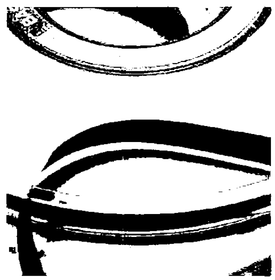
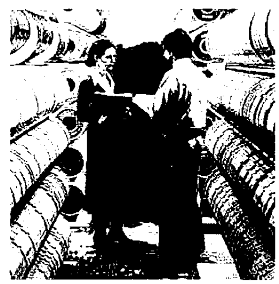
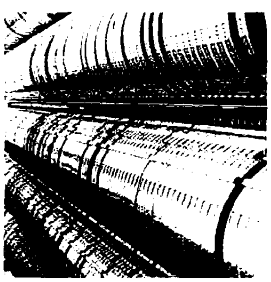
A5L

Prime Computer, Inc.

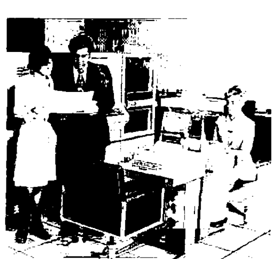
DOC5038-190P
System Operator's Guide
Revision 19.0





System Operator's Guide

DOC5038-190

First Edition

by

Michael J. Karp

This guide documents the software operation of the Prime Computer and its supporting systems and utilities as implemented at Master Disk Revision Level 19.0 (Rev. 19.0).

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

COPYRIGHT 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, 2054

Customers Outside U.S.

Contact your local Prime subsidiary or distributor.

Prime Employees

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

PRIME INFORMATION

Contact your PRIME INFORMATION dealer.

PRINTING HISTORY — SYSTEM OPERATOR'S GUIDE

Edition	Date	Number	Documents Rev.
First Edition	July, 1982	DOC5038-190	19.0

SUGGESTION BOX

All correspondence on suggested changes to this document should be directed to:

Michael J. Karp
Technical Publications Department
Prime Computer, Inc.
500 Old Connecticut Path
Framingham, Massachusetts 01701

Radio Frequency Regulations

The Federal Communications Commission (FCC) has issued regulations requiring all data processing manufacturers to test and verify that their equipment operates within the regulation limits. Prime is currently certifying its equipment, however, it is necessary to add the following information in this user document:

> WARNING
>
> This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause 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 measure may be required to correct the interference.

If there are any questions please contact your Prime Field Office.

Contents

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 contains information on the following topics:

- PRIMOS, and the file system (Chapter 1)
- Prime's hardware (Chapter 2)
- Monitoring of hardware and computer room conditions (Chapter 3)
- System startup and shutdown (Chapter 4)
- Monitoring system and subsystem resources (Chapter 5)
- Disk formatting (Chapter 6)
- Performing backups (Chapter 7)
- Repairing disk partitions (Chapter 8)
- Working with line printers (Chapter 9)
- Working with magnetic tape (Chapter 10)
- Monitoring the Batch subsystem (Chapter 11)

* Data communications (Chapter 12)
* System halts (Chapter 13)
* PRIMOS II (Chapter 14)
* Operator commands (Chapter 15)

Reference material on topics such as subsystem messages is located in the appendixes.

OTHER PRIME DOCUMENTATION YOU SHOULD KNOW ABOUT

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

Frequent reference is made to the System Administrator's Guide, DOC5037-190. This book contains information about system building, resource allocation, and system security.

Two basic references necessary to any user of a Prime system are the New User's Guide to EDITOR and RUNOFF, FDR3104, 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 Rev. 19 update package, UPD5027-190, provide a complete description of working with Prime's magnetic tape software.

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

The PRIMENET Guide, DOC3710-190, explains PRIMENET, Prime's networking system.

If you are interested in writing programs to help run your system, the CPL User's Guide, DOC4302-190, will be of primary interest to you. This book describes Prime's Command Procedure Language (CPL), a powerful and flexible tool for program development. Additionally, see the Subroutines Reference Guide, DOC3621-190, and the LOAD and SEG Reference Guide, PDR3524.

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.

Convention	Explanation	Example
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.	LOGOUT { SET_QUOTA SQ }
underlining in examples	In examples, user input is underlined but system prompts and output are not.	OK, RESUME MY_PROG 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

1 Working with PRIMOS

INTRODUCTION

This chapter provides a brief introduction to Prime's operating system, a short discussion of those directories with which the operator should be familiar, and a discussion of Prime's file structure.

