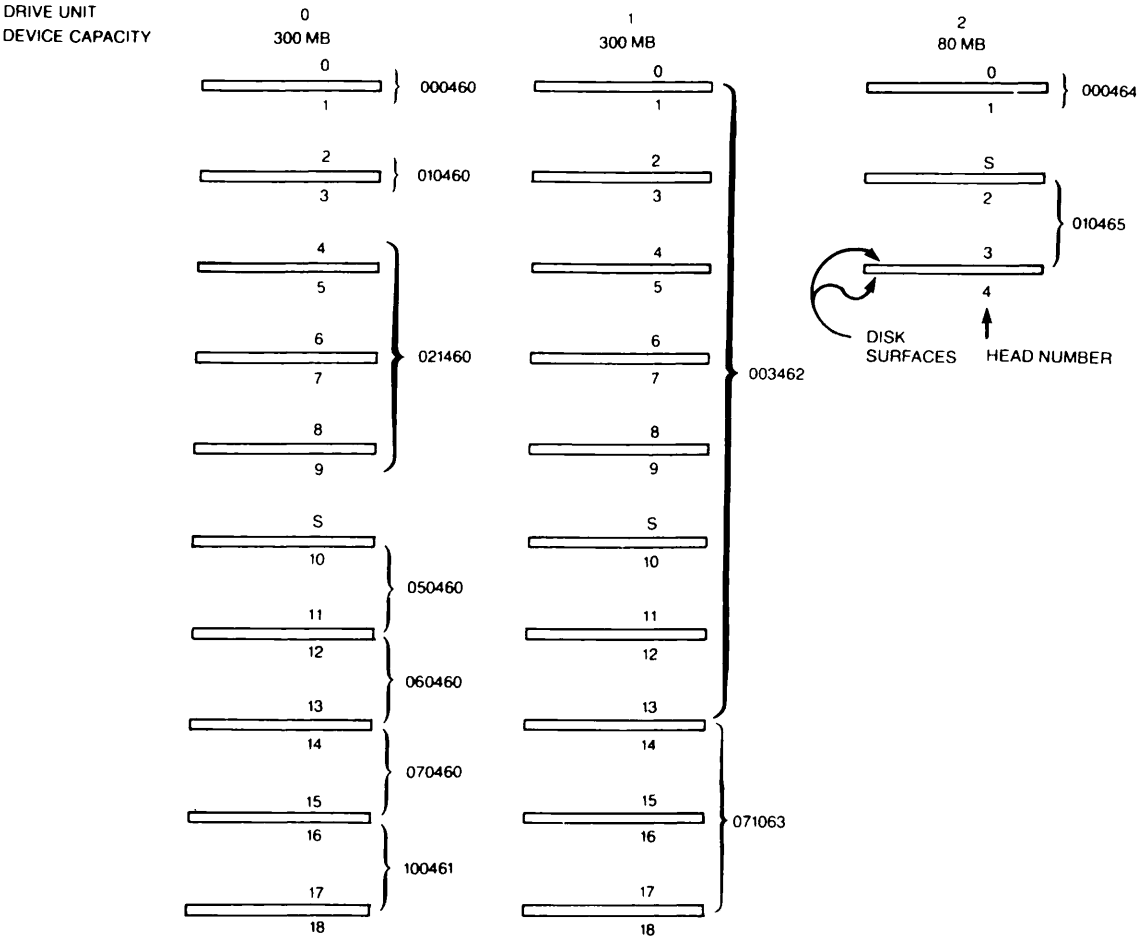
The physical device numbers are:

<u>Drive 0</u>	<u>Drive 1</u>	<u>Drive 2</u>
000460	003462	000464
010460	071063	010465
021460		
050460		
060460		
070460		
100461		

This example is illustrated in Figure A-1.

In all cases the drives are connected to the default controller address of '26. Each partition is treated by PRIMOS as if it were a separate physical device.

Binary breakdown of physical device number: For readers who are experienced in octal-to-binary and binary-to-octal conversion, Figure A-2 shows the exact breakdown of a physical device number in bits.



Example of Storage Module Partitions
Figure A-1

Cartridge Module Disks (CMDs)

Physical device numbers for CMDs are similar to those for storage module and fixed-media disks. However, there is a peculiarity in the assignment of head offsets; the removable portion of the disk is head offset 0, but the nonremovable portion starts at head offset 16.

This renders the specification of heads 1 through 15 ineffective, because there are no actual heads corresponding to the specification. Therefore, a CMD must contain at least two partitions. The removable portion of the disk is always one partition, and the nonremovable portion is one or more partitions.

Legal physical device numbers for CMDs are therefore summarized under the section on partitioning, later in this Appendix.

Floppy Disks (Diskettes)

The first four digits of a type 2 (floppy disk) physical device number are always 0. The fifth digit is always 2. The sixth digit is the drive unit number (which ranges from 1 to 8) minus 1. Therefore, the sixth digit ranges from 0 to 7.

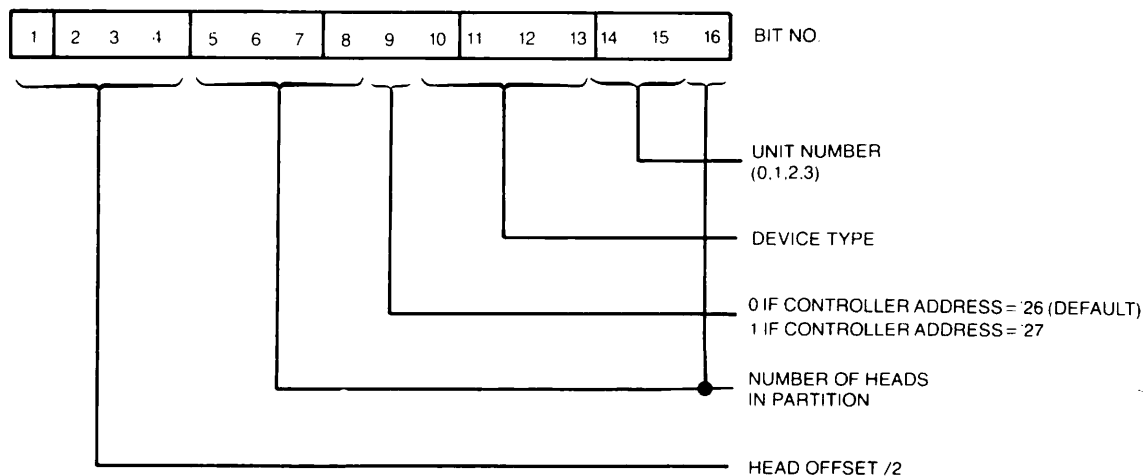
Note

A floppy disk cannot be specified as a primary or alternate paging device.

PARTITIONING OF SPECIFIC DISKS

This section describes each disk type and size that may be partitioned, and how it may be partitioned. All sizes of storage modules, cartridge modules, and Winchester disks may be partitioned. Partitioning a disk allows you to use it as several logical disks, each with its own name, rather than one physical disk.

There are various methods of partitioning the different types and sizes of disks. These methods are summarized below.



Construction of Physical Disk or Partition Number
Figure A-2

80MB Storage Module Disk

This disk has 5 surfaces. Table A-2 summarizes the legitimate physical device numbers for 80MB disks. Only those entries for which adding the number of heads to the starting head number produces 5 or fewer are legal. Because the disk has an odd number of heads, the last partition must have an odd number of heads, or else some disk space is not utilized. Therefore, one of the legal entries for 80MB disks ending in z in Table A-2 must be used to fully utilize the disk. An 80MB disk can be set up as one disk volume.

300MB Storage Module Disk

This disk has 19 surfaces. Table A-2 summarizes the legitimate physical device numbers for 300MB disks. Only those entries for which adding the number of heads to the starting head number produces 19 or fewer are legal. Because the disk has an odd number of heads, the last partition must have an odd number of heads, or some disk space will not be utilized. Therefore, one of the legal entries for 300MB disks ending in z in Table A-2 must be used to fully utilize the disk. However, use only the z entries for which adding the number of heads and the starting head number results in 19; those resulting in 5 should not be used, as they are only for 80MB disks. A 300MB disk can be set up as one disk volume.

Cartridge Module Devices (CMDs)

Cartridge module devices (CMDs) exist in three sizes: 32 MBytes, 64 MBytes, and 96 MBytes. They may be partitioned as indicated below.

The partitioning is done in a manner similar to that of the storage module (SMD) and fixed media (FMD) devices. However, the removable portion of a CMD is always a separate partition. The nonremovable portion is partitioned separately from the removable portion. The nonremovable heads are treated as if they started at a head offset of 16.

<u>CMD Type</u>	<u>Platter(s)</u>	<u>First Controller</u>	<u>Second Controller</u>
32 MB	Removable	6z (16 MB)	26z (16 MB)
	Nonremovable	10006z (16 MB)	10026z (16 MB)
64 MB	Removable	6z (16 MB)	26z (16 MB)
	Nonremovable	10046y (32 MB)	10066y (32 MB)
		11006z (16 MB)	11026z (16 MB)
		or	or
		10046z (48 MB)	10066z (48 MB)
96 MB	Removable	6z (16 MB)	26z (16 MB)
	Nonremovable	10046y (32 MB)	10066y (32 MB)
		11046y (32 MB)	11066y (32 MB)
		12006z (16 MB)	12026z (16 MB)
		or	or
		10106y (64 MB)	10126y (64 MB)
		12006z (16 MB)	12026z (16 MB)
		or	or
		10106z (80 MB)	10126z (80 MB)
		or	or
		10046y (32 MB)	10066y (32 MB)
		11046z (48 MB)	11066z (48 MB)

Notes

y is twice the drive unit number (0-3) on which the disk is mounted. z is twice the drive unit number plus one.

The removable surface of all CMD disks is always organized as one partition.

The nonremovable surface of the 32MB CMD is also organized as one partition.

The nonremovable surfaces of the 64MB CMD can be organized as one or two partitions.

The nonremovable surfaces of the 96MB CMD can be organized as one, two, or three partitions.

160MB Fixed Media Device

This disk has 10 surfaces. Table A-2 summarizes the legitimate physical device numbers for 160MB disks. Only those entries for which adding the number of heads to the starting head number produces 10 or less are legal. Because the disk has an even number of heads, no partition should have an odd number of heads, or else some disk space is not utilized. Therefore, none of the entries ending in z in Table A-2 can be used, if the disk is to be fully utilized. A 160MB disk can be set up as one disk volume.

300MB Fixed Media Device (Model 4475)

19.3

This disk has 19 surfaces. Table A-2 summarizes the legitimate physical device numbers for Model 4475 disks. Only those entries for which adding the number of surfaces to the starting surface number produces 19 or less are legal. Because the disk has an odd number of surfaces, the last partition must have an odd number of surfaces, or some disk space will not be utilized. Therefore, one of the legal entries for Model 4475 disks ending in z in Table A-2 must be used to fully utilize the disk. However, use only the z entries for which adding the number of surfaces and the starting surface number results in 19; those resulting in 5 should not be used, as they are only for 80MB disks. A Model 4475 disk can be set up as one disk volume.

675MB Fixed Media Device

This disk has 40 surfaces. Table A-2 summarizes the legitimate physical device numbers for 675MB disks. Because the disk has an even number of heads, no partition should have an odd number of heads, or else some disk space is not utilized. Therefore, none of the entries ending in z in Table A-2 can be used, if the disk is to be fully utilized. A 675MB disk cannot be set up as one disk volume, since a partition cannot be more than 31 heads in size. As a result, at least two partitions must be set up to fully utilize a 675MB disk.

B

FIXRAT

INTRODUCTION

FIXRAT is a maintenance program that checks the PRIMOS file integrity on any pre-Rev. 19 disk pack. FIXRAT may be run from a command file. It fully supports nested UFDs and nested segment directories. (This appendix assumes that the reader is familiar with the file structure described in the Subroutines Reference Guide. This guide should also be consulted for a description of segment directories and nested directories.)

WARNING

FIXRAT must not be used on any Rev. 19 partition. Use of this command on a Rev. 19 partition may result in the loss of data. When working on a Rev. 19 partition, use the FIX_DISK command only.

FIXRAT reads every record in every file, UFD, and segment directory, and checks that the information in each record header is consistent with record headers in the rest of the file and with the file directory that points to the record. Any inconsistencies generate an error message.

FIXRAT also builds a record availability table (RAT) from the existing file structure and compares it with the DSKRAT file. If discrepancies are found, FIXRAT prints an error message.

Note

The packname is the name of the file containing the disk record availability table. This file will be called the DSKRAT file, or just DSKRAT, in this discussion. Users can assign other names to DSKRAT either when the disk is created, by using the MAKE command, or at other times, by using the CNAME command.

If requested, FIXRAT will not only check the file structure but also repair pointers (if possible), truncate or delete defective files, and generate a corrected DSKRAT file. Up to two repetitions of FIXRAT may be necessary to repair a damaged file structure. The recommended procedure is to repeat FIXRAT until an error-free printout is obtained.

One suggested procedure for maintaining a disk pack is to run FIXRAT every morning and, if no errors occur, to then copy the pack onto a daily backup pack. If any files are truncated or deleted from the pack, they are copied from the existing daily backup disk to the disk pack. The owners of the bad files must be notified that those files have been copied from the backup and that any later modifications made to those files may have been lost.

FILE STRUCTURE

The file structure on any disk pack is a tree structure where the MFD can be compared to the trunk of the tree, the links between directories and files or subdirectories to branches, and the directories and files are like leaves.

A directory tree consists of all files and subdirectories that have their origin in that directory. FIXRAT traverses the file structure, generating terminal output.

FIXRAT prints BEGIN directory-name when beginning processing of a directory tree. On leaving a directory tree, FIXRAT prints END directory-name followed by the number of physical records (in decimal) used by all files and directories in the directory tree. FIXRAT indents the printed output one space for each level out in the tree in which the directory is located. This indented format makes it easy to understand the relationship of each directory to the other directories in the tree. To prevent excessive output, FIXRAT as a default prints out only directory names at levels 1 and 2 in the tree, as shown in the following example:

```
DISK PACK ID IS DSKRAT

BEGIN MFD
  BEGIN UFD1
    END UFD1      21
  BEGIN UFD2
    END UFD2      11
```

```
END   MFD           35
RECORDS USED(DECIMAL)=      35
RECORDS LEFT=              6223
DSKRAT OK
OK:
```

RUNNING FIXRAT

When invoked, FIXRAT asks a series of questions; each answer must be followed by a carriage return (CR). The command format is:

FIXRAT [OPTIONS]

If the optional argument OPTIONS is included in the command line, FIXRAT requests answers to the following questions after the device to be checked is specified:

1. Level to which directory names are to be printed? (Default is 2.)
2. Are directories that are nested too deeply to be automatically truncated? (Default is NO.)
3. Are file names to be typed? (Default is NO.)
4. Are file chains to be typed? (Default is NO.)

If the OPTIONS argument is omitted, FIXRAT uses the default answers, printing only the name and number of records used (in decimal) in the MFD and in each directory in the MFD, and not automatically truncating deeply nested directories.

FIXRAT first asks:

FIX DISK?

Caution

It is recommended that the answer be NO the first time FIXRAT is run. If any problems occur, it is advisable to make sure that these problems are not caused by hardware or the operating system. (This procedure also indicates what records have been lost.) Then, rerun FIXRAT and answer YES to this question.

If the answer is YES, FIXRAT compresses UFDs, truncates or deletes defective files, and generates a corrected DSKRAT file, in addition to checking the file structure and repairing all file structure errors.

WARNING

If a disk error has occurred, and if the user has valuable information on the disk that is not backed up, then do not run FIXRAT and answer YES. Consult a senior programmer or a Prime field analyst.

If the answer is NO, FIXRAT will ask:

UFD COMPRESSION ?

YES causes FIXRAT to compress UFDs, eliminating entries for deleted files or directories. A NO answer means that FIXRAT will not perform any disk modifications. This allows FIXRAT to be run on write-locked disks. FIXRAT tests the integrity of the file structure and prints error messages, whether or not it modifies the disk.

After the FIXRAT operations have been selected, the next question is:

PHYSICAL DISK =

Enter the number of the physical device (or partition) on which FIXRAT is to be run; FIXRAT then prints the disk packname (which is the name of the DSKRAT) and begins processing the file structures. (The DSKRAT is always the first file in the MFD.)

For partitioned disks, disk numbers include information on head offset and number of heads. If the user gives an incorrect disk number, one of the following messages is printed at the terminal:

- DEVICE, DSKRAT DIFFER IN HEAD COUNT. ABORT?
- DISK READ ERROR with status of 177777 ...FIXRAT aborts
- WRONG RECORD SIZE IN RAT HEADER ...FIXRAT aborts
- RAT HEADER WRONG LENGTH ...FIXRAT aborts

Note

If you get one of the above messages and you confirm that you gave a correct disk number, then the RAT or its header has been damaged.

If the FIXRAT command was given with OPTIONS specified, FIXRAT will ask print option questions. (See the next section.) Next FIXRAT prints the number of records used and the number of records left on the pack for file system use. Finally, FIXRAT compares a record availability table (built from the existing file structure) against the DSKRAT. If they match, FIXRAT types DSKRAT OK and exits to PRIMOS. If they do not match, FIXRAT types DSKRAT FILE DIRECTORIES MISMATCH.

Running FIXRAT with OPTIONS

If the command was invoked as FIXRAT OPTIONS, three print option questions will be asked.

The first question is:

TYPE DIRECTORIES TO LEVEL =

Enter an octal number corresponding to the lowest level in the tree structure in which directory names are to be printed. The following table describes the output:

<u>Level</u>	<u>Output</u>
blank	All directories
1	MFD only (level 1 directory)
2	MFD and all directories in MFD file (level 2 directories)
3	All output for level 2 and all directories at level 3
.	.
.	.
.	.

Note

FIXRAT will trace the nesting of directories to a depth of 700 levels (default value).

The next question asked is:

MAX NESTED DIRECTORIES LEVEL?

Enter an octal number that specifies the maximum level of directories that may be nested in a directory tree. (Default maximum is 700.)

FIXRAT then asks:

AUTO TRUNCATE DIRECTORIES NESTED TOO DEEPLY?

The default answer is NO. If the answer is YES, FIXRAT truncates directories that are nested too deeply within a directory tree without asking for confirmation from the user. If the answer is NO, FIXRAT prompts the user and provides the choice of either truncating the UFD that is nested too deeply or leaving it as it is and continuing with the FIXRAT operation.

FIXRAT will then ask:

TYPE FILE NAMES?

If the answer is YES, FIXRAT prints all filenames in all directories, indented appropriately. This option is useful for listing the contents of a disk. Unless the user requests suppression of directory name output by answering 1 to the question TYPE DIRECTORIES TO LEVEL =, directories will be printed three times: twice as directories and once as files. (If only a listing is desired, use the LISTF or LISTSAVE command of FUTIL, as described in Appendix C.)

FIXRAT will then ask:

TYPE FILE CHAINS?

If the answer is YES, FIXRAT prints the disk address of all records in all files on the disk. All files consist of one or more records chained together by pointers. This option is useful to see how files are distributed on a disk.

FIXRAT Output Example

The following is an example of FIXRAT output generated after all questions have been answered:

```
DISK PACK ID IS DSKRAT

BEGIN MFD
  BEGIN CMDNCO
    END CMDNCO      21
  BEGIN DOS
    END DOS         11
  END MFD           35
RECORDS USED(DECIMAL)=      35
RECORDS LEFT=              6223
DSKRAT OK
OK:
```

The first line of output indicates the disk packname, the name of the DSKRAT file.

In the next section of output, FIXRAT examines the file structure on the disk for consistency. The example above was generated from a disk that contains only two directories, CMDNC0 and DOS, in the MFD. If either of these directories contains subfile directories, FIXRAT traces the nested directory structure but does not print the names of the subfile directories. Each directory is printed twice: following the word BEGIN when FIXRAT enters the directory, and following the word END when FIXRAT is finished processing the directory and any subfile directories nested within it. Directories in the MFD are indented one space when typed, to show the nested structure.

Following the directory name, FIXRAT prints the number of records used in the directory plus all files nested within that directory. (Since all files on a pack are nested within the MFD, the number of records used in the MFD always matches the number of records used on the disk pack.) After the file structure analysis, FIXRAT prints the number of records used on the pack and the number of records left on the pack for file system use.

Finally, FIXRAT compares a record availability table (RAT) built from the existing file structure against the DSKRAT. In the preceding example, they match and FIXRAT prints:

DSKRAT OK

Then FIXRAT exits to PRIMOS or PRIMOS II.

If the RAT and DSKRAT totals do not match, FIXRAT prints:

DSKRAT, FILE DIRECTORIES MISMATCH

If the user typed YES to the question FIX DISK ?, FIXRAT repairs the DSKRAT, then prints the message:

DSKRAT FIXED

FIXRAT exits to PRIMOS. If the FIX DISK ? prompt was answered with NO, the question is asked again. If the user answers YES, the DSKRAT is repaired and the message "DSKRAT FIXED" is printed. With a NO response, the DSKRAT is not repaired and FIXRAT exits to PRIMOS. (This option is useful if there are no file structure errors but there is a bad DSKRAT.)

Broken File Structure Messages

When FIXRAT detects a problem in the file structure, it prints an error message. (All numeric values are octal except the index level, which is decimal.) The format of the error message is:

```
reason for error
FILE= filename TYPE= filetype
BRA= bra      FATHER= fra      INDEX LEVEL= index-level
```

BAD RECORD= cra TRACK= track HEAD= head
 DIRECTORY PATH= pathname (or MFD if file is MFD)
 FILE DELETED, FILE TRUNCATED or blank

Explanations of various elements of the error message are as follows:

filename	The name given to the file.
filetype	The type of file: SAMFIL, DAMFIL, SAMSEG, DAMSEG, SAMUFD, or ILLEGAL.
bra	The beginning record address of the file.
fra	The record address of the parent directory.
index-level	The current index level (0 except for DAM files).
cra	The current record address.
track	The cylinder number at which the error occurred. The outside rim of the disk is track 0. Track numbers increase inwards up to 822.
head	The surface of the disk. The bottom of the disk pack is head 0. (Storage modules have up to 19 heads.)
directory-path	The list of nested file directories needed to get from the MFD to the bad file. Because all treenames have the MFD as a root, "MFD" is not printed as part of the path.

After printing the directory path, FIXRAT prints the disposition of the bad files. If the FIX DISK question was answered NO, FIXRAT does nothing to the files, and therefore prints nothing.

If FIXRAT is requested to FIX DISK and detects a bad file, it either truncates or deletes the file, depending on where in the file a problem is detected. It then prints the message FILE TRUNCATED or the message FILE DELETED. If FIXRAT deletes a file, the action taken depends on the type of its parent directory. If the parent directory is a UFD, FIXRAT removes the entry from the directory in an action similar to that of the DELETE command. If the parent directory is a segment directory, FIXRAT identifies the entry as a null entry.

More detail is given in the section FIXRAT Error Messages.

Segment Directories

A segment directory may contain references to files and other segment directories. The distinction between a UFD and a segment directory is that entries in a UFD are referenced by name and those in a segment directory by position in the directory. Each entry in a UFD consists of a disk address that is the beginning record address of the file, followed by a name. (Refer to the Subroutines Reference Guide.) In a segment directory, FIXRAT prints the absolute position of the file in the segment directory as a decimal number. As with user file directories, identically named files in different segment directories represent unique files.

Directories Nested Too Deeply

FIXRAT truncates directories that are nested too deeply in a directory tree (i.e., greater than 700 levels out). When this condition is detected, FIXRAT prints the message:

```
DIRECTORY ufcname NESTED TOO DEEPLY
```

and then asks:

```
TRUNCATE DIRECTORY?
```

If the answer is NO, FIXRAT aborts. Otherwise, FIXRAT truncates the UFD named ufcname, by making it an empty UFD (i.e., its entry as a UFD in the parent directory is preserved, but ufcname will have no files or directories subordinate to itself). After truncation of ufcname, FIXRAT continues.

Disks With Badspots

FIXRAT handles disks with badspots. (Refer to MAKE, Chapter 6.) In checking the integrity of the DSKRAT, FIXRAT also examines the file BADSPT in the MFD. The file BADSPT (if it is present) contains the record addresses of badspots on the disk. Since disk records that have badspots are not available for file system use, it is useful for FIXRAT to know the location of badspots when fixing a disk or checking file integrity.

The file BADSPT may not be present on the disk, or it may be empty. In either of these cases, FIXRAT performs no badspot handling.

If the format of BADSPT is incorrect, FIXRAT prints the message:

```
BAD BADSPT FILE, IGNORED
```

Otherwise, FIXRAT types:

BADSPT FILE HAS ENTRIES = number-of-badspots

When FIXRAT is finished processing the disk, if there are any badspots that affect the current disk or partition, FIXRAT prints:

BADSPT RECORDS LOST = number-of-records-lost

If a badspot is found on the disk in records 0 to 15 (BOOT, MFD, etc.), FIXRAT prints the following message and aborts:

CANNOT PROCESS BADSPOT FOR RECORDS .LT. 16

Bad BOOT

If the BOOT file in the MFD is accidentally deleted or broken, PRIMOS will allocate record number 0 to the next new file. FIXRAT will send the following message if any file except the BOOT in the MFD contains record 0:

BAD DISK ADDRESS BAD RECORD = 0

If this occurs, restore (REST) the BOOT from a good MFD on another disk and save (SAVE) it into the MFD before doing anything else.

FIXRAT ERROR MESSAGES

This section lists all error messages generated by FIXRAT and gives an expanded explanation of them. The user should be familiar with the details of the file structure. Error messages are of the form:

```
reason for error
FILE= filename TYPE= filetype
BRA= bra      FATHER= fra      INDEX LEVEL= index-level
BAD RECORD=  cra      TRACK= track HEAD= head
DIRECTORY PATH= pathname (or MFD)
```

Following is a list of FIXRAT error messages with explanations:

- FILE = MFD BAD RECORD = n
 DIRECTORY PATH = MFD
 FIXRAT ABORTED

An MFD has been altered and damaged. The best action to take is to copy the backup disk onto the original disk and continue.

- DSKRAT NOT IN MFD
FIXRAT ABORTED

The DSKRAT has been accidentally deleted from the MFD. Suggested action is the same as for a damaged MFD.

- RECORD READ OK NOW CHECKS BAD
POSSIBLE DRIVER ERROR, FIXRAT ABORTED

Suggested action is to run the disk diagnostic test on a scratch disk pack.

- DIRECTORY RECORD READ OK NOW CHECKS BAD
POSSIBLE DRIVE ERROR, FIXRAT ABORTED

Suggested action is to run the disk diagnostic test on a scratch disk pack.

- Check For MFD

FIXRAT checks that the first three entries in the MFD are DSKRAT, MFD, and BOOT. (The DSKRAT may have any name. The name is used on the disk pack ID.)

- DSKRAT NOT IN MFD, REPLACE IT?
- MFD NOT IN MFD, REPLACE IT?
- BOOT NOT IN MFD, REPLACE IT?
- MFD HAS BAD NAME, REPLACE?

If there is a YES (followed by CR) response to any of these questions, the specified action is performed. These messages can occur when one of the indicated files is modified. Users must neither delete nor alter the DSKRAT, MFD, or BOOT, since these are system files used by PRIMOS.

- 2 FILES POINT TO SAME RECORD

Two files point to the same first record; FIXRAT prints the name of the second file only. This error may occur if the DSKRAT is changed by a user overwriting PRIMOS II, or if the BADSPT file is changed after FIXRAT has been run, in which case records have been erroneously made available to new files.

- BACK POINTER MISMATCH SHOULD BE good-pointer IS bad-pointer

The back pointer of a record does not point to the previous record of the file. If the current record is the first record of a file, the back pointer is not 0.

- BAD DAM POINTER

A DAM data file or DAM segment directory has a bad index in the first record of the file, and the nth index of the file does not point to the nth record of the file for all records of the file. This error is repaired by FIXRAT.

- BAD DISK ADDRESS

A pointer to a disk record is out of range. Acceptable range is between 1 and NRECS-1, where NRECS is the number of records available for file system use. NRECS is stored in the DSKRAT data header. A record address of 0 is acceptable only for the disk bootstrap loader file BOOT in the MFD.

- BAD FILE TYPE file-type-number

The file type, in the first record of the file, is not between 0 and 4.

- BAD FORWARD POINTER forward-record-address

The forward pointer address is not in the current physical disk or disk partition.

- BAD INDEX LEVEL SHOULD BE good-pointer IS bad-pointer.

The index has an incorrect level indicator.

- BAD RECORD ID

The first word of a record contains a number unequal to its record address. This message may be preceded by 10 disk error messages because this problem could indicate a disk drive problem.

FIXRAT has difficulty determining whether the error is a disk drive error or a broken file. The disk driver retries 10 times, producing 10 disk error messages, and then returns to FIXRAT, which prints the message BAD RECORD ID. Be sure to allow FIXRAT 10 disk error messages before assuming that there is disk drive trouble. (Refer to UNRECOVERED DISK READ ERROR below.)

- BAD UFD HEADER

The directory header contains bad data other than that covered by other error messages.

- BAD WORD COUNT word-count

The data word count of a record is not reasonable. A word count of 0 indicates an empty record.

- BRA POINTER MISMATCH SHOULD BE good-pointer
IS bad-pointer.

The beginning-record address word of the second record (or greater) of a file does not point to the beginning record of the file.

- CANNOT DELETE BOOT, RAT, OR MFD

An error, which would normally cause deletion of a file, has been found in the BOOT, RAT, or MFD file. FIXRAT aborts.

- DAM INDEX TOO LONG

The index for a Direct Access Method (DAM) file is too long to represent the file. FIXRAT truncates the index.

- DAM INDEX TOO SHORT

The index for a Direct Access Method (DAM) file is too short to represent the file. FIXRAT truncates the file.

- DAM POINTER MISMATCH SHOULD BE good-pointer
IS bad-pointer

The record pointers in the index for a Direct Access Method (DAM) file do not match the record pointers in the file.

- DIRECTORIES NESTED TOO DEEPLY

Directories may be nested to a depth of 700 levels. FIXRAT cannot follow the directory tree because the user has nested directories to more than 700 levels. FIXRAT aborts.

- DISK ERROR, FIXRAT ABORTED

An error occurred in reading the MFD or DSKRAT file.

Note

If FIXRAT aborts, an operator can only try again or seek expert advice. One possible action is to check the physical integrity of the disk.

- DSKRAT BAD

This message is obtained if the DSKRAT file contains either bad record pointers or inconsistent information. If the DSKRAT is BAD, FIXRAT reconstructs it, using parameters typed by the user in response to the following questions:

```

INPUT DECIMAL RECORD SIZE =
INPUT DECIMAL FILE SPACE RECORD COUNT =
INPUT DECIMAL CYLINDERS =
INPUT DECIMAL HEADS =

```

If the user types CARRIAGE RETURN to any of the questions, FIXRAT uses default values for the particular disk model. FIXRAT prints the default or the specified values back to the user for verification and then asks "OK?". If the answer is YES, FIXRAT repairs the DSKRAT and continues; otherwise it requests the parameters again. (Refer to Appendix A for tables of disks and values.)

- FATHER POINTER MISMATCH SHOULD BE good-pointer IS bad-pointer

The father-record word of the first record of a file does not point to the beginning record address of its file directory.

- FILE TYPE MISMATCH

The file type in the file header does not match the file type in the UFD entry for this file.

- INCONSISTENT ENTRY IN UFD: RECORD=record, WORD=word.
CHANGED TO VACANT

Information in a file entry in a UFD is not self-consistent. The entry is flagged as being deleted. If UFDs are compressed, this entry will be eliminated.

- NOT ENOUGH MEMORY

There is insufficient space to read the DSKRAT file into memory.

- RAT MISSING

The disk record availability table is not in the MFD. FIXRAT aborts.

- UNRECOVERED DISK READ ERROR

On an unrecovered disk read error, the track and head of the bad record are reported. Normally, a succession of many unrecovered read errors indicates a malfunctioning or misaligned disk drive, a head crash, or a bad disk pack. An unrecovered read error may also indicate a pack that has badspots that were not entered into the BADSPT file during the last MAKE operation. When an unrecovered read error occurs, the disk must be remade by MAKE so that the BADSPT file may be updated.

- 2 FILES POINT TO SAME RECORD

These errors occur when FIXRAT is rerun as a result of the unrecovered read error recovery procedure, since a badspot may affect up to eight records belonging to multiple files.

C

FUTIL

INTRODUCTION

FUTIL is used to copy and delete files and directories. FUTIL may be invoked from either PRIMOS or PRIMOS II. For operations under PRIMOS, it is recommended that the COPY and DELETE commands be used rather than FUTIL.

To invoke FUTIL, input the command name FUTIL. When invoked, FUTIL prints the prompt character, >, and awaits a subcommand string.

```
OK: FUTIL
[FUTIL rev 19.0]
>
```

Restrictions

Use Under PRIMOS: When using FUTIL under PRIMOS, certain operations may interfere with the work of other users. For example, a UFDCPY command may fail if it is used to copy all files from a UFD currently used by another logged-in user. If any file in that directory is open for writing by that user, UFDCPY will encounter the error "file in use" and will skip the file. If the user attempts to open a file for writing while UFDCPY is running, the user may encounter that error. The FUTIL commands LISTF and TRECPY cause the same interaction problems. FUTIL commands such as COPY and DELETE can also interfere with other users, but with these commands only one file is potentially involved in a conflict. To minimize conflicts, it is recommended that the COPY command be used rather than FUTIL. If FUTIL is used under PRIMOS, it is recommended that the PRIMOS LD command be used rather than FUTIL's LISTF subcommand.

Working in the MFD: A UFDCPY of the MFD to the MFD of another disk has the effect of merging the contents of two disks onto one disk. The operator should be sure there is enough room on the TO disk before attempting this operation or the merge will not be successfully completed. The names of segment directories on the two disks must not conflict. Files of the same name will be overwritten and UFDs of the same name will be merged. To avoid conflict, it may be desirable to copy (UFDCPY) the MFD of one disk into a user file directory on another disk. Each directory originally on the FROM disk becomes a subdirectory in that UFD on the TO disk. A UFDCPY of an MFD does not copy the DSKRAT, MFD, BOOT, or BADSPT to the TO directory. If a user wishes to copy BOOT to the TO directory, the COPY_DISK command should be used. The DSKRAT and BADSPT files should never be copied from one MFD to another.

The effect of a UFDCPY from the MFD of a disk in use to the MFD of a newly formatted disk is to reorganize the disk files so that all files are compressed. That is, all files have their records close to each other on the new disk. After such compression, the access time to existing files on the new disk is less than the access time on the old disk. Furthermore, new files tend to be compact since all free disk records are also compressed. The use of such compressed disks should improve the performance of all PRIMOS systems.

WARNING

FUTIL operations affecting the MFD should be done carefully. Never give the command TREDEL MFD, since the command will delete every file on the disk except the MFD, disk record availability table, BOOT, and BADSPT. When the system is operating under PRIMOS, a LISTF or UFDCPY of the MFD should be done only if no files or directories on the disk are being used.

FUTIL SUBCOMMANDS

FUTIL subcommands are briefly described below. A complete description of FUTIL is available in the PRIMOS Commands Reference Guide.

Caution

Do not abort copying or deleting operations under PRIMOS II; allow them to run to completion. Aborting FUTIL while copying or deleting files may cause a pointer mismatch, bad file structure, or a directory with a partial entry. PRIMOS II will not run correctly with a directory containing a partial entry. FIX_DISK or FIXRAT should be run immediately if these conditions are encountered.

Many FUTIL commands are significantly affected by the current value of the FROM and TO directories. For an explanation of FROM and TO directories, refer to the description of the FROM and TO subcommands below.

► ATTACH directory-pathname

Moves the current UFD to the directory defined by pathname. The pathname may contain, at most, 10 directories. The first directory in the pathname may be * (current UFD). All directories in the pathname must be UFDs or sub-UFDs.

► CLEAN prefix [level]

Deletes all files in the FROM directory whose filenames begin with the characters specified as prefix. If level is specified greater than 1, that many levels of sub-UFDs (including the FROM UFD) are scanned for prefix matches. In no case does CLEAN delete a UFD, sub-UFD, or a segment directory.

► COPY filea [fileb] [,filec [filed]]...

Copies filea in the FROM directory into fileb in the TO directory and, optionally, copies filec in the FROM directory to filed in the TO directory, etc. Filename pairs must be separated by commas. If the second filename of a pair is omitted, the new file is given the same name as the old file. The files filea, filec, etc., must be SAM or DAM files and cannot be directories. If fileb exists prior to the copy, it must be a SAM or DAM file and the user must have read, write, and delete/truncate access rights to the target file (fileb in this case). If fileb exists, it is deleted; then filea is copied to fileb. The file type of fileb will be the same as filea.

► COPYDAM filea [fileb] [,filec [filed]]...

Functions in the same way as COPY, but COPYDAM sets file type of fileb and filed to DAM, instead of copying the type of filea and filec.

► COPYSAM filea [fileb] [,filec [filed]]...

Functions in the same way as COPY, but COPYSAM sets file type of fileb and filed to SAM, instead of copying the type of filea and filec.

► CREATE ufdname [owner-password [nonowner-password]]

Creates a UFD in the TO directory and assigns any owner and nonowner passwords specified. A UFD of the same name cannot already exist in the TO directory. If a password is not specified, it is set to six spaces (null). If a password longer than six characters is specified, only the first six characters are used. The access rights of the new UFD are the default access rights set by PRIMOS.

► DELETE filea [fileb]...

Deletes specified files from the FROM directory. filea and fileb cannot be directories.

► FORCE { ON
OFF }

FORCE ON causes read-access rights to be forced on any files or subdirectories within the FROM directory. The option remains in operation until the command FORCE OFF is specified. UFDCPY never forces rights on the primary level of the FROM or TO directory.

Note

Use of FORCE ON causes LISTF, LISTSAVE, SCAN, UFDCPY, and TRECPY to fail on write-protected disks.

► FROM pathname

Defines the FROM directory in which files are to be searched by FUTIL subcommands. pathname may contain up to 10 directories that can be segment directories as well as User File Directories (UFDs). If segment directories are specified, the user must have read-access rights to them. If any error is encountered, the FROM directory is set to the current UFD (*). The first directory in the pathname may be *, which refers to the current UFD. The default FROM directory is the current UFD.

Note

The FROM command never changes the current UFD. If the FROM name is a relative pathname (that is, begins with *), any subsequent ATTACH commands that change the current UFD will reset the FROM name to *.

► LISTF [level] [FIRST] [LSTFIL] [PROTEC] [SIZE]
 [RWLOCK] [TYPE] [DATE] [PASSWD]

Lists the FROM directory pathname, the TO directory pathname, and all files and directory trees in the FROM directory at the terminal. LISTF optionally follows each filename by its protection attributes, size in disk records (2048 bytes per record), file type, date/time modified, and (on directories) owner and nonowner passwords.

► LISTSAVE filename [level] [PROTEC] [SIZE] [TYPE]
 [DATE] [RWLOCK] [FIRST] [PASSWD]

Functions identically to the LISTF command with the LSTFIL option specified, except the output listing file is named with the name specified by filename rather than LSTFIL, and the LSTFIL option is redundant.

► PROTECT filename [owner-access] [nonowner-access]

Protects filename in the FROM directory with the owner and nonowner protection attributes specified.

► QUIT

Returns to PRIMOS II command level.

► SCAN filename [level] [PROTEC] [SIZE] [TYPE]
 [DATE] [PASSWD] [LSTFIL] [FIRST] [RWLOCK]

Returns to PRIMOS II command mode. Searches the FROM directory tree for the occurrence of all files, sub-UFDs, and segment directories that are named with the name specified by filename.

If the level specified by the argument level is 1 (the default), only the filename followed by the information specified by the optional arguments is printed. If the level specified by level is greater than 1, the pathname (treename) to the file or directory, starting from the FROM directory, is printed. In addition, the information specified by any optional arguments may be printed after the treename. See LISTF for a description.

► SRWLOC filename number

Sets the per-file read/write lock for the file specified by filename. The parameter number is a number that is the read/write lock setting. 0=system default, 1=n readers or 1 writer, 2=n readers and 1 writer, 3=n readers and n writers.

► TO pathname

Defines the TO directory in which files are searched. The TO directory is defined from the pathname parameter, which has a format similar to the directory pathname specified for the FROM command. The pathname may contain at most 10 directories that may be segment directories as well as UFDs. If segment directories are specified, the user must have read and write access to them. The first directory in the pathname may be the current UFD (*). The default TO directory is the current UFD. If any error is encountered, the TO directory is set to the current UFD (*).

Note

The TO command never changes the current UFD. If the TO name is a relative pathname (i.e., begins with *), any subsequent ATTACH commands that change the current UFD will reset the TO name to *.

► TRECPY dira [dirb] [,dirc [dird]]

Copies the directory tree specified by directory dira to directory dirb, and optionally copies dirc to dird. dirb and dird must not exist prior to the TRECPY command. If dirb is omitted, dira is taken as the name of the directory to be copied to. dira and dirc must be in the FROM directory; dirb and dird are created in the TO directory.

The directories dirb and dird are created with the same directory types and passwords as dira and dirc, and with default access rights. Also, the per-file read/write lock setting is copied by TRECPY. The names, access rights, and passwords of all files and subdirectories are also copied.

► TREDEL dira [dirb]

Deletes the directory specified by directory dira and optionally deletes dirb from the FROM directory. (dira and dirb must be directories.)

► TREPRO directory [owner-access [nonowner-access]]

The TREPRO command is the same as PROTECT, except that directory is a UFD or segment directory in the FROM directory and it and all files under it (UFDs only) are protected with the specified access rights. The default access rights are <1 0>.

► TRESRW directory number

Sets the per-file read/write locks for all files in the subtree beginning with the directory (segment directory or UFD) specified by directory. The parameter number indicates the read/write lock settings, which are discussed in the description of the FUTIL command SRWLOC.

► UFDCPY

Copies all files and directory trees from the FROM directory to the TO directory. Files already existing in the TO directory with names identical to those in the FROM directory are replaced.

Segment directories already existing in the TO directory with names identical to those in the FROM directory are not allowed and will not be copied. Files and directories created in the TO directory will have the same file types and access rights as the old files. When the copy is finished, the new file will have the same protection attributes as the corresponding file in the FROM directory. The names, access rights, per-file read/write lock settings, and passwords of all files and subdirectories within directory trees being copied are also copied. Other existing files and directories in the TO directory are not affected. UFDCPY is effectively a merge of two directories (including the merge of sub-UFDs). Both the FROM and the TO directories must be user-file directories.

► UFDDEL

Deletes all files and directory trees (specified by directories) within the FROM directory. The owner password must be given in the FROM command to provide read, write, and delete access to all files and directories within the FROM directory. These rights are not required for files and subdirectories nested within the directories in the FROM directory.

Note

Read- and write-access rights to a sub-UFD are sufficient to command the deletion of the contents of that directory, but not to delete the directory itself.

► UFDPRO [owner-access [nonowner-access [levels]]]

The UFDPRO command is used to protect all files and directories within the FROM directory with the specified rights, going down sub-UFD trees the specified number of levels. The default rights are <1 0> and the default level is 1.

► UFDSRW number [levels]

Sets the per-file read/write locks for levels of files in the FROM directory. The parameter number is the read/write lock setting, which is discussed in the description of the command SRWLOC. The default level is 1.

D

LOGPRT

INTRODUCTION

LOGPRT is a program that writes the contents of the input system event logging file or the input network event logging file to a disk file or to the operator's terminal. When running under PRIMOS, the commands PRINT_SYSLOG and PRINT_NETLOG are preferred. However, only LOGPRT will run under PRIMOS II.

INVOKING LOGPRT

The command line to invoke LOGPRT is as follows:

```
LOGPRT [outtreename] [options]
```

Note

If LOGPRT is not present in OMDNC0, LOGPRT can be invoked by the command line:

```
ATTACH TOOLS  
R LOGPRT [outtreename] [options]
```

outtreename

The pathname of the destination for LOGPRT's output. If TTY is specified, the output will be to the user's terminal. If outtreename is omitted, output will be to the file LOGLST in the home UFD. (If the -NET option is given, the file's default name is NETLST.)