PRIMOS II

PRIMOS II is Prime's single-user operating system. Because of its limited functionality as an offline environment for PRIMOS, it is not ordinarily used. For a discussion of PRIMOS II, see Chapter 14.

PRIMOS

PRIMOS is the time-sharing operating system of the 50 Series computers. 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 terminal access and phantom user noninteractive jobs
- Communications systems
- System utilities
- Database management

The majority of the operator's work (and the entirety of users' work) will be done in PRIMOS.

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 the user can define and express in 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 containing a list of subdirectories, a list of filenames, and information regarding the files. Directories are nested, one beneath the other, to form a tree structure.

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.

Many of these UFDs and files are used to support various user requests and processes, while others are used to start up the system. Support UFDs provide for system versatility and have little or nothing to do with system startup. UFDs involved in system startup, having performed their vital tasks during the startup process, continue to play an active role after the system is up and running.

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 almost all 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 of either 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.

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 Chapter 4 and Appendix A.

The BADSPT file: This file contains a listing of all bad records (physical defects such as scratches, or locations on the disk surface with little or no coating) on the partition, and appears only on partitions that have badspots. This file is searched any time a disk is copied for the purpose of identifying unusable disk records. For more information, see Chapter 6, Chapter 7, Chapter 8, and Chapter 14.

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 8 and Chapter 14.

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 like COBOL and FORTRAN. Also in this directory is the DISCS file (see Chapter 5).

The Directory PRIRUN: The UFD PRIRUN contains load maps and the PRIMOS run files, those 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 9.)

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.

Additional system directories are described in Chapter 5.

2
Hardware Overview

INTRODUCTION

All Prime computer systems consist of at least the following components: a central processing unit (called the CPU), physical memory, storage memory (disk and/or magnetic tape), and a supervisor terminal.

A more fully equipped system might also include:

- Up to 128 terminals (including the supervisor terminal)
- Additional memory
- Additional disk drives (to a maximum of 8)
- Additional magnetic tape drives (to a maximum of 8)
- Diskette drives (to a maximum of 8)
- 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)

This chapter will introduce you to most of the equipment with which you will be working on a day-to-day basis. For an in-depth view of a specific piece of hardware, see the appropriate hardware manual.

THE CENTRAL PROCESSING UNIT

The P250-II, P550-II, and P750

The CPU is the heart of any computer system. On the orange "eyebrow" panel at the top of the cabinet is a lock, four buttons, and four indicator lights. The eyebrow panel is hinged at the top to tilt upwards.

- The turnkey LOCK enables/disables all buttons except the power button.
- The POWER button controls the power to the CPU, memory, and peripheral controllers. It is protected by a hinged, transparent cover plate.