<u>Option</u>	<u>Meaning</u>
<u>-HELP</u>	Prints a list of LOGPRT options. The LOGPRT command must be reissued after the list of options is printed.
<u>-CENSUS</u>	Prints number of each event type processed. Only selected types are counted and only nonzero counts are displayed. The number of date/time stamps is displayed but date/time stamp entries are not included in the end-of-file total message. The total number of overflows is also displayed.
<u>-CONTIN</u>	Continues LOGPRT after encountering an invalid entry. LOGPRT normally halts if an invalid entry is encountered in the input system event logging file or the input network event logging file. When this option is specified, LOGPRT continues processing and attempts to find the next valid entry.
<u>-DELETE</u>	Deletes the output file (after spooling). This option is ignored if TTY is specified for <u>outtreename</u> . <u>May only be used with the -SPOOL option.</u>
<u>-DUMP</u>	Dumps each entry in octal.
<u>-FROM</u> { <u>mmddyy</u> [hhmm] } <u>TODAY</u>	Prints entries from the specified date <u>mmddyy</u> or today's date to the latest entry.
<u>-INPUT</u> [pathname]	Specifies the pathname of the input system or network event logging file to be processed as <u>pathname</u> . Input logging files are located on logical device 0, in the UFDs LOGREC* and PRIMENET*. The filename format for event logging files is LOG.mm/dd/yy and NET_LOG.mm/dd/yy, where <u>mm/dd/yy</u> is the date on which a cold start of the machine was made, or the EVENT_LOG -ON command was issued. The default is the most recently created event logging file in the UFD LOGREC* on logical disk zero or the most recently created event logging file in the UFD PRIMENET* on logical disk zero, if the -NET option is used.
<u>-NET</u>	Processes the input network event logging file. <u>Must precede all other options.</u>

<u>Option</u>	<u>Meaning</u>
<u>-PURGE</u>	Empties the input system event logging file after LOGPRT has finished processing. The default is to leave the file unmodified. Requires write access rights to the event logging files.
<u>-REMARK</u>	Enters comment directly into the event logging file. An example would be an observation of some event which might affect the subsequent operation of the system. All other LOGPRT options, except -INPUT and -NET, are ignored if -REMARK is specified. -REMARK must be the last option specified on the command line and all text following -REMARK is taken as text to enter into the event logging file. The text may be up to 160 characters and need not be surrounded by apostrophes. Write access is required.
<u>-SPOOL</u>	Automatically spools the output file. LOGPRT prints the name of the spool file and indicates whether the spool file is long or short. This option is ignored if TTY is specified for <u>outtreename</u> .
<div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center;"><u>Caution</u></p> <p>Do not use this option when running LOGPRT under PRIMOS II. If you attempt to do this, the system will halt. If this happens, warmstart PRIMOS II with the VCP command:</p> <p style="text-align: center;">DOS</p> <p>If your system does not support the VCP DOS command, enter:</p> <p style="text-align: center;">SYSCLR RUN 170000</p> </div>	
<u>-TYPE</u> t1 t2...tn	Processes entries only of the indicated event type(s). The system and network event types are listed in the following two sections.

SYSTEM EVENT TYPES

System event types that may be specified with LOGPRT's -TYPE option are as follows:

<u>Type</u>	<u>Meaning</u>
COLD	Cold starts
WARM	Warm starts
TIMDAT	Time/date entries (see Note)
CHECKS	Machine checks (including memory parity)
MCHECK	Machine checks (excluding memory parity)
DISKER	Disk errors
OVERFL	Record event logger overflow entries
SHUTDN	Operator shutdowns
CHK300	Prime 300 machine checks
PAR300	Prime 300 memory parity checks
MOD300	Prime 300 missing memory module checks
TYPE10...TYPE15	Entries for user-defined types 10 to 15
DSKNAM	ADDISK entries
POWERF	Power fail checks
SETIME	SETIME command issued
QUIET	Quiet machine check mode
REMARK	Operator message
PACL	SET_PRIORITY_ACCESS commands
SENSOR	System shutdown due to sensor check
BADENT	Bad entries not of types listed above

NETWORK EVENT TYPES

Network event types that may be specified with LOGPRT -NET's -TYPE option are as follows:

<u>Type</u>	<u>Meaning</u>
COLD	Cold starts
WARM	Warm starts
TIMDAT	Time/date entries
RESET	Circuit resets
BADSEQ	Packets out of sequence
OVERFL	Event logger overflow entries
SHUTDN	Operator shutdowns
LPE	Local procedure errors
RING1	Tokens inserted into the ring
RING2	Ring dim out of receive blocks
RING3	Ring nodes not accepting transmits
NETDMP	NETDMP calls
SMLC1	SMLC status errors
SMLC2	SMLC no STX preceding ETX
SMLC3	No system blocks for SMLC protocol messages

SMLC4	SMLC resets
HOSTDN	Level III protocol down
POWERF	Power fail checks
INCREQ	Incoming call requests for FAM I debug
OUCREQ	Outgoing call requests for FAM I debug
REMARK	Operator remark
NPXTHR	NPX throttled on transmit or receive
NPXRCV	NPX got an unanticipated receive status
NPXCLR	Unexpected clearing cause on NPX master's circuit
NPXSEQ	NPX found sequence error in bounce detect
NPXCON	Unexpected circuit status, NPX call setup
NPXRLS	Bad virtual circuit clearing in NPX
RNGRCV	Spurious receive interrupt was encountered
RNGHRD	PNC hardware failure received
RNGRES	Resource failure on communications queue
SMLC5	Indicates a CMDR has been sent
SMLC6	Internal error has been encountered
RNGTMT	Ring node receive timeout
DIAPKT	Diagnostic packet received
ICS1.1	Deconfigure code word was not queued
ICS1.2	Logical connection deleted
ICS1.3	Logical connection could not be deleted
ICS1.4	LCAD1_ was not found in the LCB
ICS1.5	IPQNM IE\$FRM error was received
ICS1.6	Flush Timeout
ICS1.7	Illegal flush was completed
ICS1.8	Line is not assigned
ICS1.9	Unidentified ICS error occurred
ICS1.0	Line is not defined
BADENT	Bad entries not of types listed above

19.3

Note

The time/date stamps associated with the selected entries will not be processed unless TIMDAT is explicitly selected. For example: -TYPE DISKER TIMDAT will process all disk errors and their associated time/date stamps. If TIMDAT alone is specified, all time/date stamps will be processed. If TIMDAT is specified in conjunction with one or more other types, only the time/dates of the selected types will be processed. If the -TYPE option is not specified, all entries will be processed.

LOGPRT MESSAGES

If the output file already exists, LOGPRT prints:

OK TO DELETE OLD outtreename? ANSWER: 'Y' OR 'N'!

The reply should be "Y" to delete the file or "N" to enter a new outtreename. If "N" is entered, the user is asked:

NEW OUTPUT TREENAME:

Under PRIMOS II, LOGPRT then prints the prompt:

REPLY PU TO PURGE WHEN DONE:

Any reply but "PU" causes LOGPRT to leave the input event logging file unmodified.

PURGING INPUT EVENT LOGGING FILES

System and network input event logging files may occupy an excessive amount of disk space due to either of two conditions: if your system always stays up, a single long input file will be created; if your system is brought up and down frequently, a number of small input files will fill the directory. In either case, input event logging files should be spooled periodically using the -SPOOL option of the LOGPRT command. Once spooled, the files should be deleted from the directory if they are no longer needed. This allows you to maintain a record of the system's operation without allowing the files to take up large amounts of disk space.

INFORMATION PRINTED IN THE EVENT LOGGING FILE

The first line of the event logging file is a header line containing the pathname of the input file and the system time and date in the format:

```
***** pathname, hh:mm:ss day dd mmm yyyy *****
```

The header is followed by formatted entries, one or more lines per entry. (All numbers are octal except where noted.) Each entry in the file is preceded by a date/time record that indicates when the event logging buffer (LOGBUF) was written to the event logging file on the disk (LOGREC*). All events following this entry and before the next date/time record happened during the minute immediately before the time shown. In the next example, the file header and the first entry in the system event logger file are shown:

```
***** <0>LOGREC*, 19:23:44 TUE 02 NOV 1982 *****
13:21:20 TUE 02 NOV 1982
SHUTDOWN BY OPERATOR
```

EXAMPLE OF EVENT LOGGING SESSION

An entire event logging session might look like this:

OK, LOGPRT
 LOGPRT REV 19.0
 OK TO DELETE OLD LOGLST? ANSWER : 'Y' OR 'N' ! Y
 OK, SLIST LOGLST

***** LOGREC*>LOG.05/10/82, 13:44:24 THU 13 MAY 1982 *****
00:00:04 MON 10 MAY 1982
 COLD START PRIMOS REV 19.0 CPU TYPE = 5 MICROCODE REV = 15
 ID = 000000 000005 000000 000017 000000 000000 000000 000000
 DISK MOUNT: PEGSYS ON 004463
 DISK MOUNT: HYDRA ON 000460
 DISK MOUNT: HRPY ON 032060
 DISK MOUNT: SCYLLA ON 071061
01:43:40 MON 10 MAY 1982
 SHUTDOWN BY OPERATOR
01:48:04 MON 10 MAY 1982
 COLD START PRIMOS REV 19.0 CPU TYPE = 5 MICROCODE REV = 15
 ID = 000000 000005 000000 000017 000000 000000 000000 000000
 DISK MOUNT: PEGSYS ON 004463
 DISK MOUNT: HYDRA ON 000460
 DISK MOUNT: HRPY ON 032060
 DISK MOUNT: SCYLLA ON 071061
08:55:28 MON 10 MAY 1982
 WARM START
09:13:28 MON 10 MAY 1982
 DISK MOUNT: CHMRA ON 001064
13:35:28 MON 10 MAY 1982
 DISK MOUNT: BASLSK ON 001065
13:41:28 MON 10 MAY 1982
 DISK MOUNT: BASLSK ON 001065
20:01:28 MON 10 MAY 1982
 DISK MOUNT: MPYTHN ON 001064
20:06:28 MON 10 MAY 1982
 DISK MOUNT: MPYTHN ON 001064
20:07:28 MON 10 MAY 1982
 DISK MOUNT: MPYTHN ON 001064
07:47:24 TUE 11 MAY 1982
 WARM START
07:58:24 TUE 11 MAY 1982
 WARM START
08:02:16 TUE 11 MAY 1982
 SHUTDOWN BY OPERATOR

<u>TYPE</u>	<u>NUMBER</u>
COLD	2
WARM	3
TIMDAT	13
SHUTDN	2
DSKNAM	14

***** END OF FILE -- 21 ENTRIES, 21 PROCESSED *****
 OK,

LOGLIST MESSAGES

- BAD ENTRY: xxxxxx

An entry of unrecognized type or length longer than 81 words was encountered. The length of the entry is printed.

- COLD START PRIMOS REV rr CPU TYPE = m MICROCODE REV = nn
ID= xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx xxxxxx

Indicates a cold start of PRIMOS was performed.

A cold start entry contains eight words of information obtained from the STPM (Store Processor Model Number) instruction. (See the Assembly Language Programmer's Guide.) m indicates the CPU type. nn indicates the revision of the microcode running. xxxxxx is the full eight-word identification code from the STPM instruction. rr is the ASCII version identification string, if present.

- DISK aa ERROR DVNO= pdn (cccc-d CTRLR e, UNIT f) CRA= xxxxxx xxxxxx
CYL= xxx HEAD= xx RECORD= xx RCRA= xxxxxx xxxxxx
STATUS(ACCUM)= xxxxxx STATUS(LAST)= xxxxxx RETRIES= xx (y)
WORDNO= xxxxxx CORRECTION= xxxxxx xxxxxx (z)

A disk read or write error occurred during the indicated operation.

aa is either RD for read or WT for write. pdn indicates the physical device number. cccc indicates the controller model. d indicates the device type (MHD for moving head disk, SM for storage module). e indicates the controller number. f indicates the unit.

CRA xxxxxx xxxxxx shows the error's desired Current Record Address. The cylinder, head, and record addresses are given in decimal. RCRA is the found CRA read from the disk record header on a CRA error, and it is given for read operations only.

STATUS(ACCUM) is the Boolean OR of all status bits obtained during retries. STATUS (LAST) is the status of the last operation. RETRIES= xx is the number of retries attempted. If the operation was completed successfully the value of the entry y will be RECOVERED. y is UNCORRECTABLE if the error could not be corrected.

WORDNO= indicates, after a correction, the record's corrected word number. CORRECTION= indicates, after a correction, the 32-bit correction pattern. z is either CORRECTED or UNCORRECTED, as appropriate.

- DISK MOUNT: volume-name ON pdn

An ADDISK or STARTUP command was issued. The indicated volume-name was mounted on the disk identified as pdn (which is a physical disk).

- *** END OF FILE -- xxxxxx ENTRIES, ppppp PROCESSED ***

This message is printed when LOGPRT reaches the end of the input logging file. xxxxx (decimal) gives the number of entries processed not including time/date and LOGBUF overflow entries. ppppp is the number of entries processed.

When all the entries in the input logging file (or other input file) have been processed, LOGPRT normally closes the file and exits. If the -PURGE option has been specified (or "PU" was the reply to the PURGE prompt under PRIMOS II) and the user has write access to the file, the LOGPRT program positions itself to the beginning of the input file before closing, in effect emptying the file.

Finally, if the -SPOOL option has been specified, LOGPRT sends the output file to the spool program and prints the name of the resulting spool file. If the -DELETE option is in effect, the output file is deleted.

- *** FILE EMPTY ***

This message is printed if LOGPRT finds no entries in the system logging file.

- LOGBUF OVERFLOW -- xxxxxx ENTRIES LOST

Indicates xxxxx (decimal) event entries were lost due to overflow of the event logging buffer (LOGBUF).

- MACHINE CHECK MODE NOW QUIET

PRIMOS entered Quiet machine check mode.

- MACHINE CHECK (xxx) DSWSTAT= ssssss ssssss DSWRMA= yyyyy rrrrrr
rrrrrr DSWPB= pppppp pppppp [DSWPARITY xxxxxx xxxxxx ...]

A machine check occurred. Information regarding the DSW (descriptor segment word) at the time of the check is indicated by DSWSTAT, DSWRMA (DSW Real Memory Address at last machine check), DSWPB (DSW Procedure Base at last machine check), and DSWPARITY.

If the RMA INVALID bit is set (bit 9 of DSWSTAT), yyyyy is (INV); otherwise yyyyy is absent.

DSWPARITY is not present on all CPU models. At Rev. 19, DSWPARITY is displayed only for the Prime 750 and 850. If DSWPARITY is not present, xxx is an encoding of the machine check code and not RCM Parity in DSWSTAT as follows:

<u>xxx</u>	<u>Meaning</u>
BMA	Memory address
BMD	Memory data output
BPAI	Peripheral address input
BPAO	Peripheral address output
BPD	Peripheral data output
RCD	Cache data
RCM	RCM parity error (XCS only)
RDXI	RDX-BPD input
RF	Register file

Prime 750 and 850: If DSWPARITY is present, it is broken down by reporting board (A, C, CS, J) and signal name as follows. (All signals are reported in the positive sense. For example, if ROMPE is printed, it means that the signal ROMPE- was 0.)

DSWPARITYH

01 - RPARERR1+	CS	DMX input	E6: BPD or Burst- R0,R2 E5: BPD or Burst- R0,R1,R2,R3 DMX output : BMD
02 - RPARERR2+	CS	DMX input	E6: BPD or Burst- R1,R3 E5: BPD DMX output : BMA
03 - FBDMX+	CS	Burst-mode DMX transfer	
04 - BURST-INPUT+	CS	1=DMX input, 0=DMX output	
05,06,07 - 0 - FPDPE+	D	Peripheral reports BPD error (output)	
1 - FBRFHPE+	D	Base Register File High	
2 - FMDPE+	D	Memory reports BMD error (write)	
3 - FIIBAPE+	D	Prefetch Buffer address	
4 - FPAPE+	D	Peripheral reports BPA error (output)	
5 - FBRFLPE+	D	Base Register File Low	
6 - FMAPE+	D	Memory reports BMA error	
7 - FIIBIPE+	D	Prefetch Buffer instruction	
08 - ROMPE-	A	RCM parity if no board reported error	
09 - FMDECCU+	D	Memory reports ECC uncorrectable read error	
10 - GDBDPE-	D	Prefetch board detected error	

11 - BPAIPE+	A	BPA input error (DMX or Interrupt)
12 - FRDXPE+	A	RDX error when most recently closed
13 - FRFPE+	A	Register File error
14 - FREAPE+	A	REAH or REAL error
15 - FDMX+	D	DMX cycle at time of error

DSWPARITYL

01 - GOBDPE-	C	C board detected error
02 - FBMDEVPE+	C	BMD input even word
03 - FBMDODPE+	C	BMD input odd word
04 - LMMD+	C	Missing memory module at Cache-Miss
05 - LBMAPE+	C	Memory reports BMA error at Cache-Miss
06 - LFERNEXT+	C	LSB address to memory at error (Cache-Miss)
07 - LFLRMAL15+	C	LSB address to memory at start of Cache-Miss
08 - LMISFL16+	C	Indicator of which memory module was activated
09 - LBMDECCU+	C	Memory reports ECC uncorrectable on Cache-Miss
10 - LBMDECCC+	C	Memory reports ECC correctable on Cache-Miss
11 - LRCIAPE+	C	Cache-Index error on Cache-Read
12 - LRCODPE+	C	Cache-Data-Odd word error on Cache-Read
13 - LRCDEVPE+	C	Cache-Data-Even word error on Cache-Read
14 - LFSERVDBD-	C	Purpose of Cache cycle: 1=Execute, 0=Prefetch

Prime 9950: If DSWPARITY is present, it is broken down by reporting board and signal name as follows.

DSWPARITYH

01 - Set if RCC parity error, bits 3 through 8 are set as follows:

03,04,05	-	Encoding of RCC parity error bits 1-8
06	-	Boolean OR of RCC parity error bits 1-8
07	-	RCC parity error bit 9
08	-	Reset (0)

02 - Set if I/O parity error, bits 3 through 8 are set as follows:

03	-	Set if error is in left byte of BPA or BPD
04	-	Set if error is in right byte of BPA or BPD
05	-	Set if CPU detected a parity error on BPD
06	-	Set if CPU detected a parity error on BPA
07	-	Set if controller detected a parity error on BPD
08	-	Set if controller detected a parity error on BPA

03 -	}	Depends on whether bit 1 or bit 2 is set See description of bits 1 and 2 above for specifics
04 -		
05 -		
06 -		
07 -		
08 -		

09 - Currently unused

- 10 - Set if EI board detected a parity error on BBH, left byte
- 11 - Set if EI board detected a parity error on BBH, right byte
- 12 - Set if EI board detected a parity error on BBL, left byte
- 13 - Set if EI board detected a parity error on BBL, right byte
- 14 - Set if EI board detected a parity error on BAH
- 15 - Set if EI board detected a parity error on BAL
- 16 - Set if EI board detected a parity error on BAE

DSWPARITYL

- 01 - Set if memory control unit detected a parity error on BD, bits 4-7 are set to indicate the error location as follows:
 - 04 - BDH left byte
 - 05 - BDH right byte
 - 06 - BDL left byte
 - 07 - BDL right byte
- 02 - Set if memory control unit detected a latched memory data error, bits 4-7 are set to indicate the error location as follows:
 - 04 - LMDH left byte
 - 05 - LMDH right byte
 - 06 - LMDL left byte
 - 07 - LMDL right byte
- 03 - Set if memory control unit detected a latched memory address error, bits 4-7 are set to indicate the error location as follows:
 - 04 - MCADDR high byte
 - 05 - MCADDR low byte
 - 06 - MCADDR extended byte
 - 07 - Currently unused
- 04 - }
 - 05 - } Depends on whether bit 1, 2, or 3 was set
 - 06 - } See description of bits 1, 2, and 3 above for specifics
 - 07 - }
- 08 - Set if memory control unit detected an ECC uncorrectable error
- 09 - Set if I unit detected an error, bits 10-12 are set to describe the error as follows:
 - 10,11,12 - 0 - No error
 - 1 - Currently unused
 - 2 - Currently unused
 - 3 - Decode net, right byte
 - 4 - Decode net, left byte
 - 5 - Base register file high
 - 6 - Base register file low
 - 7 - Index register file

13 - Set if S unit detected an error, bits 14-16 are set to describe the error as follows:

14,15,16 - 0 - PID or STLB control bits
 1 - LBPA out of STLB in error
 2 - Cache index, right side
 3 - Cache index, left side
 4 - Cache data, high side
 5 - Cache data, low side
 6 - LBVA out of STLB in error
 7 - Branch cache parity error

- MACHINE CHECK USER= nn PC= pppppp

The format of a machine check message on a Prime 300. USER gives the user number, nn (decimal). PC (Program Counter) gives the user's PC at the time of the check.

- MISSING MEMORY DSWSTAT= ...

A missing memory module check occurred. Information is the same as for a machine check, except that the machine check code xxx does not appear.

- MEMORY PARITY (www) DSWSTAT= xxxxxx xxxxxx DSWRMA= xxxxxx xxxxxx
 DSWPB= xxxxxx xxxxxx
 DSWPARITY= xxxxxx xxxxxx PPN,WN= xxxxxx xxxxxx BIT= y OP=z

A memory parity error occurred. www is either ECCC (corrected) or ECCU (uncorrected). The DSWSTAT is the DSW status at last memory check. DSWRMA is the DSW Real Memory Address at last memory check. DSWPB is the DSW Procedure Base at last memory check. DSWPARITY is the DSW Parity at last memory check. PPN,WN is the Physical Page Number and Word Number of the error. OP is the Overall Parity. For an ECCC error, the PPN is followed by BIT = y (y = bit in error.)

<u>xx</u>	<u>Meaning</u>
1-15	bit 1-15
RP	Right parity
LP	Left parity
C2,C4,C5	Other check bits
MB	Multiple bit
NE	No error

This is followed by OP=z, where z=0 or 1, which is the setting of DSWSTATL bit 6 (overall parity).

DSWPARITY is displayed but not decoded.

- POWER FAIL CHECK

A power fail check occurred.

- "Text of Operator remark"

Contents of the REMARK EVENT.

- SHUTDOWN BY OPERATOR

The operator issued a SHUTDN ALL command. (This automatically dumps LOGBUF.)

- TYPE=tt DATA= ddddddd ...

A system logging file entry of type 10-15 was encountered. tt is the type of entry; ddddddd ... is a display of up to nine words of information from the entry.

- WARM START

Indicates a warm start of PRIMOS was performed.

NETLST MESSAGES

- CIRCUIT RESET - a ORIGINATED - b [n] CIRCUIT STATE: xx
CAUSE: cause DIAGNOSTIC: xx

A virtual circuit was reset. a is either LOCALLY or REMOTELY, indicating origin of reset. b is either LOOP-BACK, RING NODE:, or SMLC LINE:. n is the number of either the SMLC line or the ring node.

- COLD START

A cold start of PRIMOS was performed.

- *** END OF FILE — nnnnn ENTRIES, ppppp PROCESSED ***

This message is printed when LOGPRT reaches the end of the network input logging file. nnnnn (decimal) gives the number of entries in the network input event logging file not including date/time and NETBUF overflow entries. ppppp gives the number of entries processed.