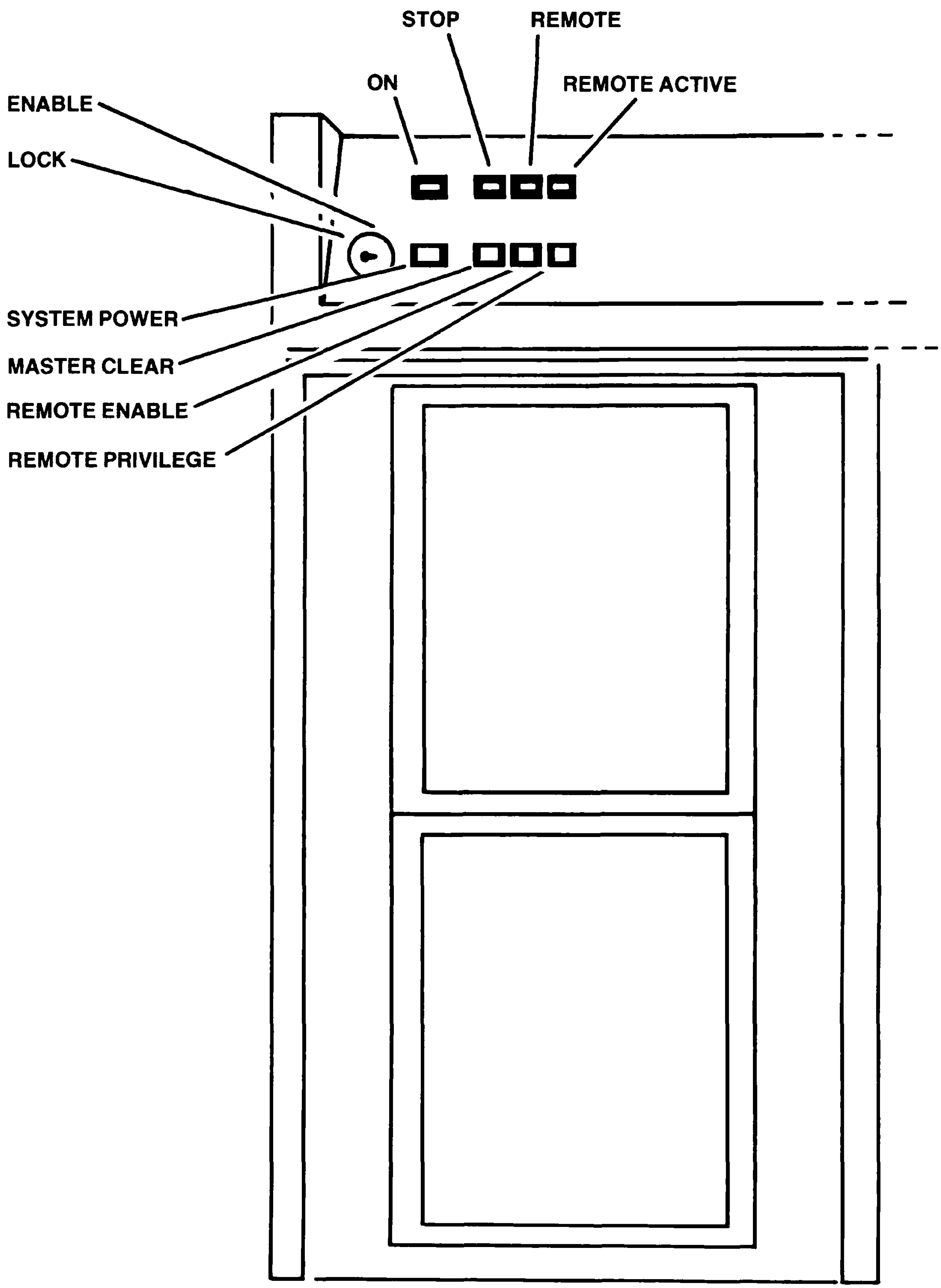
  When the light above the POWER button is on, the CPU has power.
- Pushing the MASTER CLEAR button initializes memory and verifies the CPU. It also halts the machine.

  When the light above the MASTER CLEAR button is on, the CPU has halted.
- The REMOTE ENABLE button enables/disables the remote supervisor terminal.

  When the light above the REMOTE ENABLE button is on, the remote supervisor terminal feature is enabled.
- The REMOTE PRIVILEGE button determines the access level of the remote supervisor terminal.

  When the light above the REMOTE PRIVILEGE button is on steadily, the remote supervisor terminal merely echoes what is typed at the local supervisor terminal. If the light is blinking, then the remote supervisor terminal can function as the local supervisor terminal.

Figure 2-1 illustrates a typical 50 Series cabinet.



The 50 Series Cabinet
Figure 2-1

The P850

The P850 is more powerful than other Prime CPUs, and requires two mainframe cabinets. The eyebrow panel on the P850 contains a lock, four buttons, and four indicator lights:

- The turnkey LOCK enables/disables all buttons except the power button.
- The POWER button controls the power to the CPU, memory, and peripheral controllers. It is protected by a hinged, transparent cover plate.