- *** FILE EMPTY ***

This message is printed if LOGPRT finds no entries in the network input logging file.

19.3

- LEVEL III PROTOCOL DOWN a [n]

The level III protocol for X.25 is down for this host. a is either LOOP-BACK, RING NODE:, or SMLC LINE:. n is the number of either the SMLC line or the ring node.

- LOCAL PROCEDURAL ERROR CAUSING CLEAR a [n]

19.3

A local procedural error caused a circuit clear in this host. a is either LOOP-BACK, RING NODE:, or SMLC LINE:. n is the number of either the SMLC line or the ring node.

- NETBUF OVERFLOW — nnnnn ENTRIES LOST

Indicates nnnnn (decimal) event entries were lost due to overflow of the event logging buffer (NETBUF). If the buffer frequently overflows, the system may need to be configured with a larger NETBUF, or the buffer may need to be written to the network input logging file more frequently. If NETBUF overflows occur with frequency over an extended period of time, inform your field engineer.

- NETDMP CALLED AT: xxxxxx xxxxxx [DATA: YYYYYY YYYYYY YYYYYY]

A network software problem has occurred at this address. The routine NETDMP was called. On early model machines, NETDMP will dump the three DATA words.

- NPX>R\$CALL>R\$CONN UNKNOWN CIRCUIT STATUS - NODE: xxxxxx (OCT).
VIRTUAL CIRCUIT STATE (1): xxxxxx (OCT).
VIRTUAL CIRCUIT STATE (2): xxxxxx (OCT).

PRIMENET has returned an unexpected status (error) code to NPX.

- NPX>TRNRCV MASTER'S CIRCUIT WAS CLEARED - NODE: xxxxxx (OCT)
VC STATE(1): xxxxxx (OCT). VC STATE(2): xxxxxx (OCT).

The connection between the master and the slave has been unexpectedly broken.

- NPX>TRNRCV MESSAGE OUT OF SEQUENCE IN BOUNCE DETECT. NODE: xxxxxx (OCT) MESSAGE SEQ#: xxxxxx (OCT), NS: xxxxxx (OCT).

NPX break detect/correct logic found message out of sequence. NPX has failed or data has been lost in the network.

- NPX>TRNRCV THROTTLED ON TRANSMIT OR RECEIVE -
NODE: xxxxxx (OCT), MASTER/SLAVE FLAG: xxxxxx (OCT).

NPX has attempted to send or receive a message. Network buffers are too full to proceed.

- NPX>TRNRCV UNKNOWN RECEIVE STATUS - NODE: xxxxxx (OCT).
MASTER/SLAVE FLAG: xxxxxx (OCT). RECEIVE STATE: xxxxxx (OCT).

PRIMENET has returned an unanticipated status (error) code to NPX.

19.3

- PACKET OUT OF SEQUENCE - RING a [n] CIRCUIT STATE: c
SEQ # EXPECTED: d SEQ # FOUND: e

A packet was received with an unexpected sequence number. a is either LOOP-BACK, RING NODE:, or SMLC LINE:. n is the number of either the SMLC line or the ring node.

- POWER FAIL CHECK

A power fail check occurred.

- RING DIM OUT OF RECEIVE BLOCKS

The software controlling the PRIMENET Node Controller (PNC) has been handling enough traffic to temporarily exhaust the available supply of buffers. If this event happens often, the system may need to be built with more buffers to handle this network's message load.

- RING NODE: node-number NOT ACCEPTING XMITTS. XMIT STAT IS xxxxxx

The specified node's PNC is refusing incoming messages. The most common status for xxxxxx is 020100. This indicates that the target node is connected to the network, but PRIMENET software is not allowing incoming messages. Either the target machine is OUT OF RECEIVE BLOCKS (see above) or it is halted.

- SHUTDOWN BY OPERATOR

The operator issued a SHUTDN ALL command.

- SMLC - NO STX PRECEEDING ETX PHYSICAL LINE NUMBER IS xxxxxx
DEVICE ADDRESS IS xxxxxx

Necessary ASCII control strings are missing from the beginning and the end of SMLC packets.

- SMLC RESET FOR LOGICAL LINE xxxxxx - cause

Resets can be caused in six ways: invalid address, command reject, invalid packet id number, invalid response, invalid packet id number on reject, or maximum number of retries exceeded.

- SMLC STATUS ERROR STATUS WORD IS xxxxxx [PHYSICAL LINE # IS n]
DEVICE ADDRESS IS yyyyyy [NUMBER OF OCCURRENCES IS number]

An invalid status, xxxxxx, has been reported by the SMLC. The number of occurrences is printed only on parity errors.

- SYSTEM BLOCKS UNAVAILABLE FOR SMLC PROTOCOL MESSAGE
MESSAGE IS xxxxxx LOGICAL LINE NUMBER IS yyyyyy

The level II synchronous protocol had no buffers in which to send this type of protocol-generated message.

- "Text of operator remark"

Contents of the REMARK event.

- TOKEN INSERTED INTO THE RING NETWORK

The software controlling the PNC hardware issued a ring network control token.

- WARM START

A warm start of PRIMOS was performed.

E

COPY_DISK Error Messages

INTRODUCTION

This appendix contains a listing of error messages generated by the COPY_DISK command. For a discussion of COPY_DISK, see Chapter 7.

COPY_DISK ERROR MESSAGES

- BAD BADSPT FILE ON PARTITION pdev - IGNORED

The badspot file contains an error, and it will be ignored during the rest of the COPY_DISK procedure.

- BAD COMMAND LINE PARAMETER

Illegal parameter abort.

- BAD SURFACE

Illegal disk, or, the disk is not started.

- BADSPOT FOUND track no.

Track ignored.

- BADSPOTS HANDLED ON PARTITION pdev

Badspots have been handled on the indicated partition. FIX_DISK must be run if the partition is to be used for anything other than as a target for COPY_DISK or PHYRST.

- BADSPT FILE ON PARTITION pdev HAS AN EQUIVALENCE BLOCK
PLEASE RUN FIX_DISK

The badspot file on the indicated source partition has an equivalence block. The COPY_DISK program aborts. Run FIX_DISK.

- COPY OF UNEQUAL SIZED PARTITIONS MUST BE
TO/FROM THE REMOVABLE SURFACE OF A CMD

Unequal sized partitions encountered. COPY_DISK will restart.

- DISK RD ERROR device-number Primos-record-number status

A disk read error has occurred.

- DISK WT ERROR device-number Primos-record-number status

A disk write error has occurred. If device-number indicates the target disk, check to make sure it is not write protected. If the status is '177776, check both disk drives for faults. If a fault has occurred, clear it and restart the COPY_DISK procedure.

- ERROR IGNORED, COPY_DISK CONTINUED

PRIMOS has made 10 unsuccessful attempts to read a record and has continued with the DISK_COPY operation.

- ERROR - THIS IS SURFACE n

Error loading CMD removable platter when copying to fixed part.

- ERROR READING DISK no RECORD n IGNORED, COPY CONTINUED.

Read error. Ignore and continue.

- FNDBAD CALLED

Looking for next badspot on disk.

- IF YOU DO NOT WISH TO CONTINUE WITHOUT BADSPOT HANDLING YOU WILL NEED TO RE-MAKE PARTITION pdev
OK TO WRITE IT WITHOUT BADSPOT HANDLING (YES/NO)?

The target partition cannot accommodate the source partition. The badspot information for the target partition will be lost. If you wish to continue the copy, answer YES, and the copy operation will proceed, and no badspot handling will take place. If you choose instead to find another disk with fewer badspots to replace the target disk, answer NO, and the copy operation will be terminated. This message appears in conjunction with the message:

19.1

WARNING - BADSPOT HANDLING DISCONTINUED ON PARTITION pdev(NXTRAT)
NO FREE RECORDS ON PARTITION pdev

- NEED 32K MIN FOR BIG DISK

Not enough memory to copy large disk. COPY_DISK will abort.

- NEW BADSPOT FOUND, RECORD recno, ON PARTITION pdev
IT SHOULD BE ADDED TO THE BADSPT FILE.

COPY_DISK has found a badspot on the disk that is not listed in the BADSPT file. The record will be remapped by COPY_DISK, but the BADSPT file will not be updated by COPY_DISK.

19.2

- NO COPY TO SELF

The user is trying to copy to and from the same disk. Resubmit the TO and FROM pdev specifications.

- NO FREE RECORDS ON PARTITION pdev

No free records are available on the target disk for mapping around the badspot. This message appears in conjunction with the message:

IF YOU DO NOT WISH TO CONTINUE WITHOUT BADSPOT
HANDLING YOU WILL HAVE TO RE-MAKE PARTITION pdev
OK TO WRITE IT WITHOUT BADSPOT HANDLING (YES/NO)?

19.1

A YES response causes the partition to be copied without badspot handling. A NO response causes the COPY_DISK program to exit, allowing the operator to copy to a partition with fewer badspots.

- NOT 1040-RECORD SIZE

Illegal record size. Begin again.

- REC LENGTH AND NR RECS MUST BE = FOR BOTH DEVICES

TO and FROM disks/partitions are not of equal size. Start again.

- SRWREC NOT READY

(CMD copy only.) YES was typed before the disk was ready, and COPY_DISK has aborted. Type S to restart.

- SURFACE READY?

(CMD copy only.) Type Y when new surface is loaded.

- THE MFD HEADER IS CORRUPT OR NOT MADE ON PARTITION pdev
YOU CANNOT COPY FROM A VIRGIN DISK

19.2

The user has attempted to use a virgin disk. COPY_DISK will not operate on a virgin disk at Rev. 19 and above, as it attempts to find a BADSPT file, and on a clear disk, it would seek forever. Use MAKE to format the disk. (See Chapter 6 for information on using MAKE.)

- UNRECOVERED ERROR

Ten successive unsuccessful write operations have occurred.

- WARNING - BADSPOT HANDLING DISCONTINUED ON PARTITION pdev

19.1

Indicates that COPY_DISK is unable to perform the copy operation while handling badspots. Reasons for this include a defective BADSPT file, insufficient space on the target partition, etc. The copy operation will proceed, but no badspot handling will take place.

- WARNING - SOURCE PARTITION IS PRE REV 19
NO BADSPOT HANDLING WILL OCCUR ON PARTITION pdev

The target partition, being identical to the pre-Rev. 19 source partition, will not have badspot handling.

- YOU WILL NEED n SURFACES TO COPY TO, OK?
- YOU WILL NEED n SURFACES TO COPY FROM, OK?

These messages only appear when you are copying a CMD disk. They are intended to inform you of how many removable surfaces will be required by the copy operation. Answer YES if you have the correct number of surfaces, NO if you do not. COPY_DISK will restart if you answer NO, allowing you to enter the correct information.

- VERIFY ERROR record-number word-number

A discrepancy at the indicated location has been detected during the verification procedure. Verification continues.

F

FIX_DISK Error Messages

INTRODUCTION

This appendix contains error messages generated by the `FIX_DISK` command. For a discussion of `FIX_DISK`, see Chapter 9. For a complete description of `FIX_DISK` options, see Chapter 13.

FIX_DISK ERROR MESSAGES

- The Access Category `CATEGORY-NAME` does not reference an ACL!

The ACL pointer of an Access Category does not point to a valid ACL. If the `-FIX` option is specified, the Access Category will be deleted, and all objects that it protects revert to default protection.

- Access category `CATEGORY-NAME` is not pointed at by ACL it points to!

The ACL pointer of an Access Category points to an ACL that doesn't point back to it. If `-FIX` is specified the Access Category will be deleted, and all objects that it protects revert to default protection.

- Access Category CATEGORY-NAME points outside the directory!

The ACL pointer of an Access Category points outside the directory. If -FIX is specified, the access category will be deleted and all objects that it protects will revert to default protection.

- ACL at word XX does not point to a file or Access Category!

The owner pointer of an ACL doesn't point to a file or Access Category. If -FIX is specified, the ACL is deleted.

- ACL at word XX is not pointed at by object it points to!

The owner pointer of an ACL points to an object that doesn't point back to it. If -FIX is specified, the ACL is deleted.

- ACL at word XX points outside the directory!

The owner pointer of the ACL is pointing to something in a different directory. If -FIX is specified, the ACL is deleted.

- The backward pointer of the next record does not point back to the pre-remap record!

19.3

The backward pointer of a record does not point back to the previous record, which has been remapped to a location that does not contain a badspot. In the case of the first record of a file, the backward pointer is not zero. If the -FIX option is specified, the back pointer is fixed to point to the previous record when the BRA word of this record matches the first record address of this file. The file is truncated if the BRA word of this record does not match the first record address of the file.

- The backward pointer is bad. It should be YY instead of XX!

The backward pointer of a record does not point back to the previous record of a file. In the case of the first record of a file, the backward pointer is not zero. If the -FIX option is specified, the back pointer is fixed to point to the previous record if the BRA word of this record matches the first record address of this file. The file is truncated if the BRA word of this record does not match the first record address of the file.

- Bad data count XX in segdir YY!

The segment directory contains a different number of records than the record header indicates it should contain. If -FIX has been specified, the header will be fixed.

- Bad file type: special bit not set!

This message will display if the special bit in the BOOT, MFD, or DSKRAT files has not been set. If the -FIX and -INTERACTIVE options have been specified, the DSKRAT file will be rebuilt. Otherwise it may be necessary to remake the partition.

- Bad Physical Device Number (cl_par)

The physical device number that is specified in the command line is bad.

- Bad header in BADSPT file, ignored!

The header of the BADSPT file is incorrect. FIX_DISK will continue, but badspot handling will be disabled.

- Bad record address: XX
BRA = YY Father = ZZ Type = file-type

Record address XX in the record header is bad. The Beginning Record Address (BRA), Father pointer, and file type are also displayed. If the -FIX option is specified, the address will be corrected, if possible. Otherwise the file will be truncated.

- The BADSPT file is bad, ignored

The BADSPT file that is found by FIX_DISK is bad. Badspots will not be handled.

- The BADSPT file cannot be read, ignored!

The BADSPT file was not found in the current partition. Badspot handling is disabled.

19.3

- The Beginning Record Address (BRA) pointer is bad.

The beginning record address word of the records within the file (except the first record) should point to the first record of the file. If the -FIX option is specified, the BRA pointer is fixed.

- Cannot find comdev in disk list!

-COMDEV was specified as an option on the command line, but the partition specified is not added to the system.

- Cannot process BADSPOT for records less than XX

19.3

A certain number of records is reserved at the beginning of a partition for system information. The size of this area depends on the size of the partition. Badspots cannot occur in this region. This message indicates that a badspot was entered that falls in the illegal region. If there is an uncorrectable badspot in this region, PRIMOS cannot use the disk.

- The current record address (CRA) is bad.

The current record address word of this record does not match the current record. If the -FIX option is specified, the CRA is corrected.

19.3

- The DAM index is too long to represent the DAM file
- The DAM index is too long to represent the data records

The data records of a DAM file are shorter than its index indicates. If the -FIX option is specified, the index is truncated.

- Directory is longer than 64K!

The maximum size of a directory is 64K words. If a UFD exceeds this limit, truncation occurs if -FIX is specified.

- The Directory Used count is bad. It should be YY instead of XX

The directory-used count for this directory does not match the directory-used count that is calculated by FIX_DISK. (The directory file itself and all the files in it are counted.) If the -FIX option is specified, the directory-used count is fixed.

- Disk is full! BADSPT file entry cannot be created.

There is insufficient space on the disk for FIX_DISK to write the bad record that was found into the BADSPT file. FIX_DISK will continue, but badspot handling will be disabled.

19.3

- Disk is full! BADSPT file is incomplete.

There is insufficient space on the disk for FIX_DISK to write the bad record that was found into the BADSPT file. The file is incomplete.

- Disk is full! Cannot allocate another record (fix_disk)

There is insufficient space on the disk for FIX_DISK to allocate another record in the BADSPT file.

- Disk record address XX is illegal!

A record points to address XX which is outside the current partition. If the -FIX option was specified, the address will be corrected.

- Disk read/write error. Record = XX TRACK = YY HEAD = ZZ

An error occurred while reading or writing record XX. If the -FIX and -TRUNCATE options are specified, the file is truncated on read errors and this badspot record is added to the BADSPT file. If the -TRUNCATE option is not specified, a zero record will be added.

19.3

- DSKRAT MISMATCH!

The record allocation information in the DSKRAT disagrees with the record allocation information generated by FIX_DISK during processing, and the -FIX option was not specified.

- DSKRAT UPDATED!

The record allocation information in the DSKRAT disagrees with the record allocation information generated by FIX_DISK during processing, and the -FIX option was specified. FIX_DISK has corrected the DSKRAT file.

- The file structure of DSKRAT is bad

This message is obtained if the DSKRAT file contains any bad record pointers or contains inconsistent information. If both the -INTERACTIVE and the -FIX options are given, FIX_DISK attempts to reconstruct the DSKRAT file structure. Otherwise, FIX_DISK aborts.

- EOF occurs in the middle of an entry

A directory ends in the middle of the last UFD entry. If the -FIX option is specified, the entry will be deleted.

- Error occurs during validation

FIX_DISK was unable to read the MFD record on a file system disk. The partition may need to be remade.

- Fatal error has occurred in the file structure of a DAM segment directory.

FIX_DISK was unable to fix the pointers in a DAM segment directory. The directory will be truncated or deleted.

19.3

- Fatal error during remapping: record not found in dam index

A record was found to be bad but `FIX_DISK` was unable to remap it to a new location. The record will be deleted.

- The father pointer is bad.

The beginning record address word of the first record of a file does not point to the beginning record address of the directory or `SEGDIR` in which this file is entered (its father). If the `-FIX` option is specified, the file is deleted.

- File `FILE-NAME` does not reference an ACL or Access Category

The ACL pointer of a file doesn't point to a valid ACL or Access Category. If `-FIX` is specified, the file reverts to default protection.

- File category `FILE-NAME` is not pointed at by ACL it points to

The ACL pointer of a file entry points to an ACL which doesn't point back to it. If `-FIX` is specified, the pointer is set to the default value.

- File entry in password ufd has non-zero acl pointer

`FIX_DISK` has found a file in a password UFD that has an ACL set on it. If `-FIX` was specified, the pointer will be corrected.

- File type mismatch

The file type in the first record of the file does not agree with the file type in the UFD entry. If the `-FIX` option is specified, it is deleted.

19.3

- File `FILE-NAME` points outside the directory

The ACL pointer of a file is pointing to something in a different directory. If `-FIX` is specified, the ACL pointer is changed to the default type.

- The first file entry of the MFD file is not `DSKRAT`

`FIX_DISK` checks that the first entry in the MFD is `DSKRAT`. If this entry is missing, `FIX_DISK` aborts.

- The forward pointer XX is bad, it is not within the range of the current partition

The address that the forward pointer points to is not between zero and the maximum record address of this partition. If the -FIX option is specified, the file is truncated.

- The forward pointer of the top level index record of a DAM file is not zero. | 19.3

The top level index must only be one record long; therefore, the forward pointer of this record must be zero. If the -FIX option is specified, the pointer will be set to zero.

- The forward pointer XX is bad, it points to a record that belongs to another file. | 19.3

The record that the forward pointer points to belongs to another file. This error may occur if the current DSKRAT is bad or the BADSPT file is changed after the previous FIX_DISK was run. If the -FIX option is specified, the file is truncated.

- The forward pointer of the previous record does not point to the pre-remap record

The forward pointer of the previous record points to another file after having been remapped. This error may occur if the current DSKRAT is bad or the BADSPT file is changed after the previous FIX_DISK was run. If the -FIX option is specified, the file is truncated.

19.3

- Handling soft disk error recovery for XX

FIX_DISK has found a correctable error at record number XX, and will add it to the badspot file.

- Inconsistent entry. Record = XX, Word = YY

The information in an entry within a UFD is not self-consistent and cannot be reconciled. If the -FIX and -DUFE options are specified, this entry of this file is changed to vacant.

- The index of this DAM file is too short. The DAM index may be rebuilt

The data records of a DAM file are longer than its index indicates. If the -FIX option is specified, the index is fixed if the extra index words will fit into the index record.

19.3

- The index of this DAM file is too short to represent the data records

19.3

The data records of a DAM file are longer than its index indicates. The file is truncated because there is insufficient space for the extra index words.

- The index level of this DAM file is incorrect. It should be YY instead of XX

The index level word of this record is incorrect. It should be zero for SAM files or one less than the previous level for DAM files. If the -FIX option is specified, the index level word is fixed.

- The index level of the top index record of a DAM file is incorrect.

The index level of the top index record of a DAM file is incorrect. If -FIX was specified, the index record will be corrected.

- Internal error - UFD overflow!

19.3

The number of files, and the length of their file names is greater than the amount of space allotted for them in the FIX_DISK data base. This error is unlikely to occur unless you have a partition with an extremely large number of small files with long (32 character) filenames, and many deeply nested directories.

- Missing badspot number. (cl_par).

You specified the -ADD_BADSPOTS option, but did not enter any badspot records on the command line.

- The next index does not match the forward pointer of the current data record

The pointers of the index section and the data section do not agree. If the -FIX option is specified, the following actions will be taken. The back pointer of the record that is pointed to by the DAM index and the back pointer of the record that is pointed to by the forward pointer of the current data record are examined. The record whose back pointer points to the previous data record will be chosen. If neither back pointer points to the previous record or both back pointers point to the previous record, the file is truncated.

19.3

- Parent of a file is not a UFD or SEGDIR

The pointer to a UFD or SEGDIR identifies a file header. If -FIX is specified, this will be corrected.

- Partition not shutdown correctly during the previous session

This message is issued if the partition was not shut down with the SHUTDN command under PRIMOS. If the system crashed or the disk drive was spun down instead, this message will result.

- Physical Device Number is missing. (CL_PAR)

The physical device number is not specified in the command line.

- The Quota system may be incorrect

This message is issued if the partition was changed under PRIMOS II or if the disk was not correctly shut down. Since PRIMOS II doesn't support quotas, there may be directories on this partition with incorrect quota information.

- The file structure of DSKRAT is bad. (RAT_CK).
- The length of UFD header is incorrect.
- The number of heads is different. It should be YY is XX.
- The physical record size is different. (RAT_CK).
- The DSKRAT header has wrong length. (RAT_CK).
- The partition cannot be handled by this version of FIX_DISK.

The information contained in the DSKRAT header does not correspond to the information computed from the disk number. Either the disk number is incorrect or the DSKRAT header contains incorrect information. If the -INTERACTIVE and -FIX options are omitted, FIX_DISK aborts. Otherwise, Otherwise FIX_DISK asks:

FIX DSKRAT?

A NO response causes FIX_DISK to abort. A YES response initiates a dialog that results in fixing the DSKRAT.

FIX_DISK computes the number of records in the partition from the disk number. In case of ambiguity, FIX_DISK asks resolving questions, such as:

Storage module or CMD?

| 19.3

FIX_DISK then asks:

Split disk?

If part of the disk is to be used for paging, then answer YES; otherwise, answer NO. If the answer is YES, FIX_DISK then asks:

Paging Records (Decimal)?

Type in the number of records to be used for paging. FIX_DISK then prints the disk number, file records, and paging records in the form:

19.3 | DISK FILE-RECORDS PAGING-RECORDS (DECIMAL).
 XX YY ZZ

and asks:

Parameters OK?

If the numbers are incorrect, answer NO. FIX_DISK will attempt to recompute the parameters. If the numbers are correct, answer YES. FIX_DISK then asks:

Does this partition support Acls or Quotas?

Answer YES if the partition is to be a Rev. 19 format partition, NO if it is to be a pre-Rev. 19 format partition. Only Rev. 19 format partitions support ACLs and Quotas.

- Record address XX is out of range YY.

The next record address XX is out of range of the current record address YY in a segment directory. If -FIX is specified, the file will be deleted.

- The 1st file entry of the MFD is not DSKRAT

FIX_DISK checks that the first entry in the MFD is DSKRAT. If this entry is missing, and the -FIX and -INTERACTIVE options were not specified on the command line, FIX_DISK aborts.

- The 2nd file entry of the MFD file is not MFD

FIX_DISK checks that the second entry in the MFD is MFD. If this entry is missing, FIX_DISK aborts.

- The SEGDIR is longer than 64K words

The maximum size of a SEGDIR is 64K words. If a SEGDIR exceeds this limit, it will be truncated if the -FIX option is specified.

- Structure error for a DAM segdir. (SEGDR_CHK)

The header information for the DAM segment directory is inconsistent with the file information. If -FIX is specified, the inconsistency will be corrected.

- System file is bad - ignored!

An error that would normally cause deletion of a file has been found in one of the MFD's special files (BOOT, MFD, DSKRAT). The file will not be deleted.

- WARNING: The 3rd file entry of the MFD file is not BOOT
The partition may need to be remade

FIX_DISK checks that the third entry in the MFD is BOOT. If this entry is missing, FIX_DISK will not abort. If -UFD_COMPRESSION has been specified on the command line, the message "Compression has been disabled" will also be displayed.

- The Tree Used count is bad. It should be YY instead of XX

The tree-used count of this UFD does not match the tree-used count that is calculated by FIX_DISK. If the -FIX option is specified, the tree-used count is fixed.

- Treename is not available.

FIX_DISK found an error on its second pass in this record. Information on what file the record belongs to is missing. This should only happen if the disk is failing.

19.3

- 2 files point to the same record

Two files within the same UFD have the same Beginning Record Address (BRA). If the -FIX option is specified, all entries for the second and subsequent files pointing to that BRA will be deleted.

- The UFD header is missing

The UFD header is missing. If the -FIX option is specified, the UFD file is deleted.

- The UFD header length is incorrect

The UFD header length is wrong. If the -FIX option is specified, the length will be changed.

- Ufd nesting exceeds maximum specified

FIX_DISK cannot follow the directory tree because the user has nested directories to more than *n* levels (default *n* = 100). FIX_DISK aborts unless the `-AUTO_TRUNCATION` option is specified, in which case directories that are nested too deeply will be truncated.

- Unable to add badspot (FIX_DISK).

FIX_DISK was unable to add a badspot to the BADSPT file.

- Unable to add zero record(s). (Fix_disk)

FIX_DISK found an unreadable record, but was unable to add a record of zeros. The file will be truncated.

- Unable to skip second badspot in this file. (FIX_DISK)

FIX_DISK has found more than one bad record in one file, and is unable to keep track of how many bad records were found. If `-FIX` is specified, the file will be truncated.

- Unknown entry type XX Record YY Word ZZ

FIX_DISK does not recognize the entry type of the current record. If `-FIX` is specified, the file to which the record belongs will be truncated.

- Unknown file type XX, Record = YY, Word = ZZ

The file type XX in the file entry is unknown. It is either an illegal file type or a new file type that is not known by this version of FIX_DISK. If the `-DUFE` and `-FIX` options are specified, this file entry is deleted. If these options are omitted, the file entry is left untouched, and no compression is performed for the UFD in which this file entry resides.

- Unknown logical file type.

FIX_DISK does not recognize the logical file type. If the `-FIX` option was specified on the command line, FIX_DISK will change the file type to the default type.

- Unknown number of records removed.

FIX_DISK has found more than one contiguous bad record. Two records of zeros will be added and any additional good records will follow the zero records.

19.3

- Unrecoverable read error (VFY_RAT)
- Unrecoverable read error while reading the DSKRAT file (RAT_CK)

These messages may occur if the DSKRAT file contains any bad record pointers or contains inconsistent information. If both the -INTERACTIVE and -FIX options are given on the command line, FIX_DISK attempts to reconstruct the DSKRAT file structure. Otherwise, FIX_DISK aborts.

- The word count of record XX is bad

The data word count of a record is not reasonable. For every record except the last record, the data word count should equal the record data size. The data word count of the last record should be between zero and the record data size. If the -FIX option is specified, the word count is set to the appropriate value.

- Warning: Illegal reference to record zero ignored.

A pointer was found pointing to the BOOT file from a record unrelated to the BOOT file. This message indicates corruption of a directory entry. If the -FIX option was specified, the entry will be deleted.

- Zero record will be added to file.

FIX_DISK has encountered an unreadable record. A record of zeros will be added to mark the location of the unreadable record.

19.3

G

Batch Messages

INTRODUCTION

Messages in this appendix include both those sent to users and those most often seen at the supervisor terminal. Some merely report the progress of a job. Others report mild or serious errors.

When a serious problem occurs (for example, when the Batch monitor discovers that the Batch data base has been damaged), three things generally occur:

- A message is sent to the supervisor terminal.
- The bell at the supervisor terminal rings.
- The Batch monitor logs itself out.

When this occurs, the operator should look at the Batch monitor log file (if one was created), the error message sent to the supervisor terminal, and the file BATCHQ>ERROR. By looking at these three sources of information, the operator can discover whether the error now being reported is the result of an earlier, unreported error.

BATCH MESSAGES

Following is a list of Batch messages. The nature of each message (for instance, warning, query, etc.) is indicated in parentheses at the beginning of each explanation.

- <nn> is out of range. <option>

(Fatal) The numbers supplied as parameters to the -FUNIT or -PRIORITY options were out of range. The range for -FUNIT is from 1 to 126, and -PRIORITY is from 0 to 9. The job should be resubmitted or changed to use legal -FUNIT and -PRIORITY values. Note that the system may be configured to have fewer than 126 units per user at cold start; the -FUNIT argument will then be limited to the maximum configured unit number.

- <text> seen when end-of-line expected.

(Warning) text was seen when there should have been no more text (at the end of line). This error message is a warning, but it may indicate differing degrees of error severity depending on the program being run. Usually, the command line that was read will be lost. In BATGEN command/subcommand mode, the user will be left in command/subcommand mode. When this message occurs during the use of JOB or BATCH, or while entering the BATGEN command, the user is returned to PRIMOS, although the "ER!" condition is not raised.

- Bad \$\$ command.

(Fatal) A command file submitted using the JOB command had a \$\$ line as the first noncomment line, but the \$\$ command was not a \$\$ JOB command. The command file should be changed so that the \$\$ line is legal. The use of \$\$ is reserved for future expansion by BATCH.

- Bad queue control file.

(Severe) One of the Batch subsystem data base files is inaccessible or has a bad format. The Batch subsystem is therefore inoperative until the data base is fixed.

- Bad queue definition file.

(Fatal) A file referenced by BATGEN did not comply to format requirements; it was not a legal queue definition file. If this error occurs anywhere other than the BATGEN program, then the system Batch definition file has been overwritten with illegal data, and the Batch subsystem is inoperative.

- BATDEF file is missing.

(Message) The queue definition file, which is the crux of the data base, is not present. The monitor will log itself out after sending this message. The System Administrator should use BATGEN to generate a new BATDEF file.

- Can't log error.

An error has occurred that the monitor could not record. (This message generally accompanies other error messages.)

- Can't start batch job!

(Message) The Batch monitor was not started from the supervisor terminal, and it cannot log in processes under different login names or log out other processes. The monitor will log itself out gracefully after sending this message. Simply issue the BATCH -START command from the supervisor terminal if this happens.

- (Changes made)

(Response) The changes specified in a JOB -CHANGE operation have been made. If the job is initiated after the changes are made, then it will execute with the specified changes in place. The job status will be displayed after the above message is typed out.

- Command or CPL file required as first argument on submission.

(Fatal) The JOB command was given with job options (such as -HOME, -PRIORITY, -CPTIME, etc.) but no command file was seen before those options. The syntax is "JOB treename [options]".

- Cpu limit must be specified.

(Fatal) The queue referred to by a -QUEUE option during job submission is defined such that the -CPTIME option is a required parameter. (That is, the default CPU limit for that queue is greater than the maximum CPU limit for that queue.) The job should be resubmitted with the -CPTIME option specified. To determine the maximum limits for queues, use BATGEN -DISPLAY.

- Creating new batch definition file: <treename> (BATGEN)

(Response) The treename specified does not exist. When the FILE command is given, it will create the specified file and put the batch definition in it. BATGEN will initialize its batch environment to a null state when it can't find treename, so that no queues are initially defined.

- Database invalid.

(Message) This is a severe error. A fatal error will be generated along with this message. The monitor will log itself out after sending this message, and the Batch system will be left inoperative. (Users

will receive error messages if they try to invoke JOB or BATCH.) The System Administrator should determine what the error was and fix it if possible. If the Batch monitor runs a COMOUTPUT file, that should reveal the source of the error. The file would be named O_LOG in BATCHQ (if the file BATCHQ>START_BATCH_MONITOR.COMI runs FIXBAT.SAVE with a -STARTUP argument other than NOLOG).

In general, if the exact cause of the problem is not known (such as a Pointer mismatch error in the data base, or a disk write-protected error), FIXBAT should be run. If that fails, the BATCHQ>INIT program should be resumed using the -RESET_QUEUES option to reinitialize the entire data base. If this doesn't work, there are probably disk errors. If it does work, redefine the Batch queues using BATGEN and start the Batch monitor up again. All job data will have been lost.

- Date and time not set. (Batch)

(Fatal) A BATGEN or JOB command, or BATCHQ's INIT program, was issued from the supervisor terminal before the system date and time had been set. These parts of the Batch system cannot be run until the system date and time are set using the SETIME command from the supervisor terminal.

- Elapsed time limit must be specified.

(Fatal) The queue referred to by a -QUEUE option during job submission is defined such that the -ETIME option is a required parameter. (That is, the default elapsed time limit for that queue is greater than the maximum elapsed time limit for that queue.) The job should be resubmitted with the -ETIME option specified. To determine the maximum limits for queues, use BATGEN -DISPLAY.

- End of line.

(Fatal) A required keyword or option was not present on the command line. The message will generally contain more information on what was expected. Reenter the command with the additional requested information.

- End of line. Illegal <option> argument

(Fatal) One of the job parameter options was specified on the JOB command line, but had no argument (end of line). The information required by option should be supplied when the command is reentered.

- End of line. Queue name required

(Warning) A command entered while in BATGEN command mode required a queue name. (ADD, MODIFY, BLOCK, UNBLOCK, and DELETE all require queue names.) Reenter the command with the queue name desired.

- End of line. Value required

(Warning) While in BATGEN subcommand mode, a subcommand was given that required at least one numeric parameter, but none was given. Subcommands requiring at least one numeric parameter are CPTIME, ETIME, FUNIT, PRIORITY, TIMESLICE, and RLEVEL. Note that the CPTIME and ETIME subcommands accept two parameters, both of which may be the keyword NONE indicating no limits. Reenter the subcommand with the value desired. Example: FUNIT 13.

- Enter queue characteristics:
\$

(Response) The ADD or MODIFY command, given while in BATGEN command mode, succeeded. The user is now in BATGEN subcommand mode, identified by the '\$' prompt instead of the '>' prompt given when in BATGEN command mode. To reenter command mode from subcommand mode, use QUIT or RETURN. RETURN saves the information changed while in subcommand mode; QUIT discards it, asking for verification if any of it was changed.

- Environment modified, ok to quit?

(Query) A QUIT command was issued while in BATGEN command mode, after the environment was modified. Legal answers to this question are YES, NO, and OK. If YES or OK is the response, a subsequent START command will reenter BATGEN command mode with no loss of information about the environment.

- Extraneous text on command line (MONITOR)

A bad command line exists in BATCHQ>START_BATCH_MONITOR.COMI. The command line should read "RESUME MONITOR" or "RESUME MONITOR -HUSH"; but some excess information currently follows the -HUSH option.

- File has no non-comment lines. filename (JOB)

(Fatal) A user has submitted a command file or CPL file that either is empty or is made up entirely of comment lines.

- Home UFD required.

(Fatal) The `-HOME` option was not present on the JOB or the (optional) `$$ JOB` line during submission, and the program was unable to determine the attach point of the submitting job. Resubmit the job, and include the `-HOME` option followed by the absolute pathname indicating where the job is to execute. If the pathname is too long to fit, use a shorter description of it when you resubmit the command file. First, edit the file to include an `ATTACH` command with a relative pathname that will descend the remaining sub-UFDs to reach the destination.

- Home=<pathname>

(Response) During job submission, the `-HOME` option was not specified on the command line or in the command file (`$$ JOB`), but the job did successfully submit. The JOB command determined the home attach point of the submitting job. This message is typed out (where pathname becomes the home UFD for the submitted job) to remind the user that the `-HOME` option was not specified.

Note

JOB does not attempt to determine whether the user can attach to the home pathname as owner. If the user cannot attach, because of either a "bad password" error or an "insufficient access rights" error, the job may terminate, and a requested command output file may not be produced.

- Illegal `-CHANGE` option.

(Fatal) The options `-QUEUE` and `-PRIORITY` are illegal during a `-CHANGE` operation using the JOB command, as queue and queue priority of a job cannot be changed. Cancel or abort the job and resubmit it into the appropriate queue with the desired queue priority.

- Illegal answer.

(Warning) This warning is output when the answer to a question is not YES, NO, or OK. The question will be asked again. These questions are asked when a user tries to QUIT out of BATGEN command or subcommand mode after modifying the environment.

- Illegal combination. <option>

(Fatal) A job parameter (such as `-ACCT`, `-HOME`, `-QUEUE`, etc.) was specified on the same JOB command line as an option to perform a certain action (such as `-CANCEL`, `-DISPLAY`, `-ABORT`, etc.). Use separate JOB commands to perform separate functions.

- Illegal combination. -FUNIT (JOB)

(Fatal) A CPL job was submitted using the -FUNIT option. This option is not valid for CPL jobs. Resubmit the job without the -FUNIT option.

- Illegal limit.

(Fatal) The parameters supplied to the -CPTIME or -ETIME options during job submission/changing were not legal limits. That is, they were less than or equal to zero, were not legal decimal numbers, and were not the keyword NONE. Reenter the command with legal limits.

- Illegal name.

(Fatal) One of the Batch programs was expecting a name or command, but it read an unquoted token beginning with a dash ('-'), indicating that an option was present.

- Illegal number. <text> (BATGEN)

(Warning) The numeric parameter for a BATGEN subcommand was not a legal decimal number. Reenter the line with a legal decimal number. (All numbers input by Batch software are decimal.) Subcommands that may return this error are CPTIME, ETIME, FUNIT, PRIORITY, TIMESLICE, and RLEVEL. Note that the CPTIME and ETIME subcommands will accept the keyword NONE indicating no limits, but will flag the number 0 as an "illegal number". Also, these two subcommands interpret the numbers as FORTRAN INTEGER*4 numbers (ranging from 1 to 999999999), whereas the other subcommands use INTEGER*2 (ranging from 0 to 32767).

- Illegal number. <text> (JOB)

(Fatal) The argument for the -FUNIT or -PRIORITY option during job submission using the JOB command was not a legal decimal number. Reenter the command line with legal numeric parameters.

- Illegal option.

(Fatal) One of the Batch programs was expecting an option, namely, an unquoted token beginning with a dash ('-'). Reenter the command line with a legal format.

- Illegal queue name. <text> (BATGEN)

(Warning) An attempt was made to add a queue that had a name which did not comply with filename rules. (These rules are: the first character must not be a digit; and the character set is limited to alphabets, digits, and selected special characters). Reenter the command with a legal queue name. Note that a queue name of ALL is illegal, so that the DELETE ALL will not be issued except when deleting all queues is desired.

- Illegal queue name. <text> (JOB)

(Fatal) The queue name specified after a -QUEUE option while submitting or changing a job did not comply with queue name format rules. Use BATGEN -STATUS or -DISPLAY to determine the names of legal queues.

- In filename:

(Fatal) This opening phrase precedes JOB error messages when the errors originate in a \$\$ JOB line within the file filename. The error message also includes the \$\$ JOB line itself.

- In the submission file:

(Fatal) This opening phrase precedes JOB error messages when the errors originate in the \$\$ JOB line of a file, and the submission program cannot determine the file's pathname to print it.

- IN.USE not open.

(Message) The file which the monitor keeps open for writing while it is running has been mysteriously closed. The monitor will log itself out after sending this message. This is sometimes the result of an accidental shutdown of the disk that the monitor uses (where BATCHQ resides). Or, the CLOSE BATCHQ>IN.USE command has been given from the supervisor terminal. After determining that the BATCHQ UFD exists, restart the Batch monitor.

- Incorrect username.

(Fatal) A command file was submitted using the JOB command that had a \$\$ JOB line as the first non-comment line, but the username specified after the "JOB" specifier did not match the username of the submitting user. Edit the command file and change the username in the \$\$ JOB line to the username of the submitter.

- Info in BATCHQ>ERROR. (BILD\$B)

(Severe) The source of an error has been successfully written to the file "BATCHQ>ERROR." for perusal by the System Administrator (note that the period is included in the treename). This message is usually preceded and followed by other severe error messages.

- *** Invalid batch database, please contact your system administrator.

(Severe) This message means that the running job detected an error (such as disk failure, pointer mismatch, or misprotected file) in the Batch system data base. It will flag the data base as invalid. The System Administrator should be notified, as he or she has the responsibility for reinitializing the data base, running FIXBAT, or running FIXRAT, as appropriate. The BATCH and JOB commands will be inoperative until the situation is resolved.

- ?Job <extrnam><intrnam> <status>.

(Warning) An attempt was made to use the JOB command on a job, but its status prevented such an operation. Examples are trying to restart a completed job and attempting to release a job that is not held.

- Job <extrnam> for <username><intrnam> <status>.

(Message) This message is output by the Batch monitor when it changes the status of a job (except when it changes a restarted job back to "Waiting"). extrnam is the external name of the job, username is the submitting user, intrnam is the internal name, and status is either "aborted" or "completed".

- Job name required.

(Fatal) The options -CHANGE, -CANCEL, -ABORT, -RESTART, -HOLD, and -RELEASE all require a job identifier (an internal or external name). Reenter the command with the job-id. Examples:

```
JOB C_TOP -HOLD
JOB #10032 -ABORT
```

- (Job no longer restartable)

(Response) A JOB -CANCEL was performed on an executing job. The job itself has not been canceled, but it has been flagged as being unrestartable. In this state, use of the -RESTART option will abort the job but will not restart it.

- (Job not changed.) Queue not found. queueName (JOB)

(Fatal) A requested -CHANGE to a job cannot be done because the queue to which the job was submitted cannot be found in BATCHQ>BATDEF.

- Job not found.

(Fatal) The job referred to, in a JOB command such as -CHANGE, -CANCEL, -ABORT, -RESTART, -HOLD, or -RELEASE, could not be found by searching the active jobs list. This could mean one of three things: that no job exists with that name; that all jobs with that name have completed, aborted, or canceled; or that a job exists with that external name but the user making the request is not the same user who originally submitted the job.

- (Job not restartable)

(Warning) A JOB -RESTART was performed on an unrestartable job. An attempt will be made to abort the job.

- (Job restarted)

(Response) A JOB -RESTART was performed on a restartable job. Although an error message may appear after this message, the job will generally be restarted unless a JOB -CANCEL or JOB -CHANGE -RESTART NO command is issued. Possible errors after this message include "Insufficient access rights" if the user is logged in as SYSTEM and has restarted another user's job from a user terminal, or if the process has recently logged out. "Not found" may also be returned if the process is logged out.

- Job will be restarted.

(Message) This is sent to the supervisor terminal after a "Job <extrnam> for <username>(<intrnam>) aborted/completed" message is sent, at the time when the Batch monitor is first started up. It means that the job is eligible for restarting, and that it is therefore being reset to the waiting state. The message generally indicates that the job will be restarted following a system shutdown.

- *** Jobs are not being processed at this time.

(Severe) If followed by "**** Please contact your system administrator immediately", it indicates that the Batch data base has not been initialized, or that something has happened (such as a disk head crash). If followed by "**** Please try again later", it indicates that the Batch monitor was logged out using a method other than "BATCH -STOP", but it will verify the validity of the data base when it is started up. In either case, the user will be immediately returned to command mode; the operation the user attempted will not be performed. This message can be typed out by the BATCH or the JOB commands when they start running.

- Monitor already started.

(Message) The monitor is started already. This message is just a reminder, not a fatal error.

- Monitor in operation.

(Message) The Batch monitor has finished fixing the data base (by running FIXBAT) and is ready to process jobs.

- Monitor started up.

(Message) The monitor has been started up. It is now going through an initialization phase.

- Multiple jobs with this name (use internal name).

(Fatal) The external name used in the JOB command belongs to at least two jobs of this user. The internal name must be used in this case. Use JOB -STATUS to determine the internal and external names of all active jobs belonging to the user issuing the command.

- Multiple occurrence.

(Fatal) An option was specified twice either the JOB or \$\$ JOB line during job submission or job changing on (An example is: JOB TEST -HOME HERE -HOME THERE). If an option is specified once on the JOB line and once on the \$\$ JOB line, no error will result. The parameter on the JOB line will take precedence. Reenter the command, but specify each option only once.

- Multiple monitors illegal.

(Message) An attempt was made to start up a second Batch monitor. The monitor that sent this message will log out.

- Must be first option.