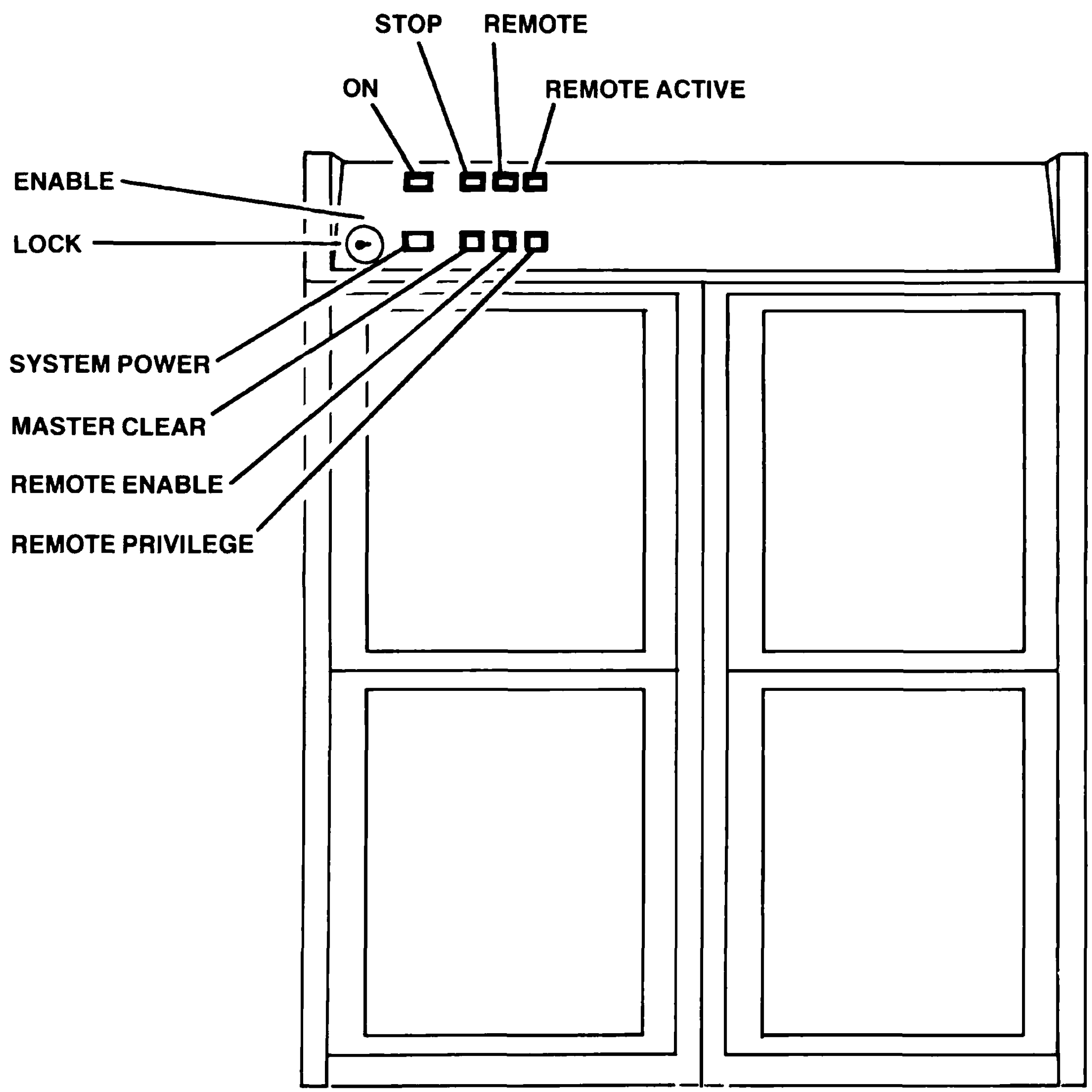
  When the light above this button is on, the CPU has power.
- Pushing the MASTER CLEAR button initializes memory and verifies the CPU.

  When the light above this button is on, the CPU has halted.
- The REMOTE ENABLE button enables/disables the remote supervisor terminal.

  When the light above this button is on, the remote supervisor terminal feature is enabled.
- The REMOTE PRIVILEGE button determines the access level of the remote supervisor terminal.

  When the light above the REMOTE PRIVILEGE button is on steadily, the remote supervisor terminal merely echoes what is typed at the local supervisor terminal. If the light is blinking, then the remote supervisor terminal can function as the local supervisor terminal.

Figure 2-2 illustrates the P850 cabinet.