(Fatal) The option `-CHANGE`, `-CANCEL`, `-ABORT`, `-RESTART`, `-STATUS`, `-DISPLAY`, `-HOLD`, and `-RELEASE` must be the first option on the JOB command line (after a sometimes-optional job identifier). Use the JOB command several times to perform several operations.

- My disk is full. Please help me.

(Message) The Batch monitor has encountered a "Disk Full" condition while trying to initiate a job. It will retry the job initiation every five minutes, sending this message at each unsuccessful attempt. This message causes a bell to ring at the supervisor terminal. The operator should either delete some files from the disk to free up space, or should run `FIXBAT` with the `-DAYS` option to remove old files from the data base. (See Chapter 11 for details on `FIXBAT`.)

- My quota is exceeded. Please help me.

(Message) The Batch monitor has encountered a "Quota Full" condition while trying to initiate a job. It will retry the initiation every five minutes, sending this message at each unsuccessful attempt. This message causes a bell to ring at the supervisor terminal. The operator should either delete some files from the disk to free up space, or should run `FIXBAT` with the `-DAYS` option to remove old files from the data base. (See Chapter 11 for details on `FIXBAT`.)

- No active jobs [named "jobname"]

(Response) This message will have either "for user username" or "in system" appended to it, depending on whether or not the user is logged in as `SYSTEM`. This message is typed out by a JOB `-DISPLAY` or `-STATUS` command, and indicates that there are no waiting, held, or executing jobs belonging to that user. If the user is `SYSTEM`, then there are no jobs that are waiting, held, or executing in the entire system.

The text in brackets is displayed if a jobname was specified for the `-DISPLAY` or `-STATUS` command; otherwise it is omitted.

- No configured queues.

(Response) A `BATGEN` invocation of `-STATUS` or `-DISPLAY` found that there were no defined queues.

- No job changes specified.

(Fatal) The `-CHANGE` option was given to the JOB command, but no actual changes were specified on the command line. Specify changes to be made following the `-CHANGE` option on the command line.

- No jobs named jobname in system.

(Response) The operator has requested information on a job (and has specified the job's internal name), but the monitor can find no active jobs with that jobname.

- No longer executing.

(Fatal) A JOB -ABORT or JOB -RESTART was performed on a job that had execution status, but by the time the execution file was read in to determine the user number of the process, the process had disappeared. If the message "(Job restarted)" had been typed out, then the job would have been restarted. Although the operation itself was unsuccessful, the desired results were achieved.

- No queue available for job.

(Fatal) A job submitted with the JOB command did not specify a -QUEUE option, and no suitable queue could be found. Suitability requirements include CPU and elapsed time limits within the confines of the queue, queue unblocked, and so on. Use of the BATGEN -STATUS or -DISPLAY command will yield a list of legal queues and their status.

- No queues have waiting or held jobs.

(Response) When a BATCH -DISPLAY command was issued, there were no queues with any waiting or held jobs. A queue may have one executing job not considered to be waiting or held.

- No right. Must be logged in as SYSTEM or BATCH_SERVICE.

(Fatal) A -HOLD or -RELEASE operation was attempted using the JOB command, and the user was not logged in as SYSTEM or BATCH_SERVICE.

- No running jobs.

(Response) When a BATCH -DISPLAY command was issued, no jobs were currently running. Jobs can be waiting when there are no running jobs, however, even when the monitor is running and there are free phantoms. There is always a small amount of turnaround time between submittal and execution of a job.

- Not an absolute treename.

(Fatal) The home UFD specified with the -HOME option of the JOB command was a relative treename. That is, it began with "*>". Resubmit the job, giving an absolute pathname after the -HOME option.

- Not your job.

(Fatal) A job was referenced using an internal name in the JOB command, but it did not belong to the user making the reference. Use the JOB -STATUS command to obtain a list of all active jobs submitted by the requesting user.

- Note: the batch monitor is currently not starting up jobs.

(Response) A job has been submitted while the monitor is in a paused state. The job will execute when the monitor is continued.

- Null home UFD.

(Fatal) The home UFD specified with the -HOME option of the JOB command was a null string. Resubmit the job with an absolute pathname after the -HOME option.

- Operator stop.

(Message) The monitor received a stop request via a BATCH -STOP command. The monitor will log out after sending this message.

- Out of range.

(Warning) A BATGEN subcommand was given a numeric parameter that was out of range for that subcommand. The ranges are: 1 to 126 for FUNIT, 0 to 9 for PRIORITY, 1 to 99 for TIMESLICE, and 0 to 7 for RLEVEL. Reenter the subcommand with the correct parameter. Note that the FUNIT argument may have a smaller upper limit than 126, depending on the coldstart configuration of the available units per user.

- PHANTOM nn: Change

(Fatal) The Batch monitor cannot process jobs correctly, because the FILUNT directive in the CONFIG file is set to a value less than 16. Change the FILUNT directive.

- Please FILE.

(Warning) A QUIT command was issued while in BATGEN command mode, after the environment had been modified; the question "Environment modified, ok to quit?" was asked, and the answer was NO. This message is a reminder to file out a modified environment.

- Please RETURN.

(Warning) A QUIT subcommand was given while in BATGEN subcommand mode, after the queue characteristics had been modified; the question "Queue definition modified, ok to quit?" was asked, and the response was NO. This message is a reminder that the proper way to leave a subcommand session is to use the RETURN subcommand.

- Please stand by.

(Response) This message and others like it (for instance, "File in use, please stand by") will be displayed if the program being run is trying to gain access to a file that is in use for more than 5 seconds. After 5 seconds, the "Please stand by." message will be output. After 30 seconds, the message "File in use, please stand by." will be output. After 60 seconds, the message "Timeout of 60 seconds has occurred." will be output and the program will abort. Usually this is a fatal error, which could indicate that system security is broken.

- Please wait.

(Response) This message asks that the user be patient because the current program has been monopolizing the Batch data base and other processes now have access to it. It is not a fatal error. It generally is displayed only when a system is heavily loaded, or when the current process has a very low priority and does not run frequently.

- Queue <name> already exists (status).

(Warning) While in BATGEN command mode, an attempt was made to add a queue which already existed. The status referred to is "blocked", "unblocked", or "flagged for deletion". To change the queue definition, use the MODIFY subcommand. However, if the queue is "flagged for deletion", any attempt to block, unblock, modify, or display it will return the "Unknown queue name" error.

- Queue <name> deleted.

(Message) The queue referred to was flagged for deletion in the BATDEF file and has just been deleted by the Batch monitor, as the queue became empty.

- Queue <name> flagged for deletion.

(Warning) While in BATGEN command mode, an attempt was made to delete a queue which was already flagged for deletion. To allow the queue to disappear, file out the BATDEF file. The queue will disappear when it contains no more waiting, held, or executing jobs. It can then be added again.

- Queue blocked.

(Fatal) The queue referred to by a -QUEUE option during job submission is currently blocked to new submissions. Try it again later, or use another queue.

- Queue definition modified, ok to quit?

(Query) A QUIT subcommand was given while in BATGEN subcommand mode, and the characteristics of the queue being added or modified have been changed. Legal answers to this question are YES, NO, and OK. Hitting the carriage return also causes the QUIT to be taken as if YES was typed.

- Queue deleted.

(Fatal) A job was submitted to an available queue, but by the time the command file had been copied and some other activities had taken place, the queue had been deleted. The job should be resubmitted to a different queue.

- Queue does not exist.

(Fatal) The -QUEUE option on the JOB command line or the (optional) \$\$ JOB line referred to a queue that either did not exist or was in the process of being deleted ("flagged for deletion"). The BATGEN -STATUS or -DISPLAY command will provide a list of currently available queues and the status of each queue.

- Queue full.

(Fatal) The queue to which the user has tried to submit the job already contains 10,000 jobs (whether active or inactive). The queue must be deleted and recreated before more jobs can be submitted to it; ask the System Administrator to do this. Meanwhile, use any other available queues.

- Register setting.

(Fatal) Register settings are illegal in the Batch subsystem, except as part of a submitted command file. Reenter the command line without the register setting.

- Removed <queue-name> from BATDEF

(Message) This message is sent to the supervisor terminal when the Batch monitor finds, in the BATDEF file, a queue that is flagged for deletion but that has never been used. The message indicates that the queue is deleted from BATDEF, and no job data was lost as a result.

- Searching for free command file, please stand by.

(Response) This and other messages such as "Queue is in heavy use...please stand by" mean that many users are submitting command files at once. The situation should resolve itself in a short amount of time.

- Someone invalidated the database.

A user has damaged the data base. (This message usually follows other messages.) After this message has been sent, the Batch monitor logs itself out. When the monitor has logged out, the operator should run FIXBAT to repair the data base.

- Specified value is out of range.

(Fatal) The -CPTIME or -ETIME option specified during job submission or a -CHANGE operation is greater than the maximum allowed by the queue to which the job was submitted. This message will be preceded by a message indicating the maximum limit for that queue ("Cpu limit is xx" or "Elapsed time limit is xx"). If the limits cannot be lowered so that the job can be successfully run, try a queue with higher limits.

- Stop request issued.

(Response) The BATCH -STOP command has resulted in a request that the Batch monitor stop. Within 20 seconds the monitor should send an "Operator Stop." message to the supervisor terminal and log out.

- Syntax error. Register settings are illegal

(Warning) This message is output if an end-of-line is expected and a register setting is found instead. Reenter the command without register settings.

- This job cannot be restarted.

(Response) This message is output by a JOB -DISPLAY command if the job being displayed cannot be restarted. A job is not restartable if a JOB -CANCEL command is issued for that job while it is executing, or if it is submitted with the -RESTART NO option. Any attempted restarts will abort the job without restarting it.

- (This job has already executed nn time(s).)

(Response) This message is output by a JOB -DISPLAY command if the job being displayed is executing and has already been executed. This is the result of a JOB -RESTART, or a system coldstart after shutdown while the job was executing.

- This job will be restarted.

(Response) This message is displayed in response to a JOB -DISPLAY command if a JOB -RESTART was done but the job is still executing. When the monitor sees that the job has aborted or completed, it will return the job to the "Waiting" state.

- Too few system units configured. Change FILUNT directive in CONFIG.

(Fatal) The FILUNT directive in the CONFIG file allows less than 16 units per user. The Batch subsystem cannot work unless the COMOUTPUT unit is at least 17 (decimal). The operator or System Administrator must change the FILUNT directive in the CONFIG file to allow at least 16 units per user.

- Too many options.

(Fatal) At least two conflicting options were entered, such as JOB -DISPLAY -CHANGE or JOB C_TEST -ABORT -CANCEL. Use separate JOB commands to perform separate operations.

- Too many queues.

(Warning) An attempt was made, using the ADD command in BATGEN, to add a queue when there were already 16 defined queues (blocked, unblocked, or flagged for deletion).

- Unknown command.

(Warning) An unrecognized command was entered while in BATGEN command mode. The user will be left in BATGEN command mode and the erroneous line will be thrown away.

- Unknown option.

(Fatal) An option to the BATCH or JOB command was entered but was not recognized.

- Unknown queue name.

(Warning) A command entered while in BATGEN command mode referred to a queue that either did not exist or was "flagged for deletion" by the DELETE command.

- Unknown subcommand.

(Warning) While in BATGEN subcommand mode, an unrecognized subcommand was given. The user will be left in subcommand mode.

- Unrecognized option.

(Fatal) BATGEN was invoked with an unrecognized option on the command line. The only legal options are -STATUS and -DISPLAY.

- Warning: jobs are not being processed at this time.

(Response) This message means that the Batch monitor is not running, so any submitted jobs will not be executed until it is started up. Then the requested operation will be performed. Note that if the system is shut down without the monitor logging itself out, a data base problem may result. In such cases, the messages "*** Jobs are not being processed at this time", "*** Please try again later" will be output instead, and the requested operation will not be performed.

FIXBAT MESSAGES

- Another user may be running FIXBAT.

FIXBAT has encountered a "File in use" error while trying to open BATCHQ>OTHER>VALID. Usually, this means that two people are trying to run FIXBAT simultaneously.

- Can't process batch jobs from system console. (FIXBAT)

Two cases may be a problem. The operator or System Administrator has either tried to run FIXBAT with the -STARTUP option interactively from the supervisor terminal or has tried to run FIXBAT as a phantom logged in as SYSTEM. FIXBAT with the -STARTUP option must be run as a phantom. The phantom must be started by issuing the BATCH -START command from the supervisor terminal.

- Deleted filename.

This message means that FIXBAT found a temporary (T\$xxxx) file, an inactive command file (Cqnnnn), or a queue file (QCTRqp) in which entries were all past the -DAYS argument. FIXBAT deleted the file.

- Execute data not found (reinitialize). jobid

A job with execution status in the queue file had no corresponding entry in EXECUT. This can occur if the system is shut down at a time between updating EXECUT and updating the queue file. Run BATCHQ>INIT to reinitialize.

- Execute/data username mismatch (reinitialize). jobid

FIXBAT has found a job that is supposedly executing, but the corresponding job-id in the EXECUT file is owned by a different user. Run BATCHQ>INIT to reinitialize.

- FIXBAT finished.

The process of fixing the Batch data base has been successfully completed. FIXBAT will now exit to PRIMOS.

- Fixing database.

This message is output when FIXBAT decides to fix the entire BATCHQ data base.

- <filename> leftover words=n

The indicated queue file had n words at the end of it, not enough for a full queue entry. This is not a fatal error. The queue file will be truncated. The message could indicate that a process submitting a job was force-logged out in the midst of creating a new queue entry.

- IN.USE open by monitor.

FIXBAT was run without the -STARTUP option while the Batch monitor was still running.

- Redundant execute entry (reinitialize). jobid

FIXBAT found an executing job with more than one entry in the EXECUT file, an impossible condition. Run BATCHQ>INTT to reinitialize the data base.

- Unknown -STARTUP argument.

The argument supplied to the -STARTUP option is not SAVE, DELETE, SPOOL, or NOLOG.

H

PRINT_SYSLOG Messages

INTRODUCTION

This appendix contains all the messages that may appear in the system event log output file, `LOGLST`. This file is generated by the `PRINT_SYSLOG` command, which generates the output by reading a system event log file named `LOGREC*\LOG.mm/dd/yy`.

For a discussion of the `PRINT_SYSLOG` command, see Chapter 5.

PRINT_SYSLOG EVENT MESSAGES

- BAD ENTRY: xxxxxx ... (OCT).

An entry of unrecognized type or of a length longer than 81 words was encountered. An octal dump of the entry is provided for the number of words contained in the length field.

- COLD START [[PRIMOS REV rr] CPU TYPE = type
MICROCODE REV = mm ID= iiii (OCT)]

A cold start of PRIMOS was performed. rr is the PRIMOS rev number. mm is the revision of the microcode running. If PRIMOS is revision 16.2 or later, the cold start entry contains eight words of information (iiii) obtained from the STPM (Store Processor Model number) instruction. (See the PMA Programmer's Guide.)

- DISK jj ERROR, DEVICE NUMBER = pdn (typecode).
CRA= aaaaaa aaaaaa (OCT), CYLINDER= xxx, HEAD= xx
RECORD= xx RCRA= xxxxxx xxxxxx (OCT).
STATUS (ACCUM)= ssssss (OCT) STATUS (LAST) = 11111 (OCT)
RETRIES= nn yy

A disk read or write error occurred during the indicated operation.

jj indicates READ or WRITE. pdn indicates the physical device number. typecode indicates controller number and device type (MHD = Moving Head Disk, FHD = Fixed Head Disk, SM = Storage Module). aaaaaa aaaaaa shows the error's record address. The cylinder, head, and record addresses are given in decimal. xxxxxx xxxxxx is the current record address read on a CRA error and is given for read operations only.

ssssss is the Boolean OR of all status bits obtained during retries. 11111 is the status of the last operation. nn is the number of retries attempted.

yy is "(RECOVERED)" if the operation was completed successfully. yy is "(UNCORRECTABLE)" if the error could not be corrected. If the error has been successfully corrected by the software, yy is "WORD NUMBER = wwwwww (OCT), CORRECTION = cccccc (OCT)". wwwwww indicates the record's corrected word number. cccccc indicates the 32-bit correction pattern.

- DISK MOUNT: packname ON pdn (OCT)

An ADDISK or STARTUP command was issued. packname was mounted on the disk identified as pdn.

- LOGBUF OVERFLOW — xxxxxx ENTRIES LOST

Indicates xxxxxx (decimal) event entries were lost due to overflow of the event logging buffer (LOGBUF).

- MACHINE CHECK MODE NOW QUIET

PRIMS entered quiet machine check mode. This occurs after 1024 ECC errors and causes subsequent ECC errors to go unreported.

- MACHINE CHECK ccc... DSWSTAT= ssssss ssssss DSWRMA= yyyyy rrrrrr rrrrrr (OCT). DSWPB= pppppp pppppp (OCT).
[DSWPARTY xxxxxx xxxxxx (OCT)]

A machine check occurred. DSWSTAT, DSWRMA, DSWPB, and DSWPARTY give information regarding the DSW (descriptor segment word) at the time of the check.

DSWPARITY is displayed for Prime 750, 850, and 9950 systems only. DSWPARITY identifies the reporting board and signal name.

If DSWPARITY is not present, ccc is an encoding of the machine check code. If the RMA INVALID bit is set (bit 9 of DSWSTATL), yyyyy is (INV). Otherwise, yyyyy is absent.

- MACHINE CHECK: USER= nn, PROGRAM COUNTER = pppppp (OCT).

This is the format of a machine check message on a Prime 300. nn is the user number (in decimal). pppppp is the user's program counter at the time of the check.

- MISSING MEMORY

A Prime 300 missing memory check occurred.

- MISSING MEMORY. DSWSTAT = ssssss ssssss (OCT). DSWRMA = yyyyyy rrrrrr rrrrrr (OCT). DSWPB = pppppp pppppp (OCT). [DSWPARITY = xxxxxx xxxxxx (OCT)]

A missing memory module check occurred. DSWSTAT, DSWRMA, DSWPB, and DSWPARITY give information regarding the DSW (descriptor segment word) at the time of the check.

DSWPARITY is displayed for P750 and P850 only.

If the RMA INVALID bit is set (bit 9 of DSWSTATL), yyyyy is (INV). Otherwise, yyyyy is absent.

- MEMORY PARITY (www) DSWSTAT= ssssss ssssss (OCT). DSWRMA= yyyyyy rrrrrr rrrrrr (OCT). DSWPB= pppppp pppppp (OCT). [DSWPARITY= xxxxxx xxxxxx (OCT)]
PHYSICAL PAGE NUMBER = nnnnnn (OCT), WORD NUMBER = wwwwww (OCT).
REAL ADDRESS = aaaaaaaa (OCT). [CACHE ADDRESS = cccccc (OCT)]

A memory parity error occurred. www is either ECC CORRECTED or ECC UNCORRECTED. DSWPARITY is displayed but not decoded.

For a corrected error, nnnnnn is followed by "Bit = y", where y indicates the bit in error (1-16 = bits 1-16; LP = left parity; RP = right parity; C2, C4, C5 = other check bits; MB = multibit; NE = no error). This is followed by OVERALL PARITY = x, where x is 0 or 1, and reflects the setting of DSWSTATL bit 6.

- POWER FAIL CHECK

A power fail check occurred.

- PRIORITY ACL SET ON DISK pdn BY USER nn (USER name)

A priority ACL was set on disk partition name pdn. nn is the decimal number of the user who set the ACL; name is the name of that user.

- "Text of Operator remark"

Contents of the REMARK event, generated by use of the -REMARK option to PRINT_SYSLOG.

- SHUTDOWN BY OPERATOR

The operator issued a SHUTDOWN ALL command. (This event causes the event log buffer to be automatically dumped.)

- SHUTDOWN DUE TO CHECK FROM SENSORS. REGISTER 26: stat1 stat2 (OCT)

A sensor check occurred causing the loading of register 26 with the sensor status data (stat1 stat2) and the shutting down of the system. Possible causes for this are:

CABINET TEMPERATURE TOO HIGH
LOW AIR FLOW
UPS BATTERY LOW

- TYPE = tt. DATA= dddddd ... (OCT).

An input file entry of types 10-15 was encountered. tt is the type of entry; dddddd ... is a display of up to nine words of information from the entry.

- WARM START

A warm start of PRIMOS was performed.

I

PRINT_NETLOG

Messages

INTRODUCTION

This appendix contains all the messages that may appear in the network event log output file, NETLST. This file is generated by the PRINT_NETLOG command, which generates the output by reading a network event log file named PRIMENET*>NET_LOG.mm/dd/yy.

For a discussion of the PRINT_NETLOG command, see Chapter 5.

PRINT_NETLOG EVENT MESSAGES

- BAD ENTRY: xxxxxx ... (OCT).

An entry of unrecognized type or illegal length was encountered. An octal dump of the entry is provided for the number of words contained in the length field.

- CIRCUIT RESET - a ORIGINATED - controller: xx
[CIRCUIT STATE: c (OCT) CAUSE: s DIAGNOSTIC: ddddd (OCT)]

A virtual circuit was reset. a indicates whether the origin of the reset was local or remote. xx indicates physical line or node. CIRCUIT STATE, CAUSE, and DIAGNOSTIC are only printed if a is REMOTE. s may be: DTE RESET, OUT OF ORDER, REMOTE PROCEDURE ERROR, NETWORK CONGESTION, or a word of octal data.

19.3

- CMDR SENT FOR LOGICAL LINE xxxxxxxx xxxxxxxx xxxxxxxx

A rejected command was sent. The xx strings represent the display (in binary) of the cause.

- COLD START [- PRIMOS REV rr ...]

A cold start of PRIMOS was performed. The PRIMOS revision number is indicated for Rev. 17.3 and later.

- *** END OF filename -- nnnnn ENTRIES, ppppp PROCESSED ***

This message is printed when NETLOG reaches the end of the input file. nnnnn gives the decimal number of entries in the input file not including date/time and NETBUF overflow entries. ppppp gives the number of entries processed.

19.3

- ICS1.1 (X.25) DECONFIGURE CODE WORD NOT QUEUED FOR LOGICAL LINE
- ICS1.2 (X.25) LOGICAL CONNECTION DELETED FOR LOGICAL LINE
- ICS1.3 (X.25) LOGICAL CONNECTION NOT BROKEN FOR LOGICAL LINE
- ICS1.4 (X.25) LCAD1_ NOT FOUND IN LCB FOR LOGICAL LINE
- ICS1.5 (X.25) LOGICAL CONNECTION LOST FOR LOGICAL LINE
- ICS1.6 (X.25) FLUSH TIMEOUT FOR LOGICAL LINE
- ICS1.7 (X.25) ILLEGAL FLUSH COMPLETE FOR LOGICAL LINE
- ICS1.8 (X.25) SYNCHRONOUS LINE NOT ASSIGNED FOR LOGICAL LINE
- ICS1.9 (X.25) UNIDENTIFIABLE ERROR FOR LOGICAL LINE
- ICS1.0 (X.25) LINE NOT DEFINED:

Indicates problems with the Intelligent Communications Subsystem, Model I (ICS1) PRIMENET X.25.