The P850 Cabinet
Figure 2-2

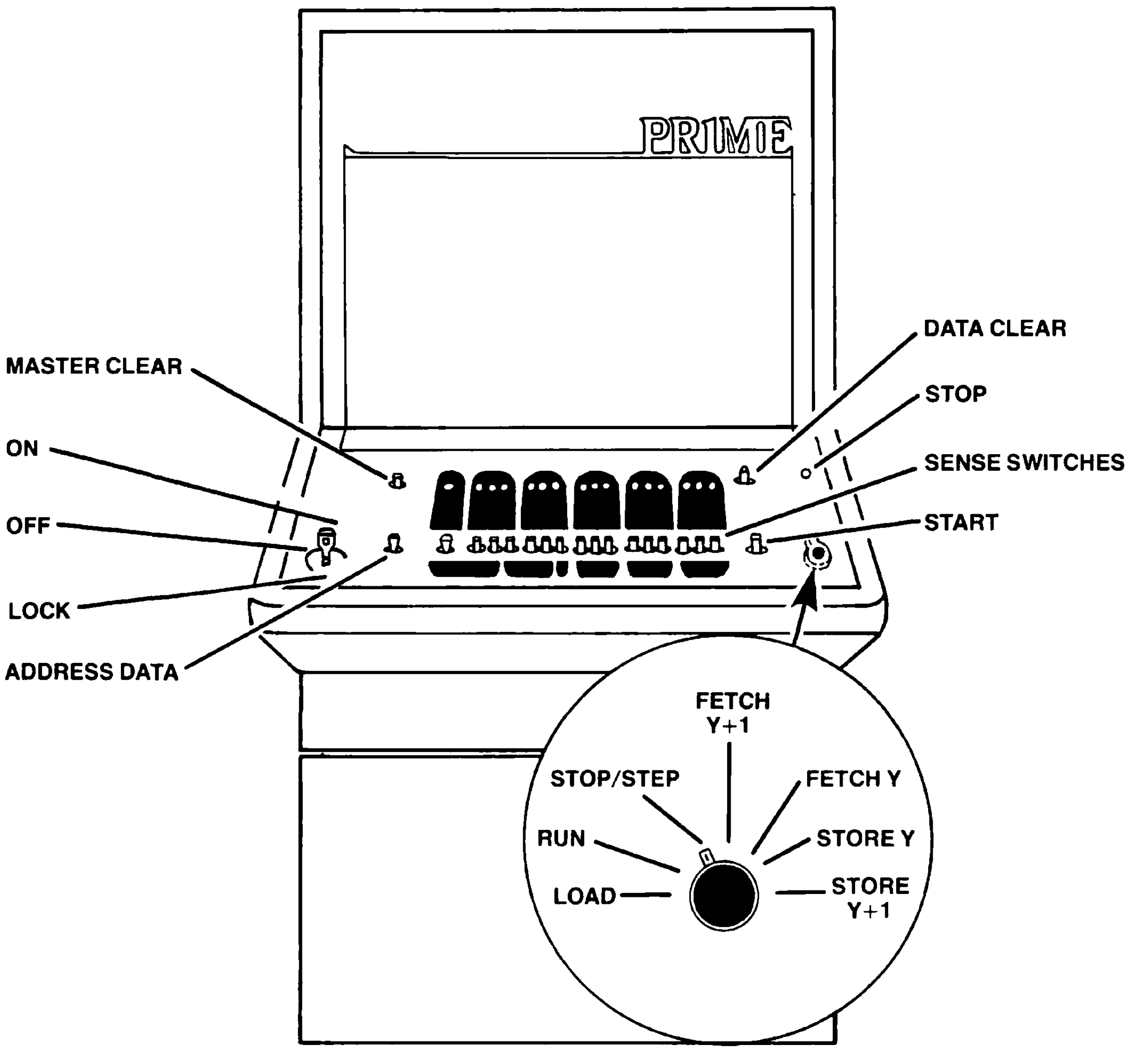
Earlier Prime Models (P300, P350, P400, P500)

Earlier models of Prime equipment are identifiable by the control panel extending from the center of the unit. The control panel has a lock, twenty toggle switches, and a large rotary switch.

- The LOCK has three settings: off, on, and lock. The lock position locks the other switches on the panel.
- Setting the MASTER CLEAR switch in the up position initializes memory and verifies the CPU.
- Setting the ADDRESS DATA switch in the up position sets addresses in the lights above the sense switches, and reads data from that memory address.
- The row of SENSE SWITCHES, labeled from 1 to 16, has two uses: indicating addresses and setting data. Each switch is set by placing it in the up position. Information is entered in octal format.

  The lights above the sense switches indicate addresses, and are read in octal.
- Setting the DATA CLEAR switch in the up position clears data from the lights above the sense switches.
- The START switch starts the CPU.
- The ROTARY switch is used to stop, load, or read certain addresses from the CPU. The seven settings for this switch are: load, run, stop/step, fetch y+1, fetch y, store y, and store y+1.

Figure 2-3 illustrates the cabinet for the P300, P350, P400, and P500.



The P300, P350, P400, P500 Cabinet
Figure 2-3

Inside the Computer

The upper and lower front panels are removable to grant access to the backplane. More general access can be attained by opening the door at the back of the cabinet.

An examination of the back of the machine reveals the following components:

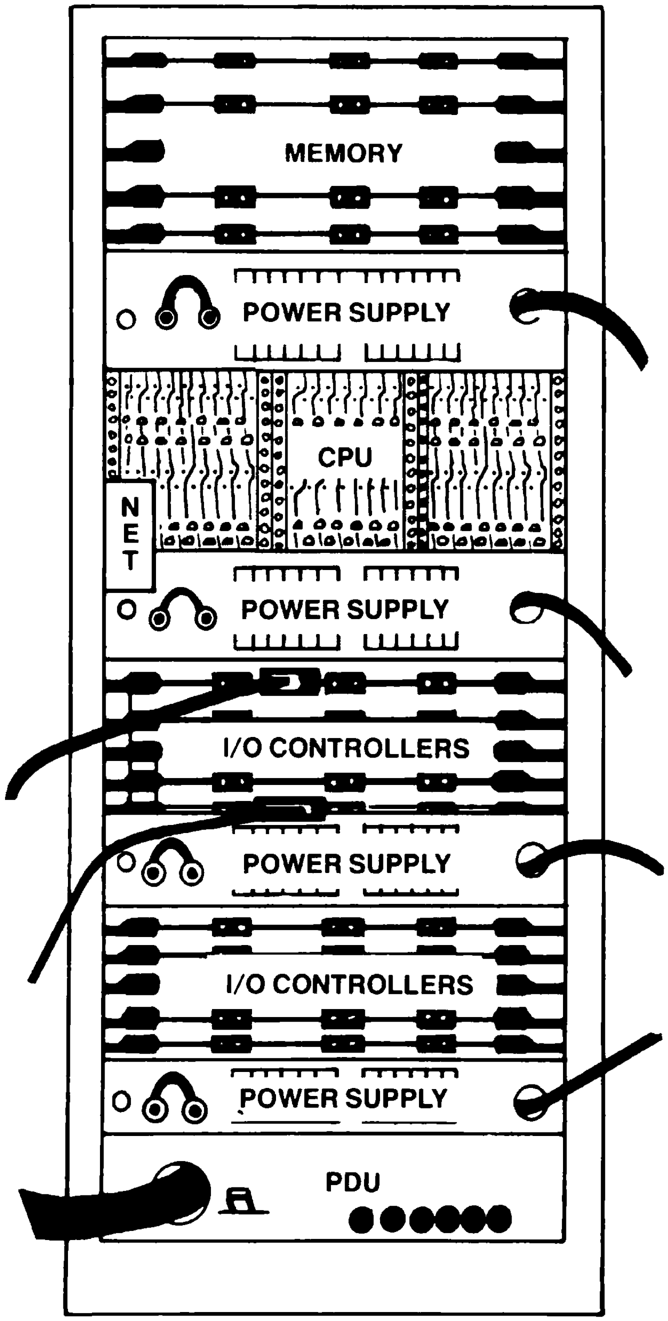
- Memory boards
- Power supplies
- The CPU (identifiable by its three "top hats," the green circuit boards that tie the multiple board CPUs together)
- Input/output controllers (identifiable by the cables connecting them with the various I/O devices), installed in the following top-to-bottom order:

    1. The parallel printer controller, supporting up to two printers and a card reader
    2. QAMLC boards (AMLC boards on earlier models), which are terminal controllers supporting up to 16 terminals each (four lines going into each of the board's four ports)
    3. Communications boards (for systems using PRIMENET or other communications products)
    4. Up to two disk controllers, each of which can support up to four disk drives
    5. A tape drive controller, supporting a maximum of four magnetic tape drives
    6. The VCP board, responsible for starting up the CPU, determining halt locations, and displaying the contents of memory. The VCP board also supports the supervisor terminal and a remote supervisor terminal (through a modem).