- INCOMING CALL REQUEST

An incoming call request was received.

19.3

- INTERNAL LEVEL 2 ERROR FOR LOGICAL LINE
ERROR CODE = xx LINE CONTROL BLOCK ADR = yy

An internal error occurred at level 2. xx is the error code that was generated. yy is the address of the line control block.

- LEVEL III PROTOCOL DOWN controller: xx (EVENT OCCURRED n TIMES)

The Level III protocol for X.25 is down for this host. xx indicates physical line or node. The message in parentheses will only display if there were multiple occurrences.

- Level3 network received a diagnostic packet p
The controller number is: xx The line number is: yy q

A diagnostic packet has been received. p may be either "This packet was sent by a DTE" or "DNIC of the PDN is: nnn". nnn is the Data Network Identification Code. xx is the number of the controller, yy is the line number. q will be one of the following:

- No additional info, dcode is: xx
- Packet not allowed, dcode is: xx
- Packet on an unassigned Lchannel
- Packet too short, dcode is: xx
- Invalid GFI, dcode is: xx
- Timer expired, dcode is: xx
- Timer expired for CLEAR INDICATION
- Timer expired for RESET INDICATION
- Timer expired for RESTART INDICATION
- The diagnostic code is: xx

19.3

In any of these, xx is an octal string representing an event in the network buffer.

- LOCAL PROCEDURAL ERROR CAUSING CLEAR
(X.25) controller: xx

A local procedural error caused the clearing of a circuit in this host. xx indicates physical line or node.

- NETBUF OVERFLOW — nnnnn ENTRIES LOST

This indicates that nnnnn (decimal) event entries were lost due to overflow of the network event logging buffer (NETBUF).

- NETDMP CALLED AT: xxxxxx xxxxxx (OCT).
[DATA: yyyyyy yyyyyy yyyyyy (OCT)]

A network software problem has occurred at this address. The routine NETDMP was called and asked to dump the three DATA words.

- NPX>R\$CALL>R\$CONN UNKNOWN CIRCUIT STATUS - NODE: xxxxxx (OCT).
VIRTUAL CIRCUIT STATE (1): xxxxxx (OCT).
VIRTUAL CIRCUIT STATE (2): xxxxxx (OCT).

PRIMENET has returned an unexpected status (error) code to NPX. This may be caused by the failure of a node in a controlled or uncontrolled way. A software failure was perhaps was caused by a hardware failure.

- NPX>R\$RLS ERROR IN VIRTUAL CIRCUIT CLEARING - NODE: xxxxxx (OCT).
VIRTUAL CIRCUIT STATE (1): xxxxxx (OCT).
VIRTUAL CIRCUIT STATE (2): xxxxxx (OCT).

There is a problem in clearing the virtual circuit (R\$RLS). The returned VC status word 2 is none of the existing XS\$ status words. The event is logged for this potential problem.

- NPX>TRNRCV MASTER'S CIRCUIT WAS CLEARED - NODE: xxxxxx (OCT).
VC STATE(1): xxxxxx (OCT). VC STATE(2): xxxxxx (OCT).

The connection between the master and the slave has been unexpectedly broken. This may be caused by the failure of a node in a controlled or uncontrolled way. A software failure was perhaps caused by a hardware failure.

- NPX>TRNRCV MESSAGE OUT OF SEQUENCE IN BOUNCE DETECT.
NODE: xxxxxx (OCT). MESSAGE SEQ#: xxxxxxxx (OCT), NS: xxxxxxxx (OCT).

NPX break detection/correction logic found message out of sequence. NPX has failed, or data has been lost in the network. This may be caused by the failure of a node in a controlled or uncontrolled way. A software failure was perhaps caused by a hardware failure.

- NPX>TRNRCV THROTTLED ON TRANSMIT OR RECEIVE -
NODE: xxxxxx (OCT), MASTER/SLAVE FLAG: xxxxxx (OCT).

Network buffers are too full to send or receive an NPX message. This may be caused by the failure of a node in a controlled or uncontrolled way. A software failure perhaps was caused by a hardware failure.

- NPX>TRNRCV UNKNOWN RECEIVE STATUS - NODE: xxxxxx (OCT).
MASTER/SLAVE FLAG: xxxxxx (OCT). RECEIVE STATE: xxxxxx (OCT).

PRIMENET has returned an unanticipated status (error) code to NPX. May be caused by the failure of a node in a controlled or uncontrolled way. A software failure perhaps was caused by a hardware failure.

- OUTGOING CALL REQUEST

An outgoing call request was received.

- PACKET OUT OF SEQUENCE - controller: xx CIRCUIT STATE: c
SEQ # EXPECTED: d SEQ # FOUND: e

A packet was received with an unexpected sequence number.

- PNC HARDWARE FAILURE: DMA FAILURE
 - or
 - NO SKIP ON INA
 - or
 - NO SKIP ON RECEIVE OTA
 - or
 - NO SKIP ON TRANSMIT OTA

There has been an apparent failure of the Prime Node Controller (PNC) hardware that controls the ring. The device has been shut down, and this node has removed itself from the ring. Hardware diagnostic tests should be run on the PNC.

- POWER FAIL CHECK

A power failure check occurred.

- RESOURCE FAILURE RING QUEUE OVERFLOW xxxxxxxx

The software controlling the PNC queue has overflowed its buffers. Since the queues are designed to be large enough to handle peak traffic numbers, this message indicates a configuration problem. The packet being queued is returned to the free pool and ignored. xxxxxxx is the additional message indicating the result of the overflow. It may be one of the following: DIM TO LEVEL II - RECEIVE PACKET LOST, LEVEL II TO DIM TRANSMIT PACKET LOST, or LEVEL II TO DIM - TIMER PACKET LOST. This error should never occur. If it does, your Customer Service Representative should be informed.

19.3

- RING DIM OUT OF RECEIVE BLOCKS

The software controlling the Prime Node Controller (PNC) has been handling enough traffic to temporarily exhaust the available supply of buffers. If this event happens often, the system should be rebuilt with more buffers to handle this network's message load.

- RING NODE: node-number NOT ACCEPTING TRANSMITS.

There are three common reasons for this:

PACKET WACKED, NODE HALTED OR CONGESTED will probably be the most common. This will usually indicate a node that has been halted.

PACKET LOST, RING DOWN indicates that packets sent to the indicated node are not physically returned to the originating node (the system that logged the event). Most probably the ring itself is broken.

NODE NOT IN RING indicates that the node has removed itself from the ring and is not reacting to the transmitted packet.

If the reason is not one of those above, the error message will print out: TRANSMIT STATUS IS: xxxxxx (OCT). The octal contents of the transmit status word from the PNC is xxxxxx.

19.3

- Ring node node-number Receive TIMEOUT - node down

A packet has not been received from the specified node within its required time period. The node is marked as down.

- RING QUEUE OVERFLOW: DIM TO LEVEL II - RECEIVE PACKET LOST
or
LEVEL II TO DIM - TIMER PACKET LOST
or
LEVEL II TO DIM - TRANSMIT PACKET LOST

One of the queues used to move packets to and from the Prime Node Controller Device Interface Module (PNCDIM) has overflowed. Since the queues are designed large enough to handle peak traffic numbers, this message indicates some kind of configuration problem. The packet being queued is returned to the free pool and ignored.

- SHUTDOWN BY OPERATOR

The operator issued a SHUTDN ALL command. This causes the network event log buffer to be automatically dumped.

- SMLC - NO STX PRECEEDING ETX. PHYSICAL LINE NUMBER = xxxxxx (OCT),
DEVICE ADDRESS IS yyyyyy (OCT)

SMLC packets must begin with DLE/STX and end with DLE/ETX.

- SMLC RESET FOR LOGICAL LINE xxxxxx (OCT) - cause

Resets can be caused in six ways: INVALID ADDRESS, COMMAND REJECT, INVALID NR, INVALID RESPONSE, INVALID NR ON REJECT, MAXIMUM NUMBER OF RETRIES EXCEEDED.

- SMLC STATUS ERROR STATUS WORD IS xxxxxx (OCT)
[PHYSICAL LINE # IS n]
DEVICE ADDRESS IS yyyyyy (OCT), [NUMBER OF OCCURRENCES IS nnn]

An invalid status, xxxxxx, has been reported by the SMLC. nnn is printed only on parity errors.

- SMLC5 - CMDR SENT FOR LOGICAL LINE xx yyy

Notification sent to logical line xx that a command was rejected at level 2. yyy is the octal representation of the problem.

19.3

- SMLC6 - INTERNAL LEVEL 2 ERROR FOR LOGICAL LINE xx

An internal error has occurred on line xx.

- SPURIOUS RECEIVE INTERRUPT ON PNC

A receive interrupt was issued by the Prime Node Controller (PNC) when there was no receive pending. This is an indication of a hardware malfunction. The PNC is disconnected from the ring. Hardware diagnostic tests should be run on the PNC.

- SYSTEM BLOCKS UNAVAILABLE FOR SMLC PROTOCOL MESSAGE
MESSAGE IS xxxxxx (OCT), LOGICAL LINE NUMBER IS yyyyyy (OCT)

The level II synchronous protocol had no buffers in which to send the indicated type of protocol-generated message.

- Ring node_ xx Receive TIMEOUT - node down

The specified node is down and the request has timed out.

19.3

- "Text of operator remark"

Contents of the REMARK event, generated by use of the -REMARK option of PRINT_NETLOG.

- TOKEN INSERTED INTO THE RING NETWORK

The software controlling the PNC hardware issued a ring network control token.

- WARM START

A warm start of PRIMOS was performed.

J

Disk Errors

INTRODUCTION

This appendix describes the meaning of disk error messages sent to the supervisor terminal. The format and meaning of these messages depend on the type of disk controller that generated the error condition.

There are two general classes of disk controllers:

- Controllers for fully-supported disks
- Controllers for floppy disks

All record, device, and status numbers are octal.

FULLY SUPPORTED DISKS

Following is a discussion of error detection and correction on Storage Module Disks (SMD), Cartridge Module Devices (CMD), and Fixed Media Disks (FMD).

Error Detection

Under PRIMOS, disk read/write errors on fully-supported disks generate the error message:

DISK xx ER phys-dev rec-num act-rec status retries

xx RD for a read error and WT for a write error.

phys-dev The physical device number of the module or partition on which the error occurred.

rec-num The desired record number within the partition (two words).

act-rec The actual record number read (two words). On read operations it should agree with the rec-num. On write operations it has no validity.

status The reason for the error. May be one of the words provided in the table below, or may be the sum of two or more words.

<u>Status Word</u>	<u>Meaning</u>
177777	Bad record identifier
177776	Device not ready
100000	Always set
020000	Write protect violation
040000	DMX overrun
010000	Check error
004000	Checksum error
002000	Header check failure
000010	Disk drive seeking
000004	Illegal seek
000002	Select error
000001	Not available or not ready

retries The number of times the read or write operation was attempted unsuccessfully before the error message was printed. Maximum is 10 ('12).

Under PRIMOS II, the error message format is:

DISK xx ERROR phys-dev rec-num act-rec status

The definitions are the same as for the PRIMOS messages. There is no retry count; PRIMOS II prints the message and tries ten times.

Error Correction

The disk controller writes a two-word correcting code checksum on each record. An error detection and correction scheme (ECC) is implemented for fully-supported disks.

The code, together with the correction logic in the controller, is capable of detecting any of the following:

1. Up to two error bursts of combined length of up to 22 bits.
2. A single error burst of up to 32 bits in length.
3. Any odd number of errors.

In addition, the ECC algorithm is capable of correcting any single error burst of up to 11 bits in length. Error correction is attempted only after ten attempts to read a record have failed.

The message printed when error correction was attempted and failed is:

UNCORRECTABLE.

The message printed when error correction succeeds is:

wordno error error

wordno The offset at the beginning of the correction
relative to the beginning of the record.

error error The 32-bit correction pattern (two 16-bit words,
of which no more than 11 consecutive bits will
be nonzero).

Following is an example of a disk error message:

```
DISK RD ER 020063 000000 016357 000000 015477 100014 000012
UNCORRECTABLE.
DISK RD ER 020063 000000 016360 000000 016355 100014 000012
UNCORRECTABLE.
```

FLOPPY DISKS

Disk error messages for floppy disks are printed in the same format as error messages for fully-supported disks. However, the meaning of status-word is different (see section on status word below). The error messages for read and write errors are as follows:

```
DISK RD ERROR device-number record-address cra status-word
DISK WT ERROR device-number record-address cra status-word
```

On read request errors cra (two words) is the actual record number read and should match the requested record record-address (two words). cra has no validity on a write request.

It is not possible, in a program, to trap a detected disk error. Under PRIMOS II a message is printed and the operation is continually retried. Under PRIMOS the operation is tried ten times.

Status Word

For a floppy disk controller, the meaning of the status word, typed as the rightmost octal number of a disk error message, is as follows:

<u>Status Word</u>	<u>Meaning</u>
177777	Bad record identifier
177776	Device not ready
100000	Normal end of instruction (good if present)
040000	Sector not found
020000	Checksum error on sector ID
010000	Track error; head is mispositioned
002000	Deleted data mark read
001000	DMX overrun
000400	Checksum error, write protect violation of file inoperable on write or format

K

FAM I

INTRODUCTION

At Rev. 19.3 FAM I is no longer supported. This appendix is included for historical reference only.

FAM I is Prime's older File Access Manager (FAM). At Rev. 19.3, it has been replaced by RFA (previously known as FAM II). This appendix describes the differences between using FAM I and RFA. The procedures for using RFA are described throughout this book.

19.3

STARTING UP FAM

FAM I must be started or restarted as a phantom from the supervisor terminal. Its phantom command file is called PH_FAM, which must be located in the UFD FAM. To start the phantom, type:

OK, PH FAM>PH_FAM

SPECIAL FAM I COMMAND LINES

The FAM I command line format for the ADDISK command is different from the RFA format. An additional command, REMOTE, is required to set FAM I access to local disks.

19.3

Adding Remote Disks (ADDISK)

When FAM I is used between two systems, the following steps must be performed in the indicated order:

1. The operator on the remote system must use ADDISK to start up the disk on that system.
2. The operator on the remote system must use the REMOTE command to make the disk accessible to your node.
3. You must start up the disk on your system using ADDISK. The format of the ADDISK command is:

ADDISK nodename pdev-1 [pdev-2 ...pdev-8]

where nodename is the name of the system on which the physical devices are located, and pdev-n are the physical device numbers of the remote disks being added. For example:

OK, ADDISK FIELD 464 101060

The ADDISK command checks to see whether the remote link and system are up and if the disk being added actually exists. If either of these conditions is not true, the command fails and the message "illegal name" is displayed.

Note

ADDISK will fail with FAM I if the disk is not up on the remote system, or if the remote system itself is not up. For this reason, it is recommended that FAM I commands not be entered in C_PRMO.

Shutting Down Remote Disks (SHUTDN)

To shut down remote disks that your system accesses via FAM I, use the command format:

SHUTDN nodename pdev-1 [pdev-2...pdev-8]

nodename is the network name of the system on which the devices are physically mounted, and pdev-n are the physical device numbers of the remote disks to be disconnected. The command detaches all local users who are attached to the disk(s), and closes the file units they have open there. The command works only if the remote link and node are up. (If the FAM I link is broken, the disks and units will have been closed down already.)

Setting Access to Local Disks (REMOTE)

The REMOTE command is used to permit or deny access to local disks from specific network nodes or from the entire network. The format of the command is:

$$\text{REMOTE } \left\{ \begin{array}{l} \text{PERMIT} \\ \text{DENY} \end{array} \right\} [\text{option}]$$

where PERMIT and DENY permit or deny access to local disks by remote network nodes named in the options. The options are:

$$\left\{ \begin{array}{ll} \text{nodename} & \text{pdev-1 [pdev-2 ... pdev-9]} \\ \text{nodename} & \text{-ALL} \\ \text{-NET} & \text{pdev-1 [pdev-2 ... pdev-9]} \\ \text{-NET} & \text{-ALL} \end{array} \right\}$$

The following examples illustrate how REMOTE is used to permit access from remote nodes. (To deny access, the DENY keyword is used.)

```
REMOTE PERMIT nodename pdev-1 [pdev-2 ... pdev-8]
```

The operator at system nodename can use the ADDISK command for any listed devices. At least one device must be specified in this list. All local devices specified in this list must already be started up with a previous ADDISK command.

```
REMOTE PERMIT nodename -ALL
```

This command permits the operator of system nodename to use the ADDISK command to start all presently started up local disk partitions. It has no effect on local partitions added after this command is carried out.

```
REMOTE PERMIT -NET pdev-1 [pdev-2 ... pdev-8]
```

All configured network nodes can access the specified local disk partitions.

```
REMOTE PERMIT -NET -ALL
```

All network nodes can access all presently-started up disk partitions.

PERMIT and DENY affect only disk partitions already started up at the time of the REMOTE command. Disks shut down and started up again have the system default permissions until an explicit REMOTE PERMIT or REMOTE DENY command changes them. The system default permissions are determined from the file NETCON which is created by NETCFG. The REMOTE

PERMIT command does not automatically add a disk to any system. The REMOTE DENY command does not revoke a system's existing access to a disk.

You must use ADDISK to start up a disk before you can use REMOTE to give it a permit/deny status. Once you permit access to a disk by a remote node, the operator on that node may use ADDISK to start up the disk there.

The REMOTE DENY command does not revoke a system's existing access to a disk, but rather prevents the system from next starting up the disk again.

FAM I MESSAGES

If your system is running FAM I, messages of the following types will appear at the supervisor terminal:

- FAM (57) LOGGED IN AT 0'01

The FAM phantom is logged in and is running. The time is 1 minute after system startup began.

- **** 0'01 SYSD FAM <182B13 > OPERATIONAL**

FAM NODES ENABLED:

SYSA

SYSB

The local FAM program is in operation.

- **** 0'03 SYSB FAM <171F25 > INITIALIZED **

The FAM running on system SYSB has been started up. It may be of a different rev. than the FAM on the local system.

- **** 8'38 SYSB FAM <171F25 > OPERATIONAL**

The FAM running on system SYSB is in working order and is in communication with your system.

INDEX

Index

A

Access, 1-3, 2-10

Access Control Lists, 2-9
 commands to monitor, 2-9
 MAGRST, 8-22
 MAGSAV, 7-37
 symbols, 2-10
 terminology, 1-3

Access rights, 13-33, 13-66,
 13-67

Adding disks, 2-8, 13-2

Adding software, 3-7

Adding UFDs, 3-6

ADDISK, 13-2

AMLC, 13-6

ASSIGN, 4-8, 13-7

Assignable disks table, 2-7,
 13-19

Assigned, 1-4

Asynchronous,
 dropping DTR, 13-20
 line configuration, 13-6
 terminal lines, 3-2
 terminology, 1-4

Automatic sampling, 5-2

AVAIL, 2-17

B

Backing up, 7-1

Badspots,
 adding to BADSPT file, 9-8,
 13-22
 COPY_DISK, 7-30
 copying information from other
 partitions, 6-7, 13-41
 disposition of files
 containing, 9-9
 entering known locations, 6-8,
 13-41
 equivalence blocks, 9-4
 file creation, 6-2
 handling, 9-18
 level of checking for, 6-6,
 13-42

Badspots (continued)

- listing, 9-10, 13-24
- redundant BADSPT files, 6-9
- turning off, 13-19, 13-49

BADSPT file, 2-3

- adding new badspots, 9-8, 13-22
- automatic input, 6-6
- COPY_DISK, 7-30
- copying from other partitions, 6-7, 13-41
- creation, 6-2
- equivalence blocks, 9-4
- handling, 9-18
- listing, 9-10, 13-24
- manual input, 6-7, 6-8, 13-42
- not using, 13-19, 13-49
- redundant, 6-9

Batch, 4-5, 11-1

- cleaning up queues, 11-15
- commands, 4-7
- continuing the monitor, 11-6
- controlling user jobs, 11-9
- database repair, 11-12, 13-25
- error messages, G-1
- example, 11-8
- FIXBAT, 13-25
- forced logouts, 11-7
- general status, 11-2
- INIT, 11-13
- monitor, 4-6, 11-4, 13-10
- monitoring, 11-1
- monitoring queues, 11-10
- pausing the monitor, 11-6
- phantoms, 1-10
- queues, 11-9, 13-11
- removal of old jobs, 11-14
- spontaneous messages, 11-4
- starting the monitor, 11-5
- stopping the monitor, 11-7
- supervisor terminal messages, 11-4
- terminology, 4-6
- user job information, 11-3

BATCH, 11-2, 11-5, 11-6, 13-10

BATCH_SERVICE, 11-4

BATCHQ, 2-3

BATGEN,

- blocking queues, 11-10
- monitoring batch queues, 11-10
- summary and grammar, 13-11
- unblocking queues, 11-10

Baud rate,

- asynchronous lines, 13-7
- supervisor terminal, 6-7, 13-42

Blocked queues, 11-10, 13-11

BOOT file, 2-2, 6-1

BOOT_CREATE, 13-12

Broadcast messages, 3-5, 13-43

C

Cartridge Module Devices (CMD),

- COPY_DISK, 7-31
- disk errors, J-1
- drive unit numbers, A-3
- MAKE specification, 6-5, 13-40
- partitioning of, A-10
- physical device numbers, A-1