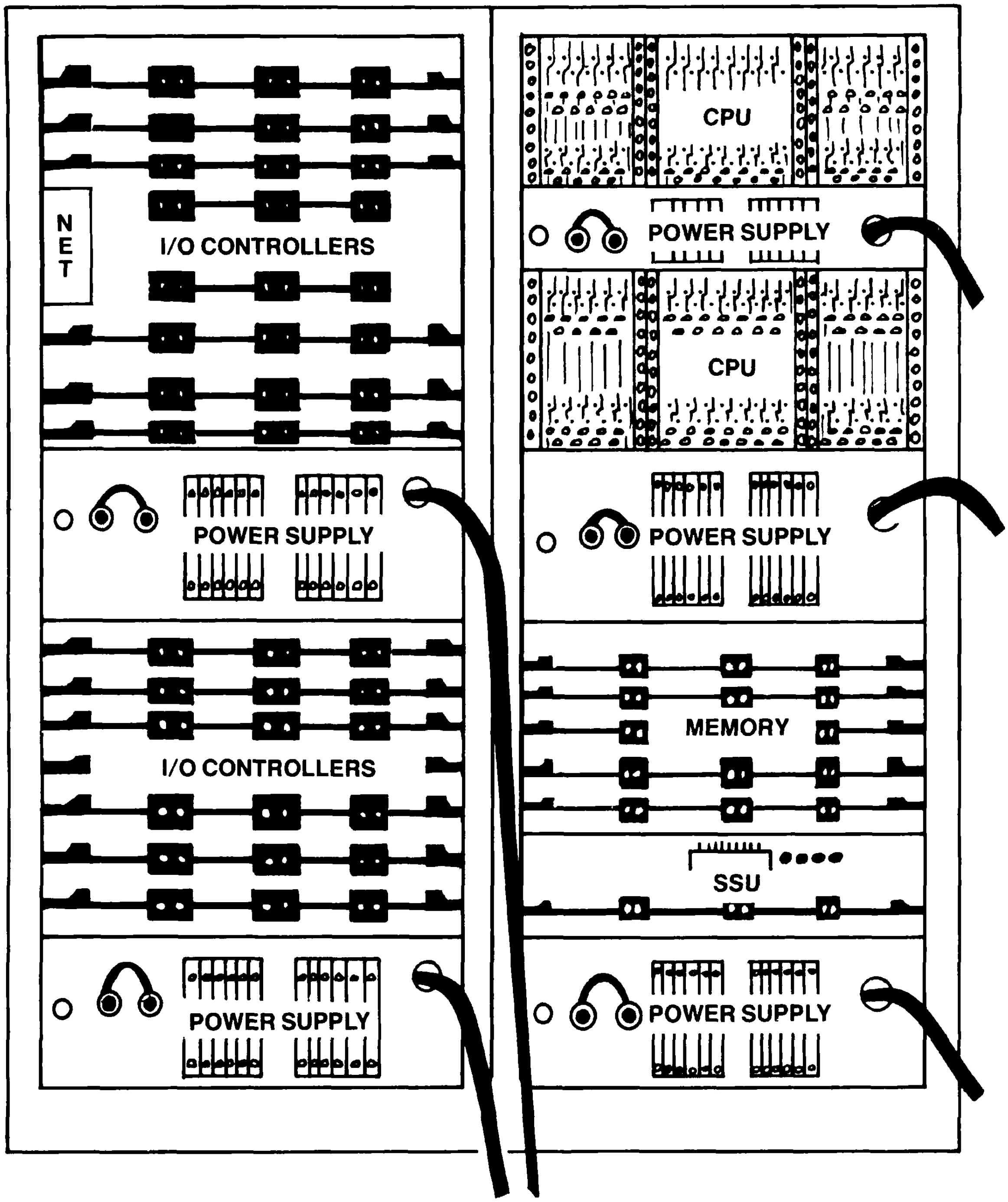
- A Power Distribution Unit (PDU) at the bottom of the cabinet, identifiable by its heavy power cable and by the white circuit breaker at the lower left

The number of power supplies, memory boards, I/O controllers, and other equipment (for example, a Prime Network Junction Box) will vary from system to system.

Figures 2-4 and 2-5 illustrate what you will see upon examining a backplane.