CHAP, 3-8, 13-17

CLOSE,

- under PRIMOS, 13-18
- under PRIMOS II, 14-5

CMDNCO, 2-4, 3-7, 6-2

Cold start, 14-6

COMINPUT, 7-1, 14-5

Command directory, 2-4

Command disk (COMDEV),

- emergency restore, 8-32
- repairing, 9-7

Command files, 1-8, 14-5

Communications controllers, 13-77

- Concepts, 1-3
- CONFIG, 5-12, 13-18
- CONFIG_NET, 13-18
- Converting disks, 9-10, 9-18, 13-25
- COPY, 8-2
- COPY_DISK,
 - emergency restore of command disk, 8-29
 - error messages, E-1
 - performance, 7-26
 - procedures, 7-15, 8-6
 - procedures illustrated, 7-18, 7-24
 - summary and grammar, 13-18
- CPL files, 1-8, 7-1, 14-5
- CREATE, 14-5
- D
- Damaged directory, 9-1, 13-22
- Date and time, 13-69
- DELETE, 14-6
- Directories, 2-1
- DISCS, 2-3, 2-19
- Disk conversion, 9-10, 9-18, 13-25
- Disk error messages, J-1
- Disks,
 - adding, 13-2
 - assigning, 13-7, 13-19, 13-78
 - backing up, 13-50
 - backup illustrated, 7-16, 7-21
 - controller restrictions under PRIMOS II, 14-1
 - copying, 13-18
 - creating, 13-40
 - DISCS file, 2-19
 - Disks (continued)
 - drive unit numbers, A-3
 - error messages, J-1
 - FAM I, K-1
 - formatting, 2-7, 13-40
 - name of, 6-1, 6-4, 13-40
 - partitioning of, A-8
 - physical device numbers, A-1
 - priority access, 13-34, 13-66, 13-67
 - repairing, 2-20, 9-1, 13-22, B-1
 - restore illustrated, 8-3, 8-10
 - restoring, 13-49
 - shutting down, 13-71, 14-7
 - sizes of, A-2
 - specifying, 9-7
 - starting up, 13-2, 14-7
 - unassigning, 13-78
 - use by file system, 2-1
- DISKS, 13-19
- DOS, 2-3, 6-2
- DPTCFG, 13-20
- DPTX, 13-20
- Drive unit numbers, A-3
- DROPDTR, 13-20
- DSKRAT, 2-3, 6-2, 9-18
- E
- ELIGTS, 13-21
- Emergency restore,
 - of command disk (COPY_DISK), 8-29
 - of command disk (MAGRST), 8-31
- Equivalence blocks, 9-18
- Error messages,
 - ADDISK, 13-4
 - bad DAM file, 9-3
 - bad truncate of segment directory, 9-4
 - Batch, G-1

Error messages (continued)

COPY_DISK, E-1
 damaged directory, 9-3
 device in use, 1-12
 disk format not supported, 9-4
 disk not assigned, 9-7
 disks, J-1
 file in use, 10-7
 file too long, 9-4
 FIX_DISK, F-1
 FIXBAT, G-19
 FIXRAT, B-4, B-10
 forced logout by operator,
 11-7
 MAKE, 6-12
 monitor already paused, 11-6
 monitor already started, 11-5
 monitor not paused, 11-7
 phantoms, 1-9
 pointer mismatch, 9-3
 PRINT_NETLOG, 13-54
 PRINT_SYSLOG, 13-59
 process not started, 11-7
 quota system, 9-3
 SHUTDOWN, 13-73
 too many subdirectory levels,
 9-4

Event logging, 5-12

 disabling, 13-21
 enabling, 13-21
 network event messages, I-1
 printing network events, 13-51
 printing system events, 13-56
 system event messages, H-1

EVENT_LOG, 5-13, 13-21

EVFU, 10-14

Exclusive assignment, 1-12, 13-7

F

F77>TOOLS, 2-5

FAM I, K-1

File system, 2-1

File Transfer Service (FTS),
 4-15, 12-1

 abandoning servers, 12-6
 commands, 4-16
 controlling servers, 12-4
 log files, 12-3
 managing requests, 12-4
 monitoring, 12-1
 starting the manager, 12-5
 stopping the manager, 12-6
 terminology, 4-15

Files, 2-1

FIND_RING_BREAK, 13-22

FIX_DISK, 9-1

 after running, 9-11
 before running, 9-4
 error messages, F-1
 examples, 9-12
 how it works, 9-18
 options, 9-6
 running, 9-6
 summary and grammar, 13-22

FIXBAT, 11-12

 during system startup, 11-12
 error messages, 11-15, G-19
 function, 4-6
 responses, 11-15
 summary and grammar, 13-25

Fixed-media Disks (FMD),

 disk errors, J-1
 drive unit numbers, A-3
 MAKE specification, 6-5, 13-40
 partitioning, A-12
 physical device numbers, A-1

FIXRAT, B-1

Floppy disks,

 disk errors, J-1
 drive unit numbers, A-3
 MAKE specification, 6-5, 13-40
 physical device numbers, A-1

Forced logouts,

 Batch, 11-7
 File Transfer Service, 12-5

Forced logouts (continued)
 from supervisor terminal,
 13-34
 preventing subsequent logins,
 13-43
 summary and grammar, 13-34

Formatting disks,
 disk creation, 6-1
 summary and grammar, 13-40
 virgin disks, 6-6

FORMS*, 2-4

FTGEN, 13-26

FTOP, 12-4, 13-26

FTR, 12-2, 12-4, 13-30

FTSQ*, 2-4

FUTIL, C-1

G

Groups, 3-3, 13-33

I

Illustrations,
 construction of physical disk
 number, A-9
 creating a PROP environment,
 10-11
 default PROP environment, 10-3
 disk repair using FIX_DISK,
 9-5
 emergency restore of command
 disk (COPY_DISK), 8-30
 emergency restore of command
 disk (MAGRST), 8-32
 example disk-to-disk backup
 (PRIMOS), 7-16
 example disk-to-disk backup
 (PRIMOS II), 7-21
 example disk-to-disk restore
 (PRIMOS), 8-3
 example disk-to-disk restore
 (PRIMOS II), 8-10

Illustrations (continued)
 example of storage module
 partitions, A-7
 modifying a PROP environment,
 10-12
 partial restore via COPY, 8-4
 partial restore via MAGRST,
 8-18
 partition restore via COPY_DISK
 (PRIMOS), 8-7
 partition restore via COPY_DISK
 (PRIMOS II), 8-11
 partition restore via MAGRST,
 8-14
 partition restore via PHYRST,
 8-25
 partition shutdown procedure,
 7-12
 procedure for running COPY_DISK
 (PRIMOS), 7-18
 procedure for running COPY_DISK
 (PRIMOS II), 7-24
 procedure for running MAGRST
 (PRIMOS), 7-39
 procedure for running MAGSAV
 (PRIMOS), 7-34
 procedure for running PHYRST
 (PRIMOS), 7-47
 procedure for running PHYSAV
 (PRIMOS), 7-43
 system shutdown procedure, 7-7

Incremental backups, 7-32

INIT, 4-6, 11-13

Interactive user, 1-9

J

JOB,
 controlling user jobs, 11-9
 monitoring user jobs, 11-3
 summary and grammar, 13-32

L

L.DEST, 10-6

L.DFLT, 10-6

L.FORM, 10-7
 LAC, 2-11, 13-33
 LD, 2-14
 LG, 2-10, 13-33
 LIB, 2-4, 3-7
 Limiting file size,
 on disks, 13-68
 printing, 10-13
 Line numbers, 3-2
 LIST_ACCESS, 2-11, 13-33
 LIST_GROUP, 2-10, 13-33
 LIST_PRIORITY_ACCESS, 2-11,
 13-34
 LIST_QUOTA, 2-13
 Local user, 1-9
 LOG.mm/dd/yy, 13-56
 Logical disk-to-tape backup
 (MAGSAV), 7-32
 Logical tape verification
 (MAGRST), 7-38
 Login, 1-4
 preventing, 13-43
 LOGLIST, 13-56
 Logout, 1-4
 LOGOUT, 13-34
 LOGPRT, D-1
 LOGREC, 5-12
 LOGREC*, 2-4, 5-13, 13-56
 LOOK, 13-34
 LPAC, 2-11, 13-34

LQ, 2-13

M

Magnetic tape, 4-8
 allocating drives, 4-9
 assigning, 13-8, 13-78
 backups, 7-32, 7-41
 commands, 4-8
 creating a boot tape, 13-12
 determining status, 4-13
 examples, 4-11
 PHYRST, 7-46
 releasing drives, 4-14
 replying to user requests,
 13-66
 restoring to disk, 13-35,
 13-49
 saving from disk, 13-37, 13-50
 setting assignment mode, 4-8,
 13-69
 unassigning, 13-78
 user requests, 4-10
 verification, 7-14, 7-38, 7-46
 MAGRST, 7-38, 8-13, 8-17, 8-31,
 13-35
 MAGSAV, 7-32, 13-37
 MAKE, 6-1
 after running, 6-13
 examples, 6-14
 messages, 6-12
 options, 6-3
 running, 6-3
 summary and grammar, 13-40
 Manual sampling, 5-2
 Master File Directory (MFD),
 2-2, 6-1
 MAXSCH, 13-42
 MAXUSR, 13-43
 MESSAGE, 3-5, 4-19, 13-43
 MONITOR_RING, 13-46
 Mount, 1-4

N

NET, 13-46

NET_LOG.mm/dd/yy, 13-51

NETCFG, 13-48

NETCFG (obsolete) (See
CONFIG_NET)

NETLST, 13-51

NETMAN, 1-10

NETREC, 5-12

Network, 1-5

Network event logging, 5-12
 disabling, 13-21
 enabling, 13-21
 event messages, 1-1
 monitoring, 4-19
 printing events, 13-51

Network event types, 13-52

Network server (NETMAN), 1-10

NETWORK_INFORMATION.SPOOL, 10-7

No assignment mode, 4-9, 13-70

Node, 1-5

O

Offline maintenance, 2-20

Online maintenance, 2-8

Operating system restrictions,
7-5

Operator intervention,
 mode, 4-9
 setting assignment mode, 13-69
 when required, 13-8

Operator mode, 13-70

Operator tasks, 1-1, 4-5

OPRPRI, 3-7, 13-49

P

Paging, 6-1, 6-5, 13-41

Partial restore from disk (COPY),
8-2Partial restore from tape
(MAGRST), 8-17Partition name, 2-6, 6-2, 6-4,
13-40

Partition restore,
 from disk (COPY_DISK), 8-6
 from tape (MAGRST), 8-13
 from tape (PHYRST), 8-24

Partitioning of disks, A-8

Partitions, 2-6

PASCAL>TOOLS, 2-5

Performance,
 affecting system, 13-21, 13-42
 affecting user, 13-17
 COPY_DISK, 7-26, 13-19
 monitoring, 5-1, 13-79
 PHYSAV, 7-44
 split disks, 6-6

Peripherals, 1-11

Phantoms, 1-8, 1-9

PHYRST, 7-46, 8-24, 13-49

PHYSAV, 7-41, 13-50

Physical device numbers, 2-6
 commands, 2-6
 determining, A-3
 illustration of construction,
 A-9
 use of, A-1

Physical disk-to-disk backup
(COPY_DISK), 7-15

Physical disk-to-tape backup
(PHYSAV), 7-41

Physical tape verification
(PHYRST), 7-46

PLIG>TOOLS, 2-5

Plotters, 4-5, 10-1

Port, 1-5

Preventing login, 13-43

PRIMENET, 4-17

PRIMENET*, 2-4, 5-13, 13-51

PRIMOS, 1-3
summary and grammar, 14-6

PRIMOS Commands, 1-6
ADDISK, 13-2
AMLC, 13-6
ASSIGN, 13-7
BATCH, 13-10
BATGEN, 13-11
BOOT_CREATE, 13-12
CHAP, 13-17
CLOSE, 13-18
CONFIG, 13-18
CONFIG_NET, 13-18
COPY_DISK, 13-18
DISKS, 13-19
DPTCFG, 13-20
DPTX, 13-20
DROPDTR, 13-20
ELIGTS, 13-21
EVENT_LOG, 13-21
FIND_RING_BREAK, 13-22
FIX_DISK, 13-22
FIXBAT, 11-12, 13-25
FTGEN, 13-26
FTOP, 13-26
FTR, 12-2, 13-30
JOB, 13-32
LAC, 13-33
LG, 13-33
LIST_ACCESS, 13-33
LIST_GROUP, 13-33
LIST_PRIORITY_ACCESS, 13-34

PRIMOS Commands (continued)

LOGOUT, 13-34
LOOK, 13-34
LPAC, 13-34
MAGRST, 13-35
MAGSAV, 13-37
MAKE, 13-40
MAXSCH, 13-42
MAXUSR, 13-43
MESSAGE, 13-43
MONITOR_RING, 13-46
NET, 13-46
NETCFG, 13-48
OPRPRI, 13-49
PHYRST, 13-49
PHYSAV, 13-50
PRINT_NETLOG, 13-51
PRINT_SYSLOG, 13-56
PROP, 13-60
REMOVE_PRIORITY_ACCESS, 13-66
REPLY, 13-66
RPAC, 13-66
SET_PRIORITY_ACCESS, 13-67
SET_QUOTA, 13-68
SETIME, 13-69
SETMOD, 13-69
SHARE, 13-70
SHUTDN, 13-71
SPAC, 13-67
SPOOL, 13-75
SQ, 13-68
START_NET, 13-76
STATUS, 13-76
STOP_NET, 13-78
summary and grammar, 13-1
system status, 1-7
UNASSIGN, 13-78
USAGE, 13-79
USERS, 13-80
USRASR, 13-80

PRIMOS II, 1-2
capabilities, 14-1
discussion, 14-1
restrictions, 14-2
restrictions on disk
controllers, 14-1

PRIMOS II Commands,
CLOSE, 14-5
COMINPUT, 14-5
CREATE, 14-5
DELETE, 14-6
FIXRAT, B-1

PRIMOS II Commands (continued)

FUTIL, C-1
 LOGPRT, D-1
 PRIMOS, 14-6
 SHUTDN, 14-6
 STARTUP, 14-7
 STATUS, 14-8
 summary and grammar, 14-1
 types of, 14-3

PRINT_NETLOG,
 event messages, I-1
 monitoring, 4-19
 summary and grammar, 13-51

PRINT_SYSLOG,
 event messages, H-1
 summary and grammar, 13-56

Printer environments, 4-2, 10-1
 aligning forms, 10-17
 CAD/CAM, 10-15
 continuing phantoms, 10-15
 creation, 10-10
 deletion, 10-6
 discussion, 10-2
 EVFU, 10-14
 example use, 10-8, 10-18
 limiting file size, 10-13
 mechanical problems, 10-19
 modification, 10-10
 OAS, 10-15
 PROP summary and grammar,
 13-60
 recovering after jams, 10-16
 related files, 10-5
 servicing by phantoms, 10-4
 shutting down phantoms, 10-15
 starting up phantoms, 10-15
 suspending phantoms, 10-15
 synonyms, 10-13

Printer problems, 10-19

Printers, 10-1

Priority access,
 listing, 13-34
 removing, 13-66
 setting, 13-67

PRIRUN, 2-3

Processes, 1-8

Projects, 3-3

PROP, 4-3
 aligning forms, 10-17
 CAD/CAM, 10-15
 continuing phantoms, 10-15
 creating environments, 10-10
 default environment
 illustrated, 10-3
 deleting environments, 10-6
 environment information, 10-3
 environment modification,
 10-12
 environment-related files,
 10-5
 EVFU, 10-14
 example use, 10-8, 10-18
 illustration of creation,
 10-11
 limiting file size, 10-13
 mechanical problems, 10-19
 modifying environments, 10-10
 OAS, 10-15
 options, 10-8
 purpose, 10-8
 recovering after jams, 10-16
 shutting down phantoms, 10-15
 starting a printer, 10-4
 starting up phantoms, 10-15
 summary and grammar, 13-60
 suspending phantoms, 10-15

Protocol, 13-6

PRTnnn, 10-4

Q

Q.CTRL, 10-7

Queues, 1-13
 batch, 4-6, 11-9, 13-11
 blocking, 13-11
 remote spool, 4-3
 unblocking, 13-11

Quota system, 2-13
 discussion, 3-7
 FIX_DISK summary and grammar,
 13-25
 repair, 9-1, 9-18

Quota system (continued)
 setting quotas, 13-68
 when not in use, 9-10

R

Remote disks, 4-17
 adding, 13-3
 listing, 3-9, 3-11
 shutting down, 13-73

Remote sampling, 5-3

Remote user, 1-9

REMOVE_PRIORITY_ACCESS, 13-66

Renaming disks, 13-2, 13-72

Repairing disks, 9-1
 FIX_DISK summary and grammar,
 13-22

Replacing software, 3-7

REPLY, 4-8, 13-66

Request queueing, 1-12, 1-13

Restoring files from backups,
 8-1

RFA, 4-18

RPAC, 13-66

RWLOCK, 5-15

S

Segments, 3-7, 13-70

Sending messages, 3-5, 13-43

SET_PRIORITY_ACCESS, 13-67

SET_QUOTA, 13-68

SETIME, 13-69

SETMOD, 4-8, 13-69

Setting date and time, 13-69

Setting quotas, 3-7, 13-68

SHARE, 13-70

Sharing segments, 3-7
 example, 3-7
 SHARE summary and grammar,
 13-70

Sharing subsystems (See Sharing
 segments)

SHUTDOWN,
 under PRIMOS, 13-71
 under PRIMOS II, 14-6

Shutting down,
 disks, 2-8, 13-71
 partition procedure
 illustrated, 7-12
 PRIMOS, 13-71
 PRIMOS II, 14-6

SIZE, 2-16

Sizes of disks, A-2

Slaves, 1-10

SMLC, 13-47

SPAC, 13-67

Split disks, 6-1, 6-5, 13-41

SPOOL, 4-2, 13-75

Spooler, 4-1, 10-1

Spooler phantoms, 4-2, 10-4

SPOOLQ, 2-3, 10-4

SQ, 13-68

START_NET, 13-76

Starting PRIMOS, 14-6

Starting up disks, 13-2

- STARTUP, 14-7
- STATUS,
 - example, 3-8
 - under PRIMOS, 13-76
 - under PRIMOS II, 14-8
 - use with magnetic tapes, 4-8
- STOP_NET, 13-78
- Storage Module Disks (SMD),
 - disk errors, J-1
 - drive unit numbers, A-3
 - MAKE specification, 6-5, 13-40
 - partitioning of, A-9
 - physical device numbers, A-1
- Subdirectories, 2-2
- Subsystem, 1-5
- Subsystem phantoms, 1-10
- Supervisor terminal, 5-14
 - as a video display unit, 5-14
 - as user process, 13-80
 - commands, 13-1
 - messages, 5-15
 - replying to magtape requests, 13-66
 - safeguard, 13-49
 - sending messages, 13-43
 - setting magtape assignment mode, 13-69
 - terminology, 1-5
 - use under PRIMOS, 13-1
 - use under PRIMOS II, 14-1
 - user requests, 3-5
- Synchronous,
 - NET summary and grammar, 13-46
 - terminology, 1-6
- Synonyms, 10-13
- SYSCOM, 2-5
- SYSOVL, 2-5
- System, 1-6
- SYSTEM, 2-3, 2-19
- System event logging, 5-12
 - disabling, 13-21
 - enabling, 13-21
 - event messages, H-1
 - printing events, 13-56
- System event types, 13-57
- System monitoring, 5-1
 - number of users, 3-12, 13-80
 - performance, 5-1, 5-10, 13-79
 - system status, 13-76
 - system UFDs, 2-5
 - user status, 3-8
- System preparation, 7-2
 - choosing the procedure, 7-3
 - partition shutdown procedure, 7-10
 - system shutdown procedure, 7-6
 - tape verification, 7-14
- System resources, 4-1
 - possible bottlenecks, 5-10
- System shutdown,
 - PRIMOS, 13-71
 - PRIMOS II, 14-6
 - procedure illustrated, 7-7
- System startup, 14-6
- System status, 13-76
- T
- Tape verification, 7-14
 - MAGRST, 7-38
 - PHYRST, 7-46
- Terminals,
 - configuring, 13-6
 - dropping DTR, 13-20
 - sending messages, 13-43
 - supervisor as user, 13-80
- Terminology, 1-3
- Time of day, 13-69
- Time-sharing, 1-3

Types of disks, A-1

U

UNASSIGN, 4-8, 13-78

Unblocked queues, 11-10, 13-11

USAGE, 5-1
 interpreting reports, 5-10
 summary and grammar, 13-79

User community, 3-1

User complaints,
 batch error messages, G-1
 can't set quota, 13-68
 cannot log in, 3-5, 13-43
 device hogging, 1-12
 disk full, 2-17, 3-5
 DISK UNREC ERR, J-1
 error messages, 2-8, 9-3
 file in use, 10-7
 file transfer requests on hold,
 12-2
 files aren't being printed,
 4-4, 10-2, 10-13
 illegal name, 10-6
 insufficient access, 3-3
 network problems, 13-22
 no room in spool queue, 10-7
 problems with files, 9-2
 project id, 3-3
 queue full, Batch, 11-15
 system appears to hang, 13-18
 system performance, 5-10

User File Directories (UFD), 2-2

User mode, 4-9, 13-69

User numbers, 3-2

User phantoms, 1-9

User priority, 3-8, 13-17

User profiles, 3-4

User requests, 3-5
 magnetic tape, 4-10
 printing files, 4-4
 tape assignment, 13-66, 13-69

User status, 3-8, 13-77

User terminals, 1-6
 configuring, 13-6
 dropping DTR, 13-20
 sending messages, 13-43
 supervisor as user, 13-80

User time slice,
 eligibility, 13-21
 setting, 3-8, 13-17

Users, 3-1

USERS, 3-12, 13-80

USRASR, 13-80

V

Verification,
 COPY_DISK, 13-19
 magnetic tape, 7-14
 MAGRST, 7-38
 PHYRST, 7-46

Virtual Control Panel, 1-5

Volume name, 6-2, 6-4

W

Winchester disks (See
 Fixed-media Disks)

Write-protection, 13-2

Y

YTSMAN, 12-5

SURVEYS

READER RESPONSE FORM

DOC7324-192 System Operator's Guide Volume II

Your feedback will help us continue to improve the quality, accuracy, and organization of our user publications.

1. How do you rate the document for overall usefulness?

☐ excellent ☐ very good ☐ good ☐ fair ☐ poor

2. Please rate the document in the following areas:

Readability: ☐ hard to understand ☐ average ☐ very clear

Technical level: ☐ too simple ☐ about right ☐ too technical

Technical accuracy: ☐ poor ☐ average ☐ very good

Examples: ☐ too many ☐ about right ☐ too few

Illustrations: ☐ too many ☐ about right ☐ too few

3. What features did you find most useful? _____

4. What faults or errors gave you problems? _____

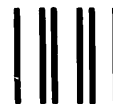
Would you like to be on a mailing list for Prime's current documentation catalog and ordering information? ☐ yes ☐ no

Name: _____ Position: _____

Company: _____

Address: _____

_____ Zip: _____



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

First Class Permit #531 Natick, Massachusetts 01760

BUSINESS REPLY MAIL

Postage will be paid by:

PRIME

Attention: Technical Publications
Bldg 10B
Prime Park, Natick, Ma. 01760