The Backplane of a Typical Prime Computer
Figure 2-4



The Backplane of the P850
Figure 2-5

THE SUPERVISOR TERMINAL

Although any terminal may serve as the supervisor terminal (as long as it is RS-232-C compatible), certain terminals are specifically designed for this purpose. The usual supervisor terminal supplied with Prime computers, the Terminet 30, has the following features:

- Standard keyboard layout
- Hard-copy printing capacity
- MOTOR button, used to turn on the printing mechanism
- LOCAL button, used to switch the terminal between online and local use. This button must be in the up position for online use.
- ALL CAPS button, used to type in uppercase only
- INTERRUPT button for clearing errors. (Errors, shown by an indicator light located near the INTERRUPT button, will frequently cause the keyboard to freeze. The two most common sources of such errors are: the terminal is out of paper; or the terminal's cover was raised and the interlock switch was not flipped up. After resolving the difficulty that caused the interrupt, push the INTERRUPT button to clear the keyboard.)

THE DISK DRIVE

Disk drives for Prime systems come in three general categories: storage modules, cartridge module devices (CMDs), and Winchester drives.

Storage modules are larger, stand-alone units using removable disk packs. They are loaded by lowering the disk pack into the device from above, then removing the pack cover. Follow this procedure:

1. Lift open the cover of the disk drive unit.
2. Remove the bottom cover of the disk pack by squeezing the release mechanism in the center of the pack's bottom.
3. Mount the pack on the drive shaft.
4. Tighten the pack on the drive by rotating its handle clockwise.
5. Remove the cover of the pack from the drive unit.
6. Close the cover of the disk drive.
7. Press the START button to ready the drive. Wait for the READY light.

Cartridge module devices are generally smaller, are mounted in the mainframe or tape drive cabinet, and have one removable platter and from one to five fixed (nonremovable) surfaces. CMDs are loaded horizontally, sliding the cartridge along a track located at the top of the CMD. Follow this procedure:

1. Release the bottom of the cartridge.

2. Open the CMD door.

3. Slide the cartridge onto the drive.

4. Close the drive door.

Winchester drives vary widely in size, may stand alone or be mounted in the mainframe or tape drive cabinet, and have as many as 40 fixed disk surfaces. Winchesters, because they are a sealed storage medium, are not loaded.

Disk drives are controlled with the following buttons:

- The START button (START/STOP on the CMD) is used for startup and shutdown of the disk drive.

  The device is active when the indicator light above this button is on.

- The READY button has the drive unit number printed on it, and is nonoperative.

  The light above this button indicates whether the disk pack is ready for use. A blinking light indicates the drive is not ready for use as the disk has not yet reached maximum speed; a steady light indicates maximum speed has been achieved and the drive is ready for use.

- The FAULT button (FAULT/CLEAR on the CMD) is pressed to clear a fault.

  The indicator light goes on to indicate a problem with the disk or the drive. If this light comes on and stays on, the drive and the disk should be promptly checked to determine the nature of the problem.

- The WRITE PROTECT button, when pushed, effects write protection on both the disk pack and the Winchester drive.

  A light indicates that write protection has been enabled.

  CMDs are write protected somewhat differently. To write protect the removable disk (the cartridge), push the WRITE/CART button. To write protect the fixed disk(s), push the PROTECT/FIXED button.

Circuit breakers or fuses for these devices are located behind the door on the back of the drive.

Figure 2-6 illustrates a storage module, a cartridge module device, and a Winchester drive.

THE MAGNETIC TAPE DRIVE

Prime tape drives operate at 45 ips (inches per second) or 75 ips, and with tape densities of 800 bpi, 1600 bpi, or 6250 bpi.

The 800/1600 BPI Magnetic Tape Drive

The 800/1600 bpi tape drive is controlled with four buttons and a small density setting switch:

- The POWER button provides power to the unit. This button must be pushed prior to loading a tape.

  The indicator light above this button lights when power is on.

- The REWIND button is pressed to rewind a tape.

  The indicator light above this button is on when a rewind is in progress.

- The LOAD button is pressed to bring the tape to the load point marker.

  The indicator light above this button lights when the tape has been loaded to the beginning of tape (BOT) marker.

- The ON LINE button is pushed to enable transmission of data between the CPU and the tape drive.

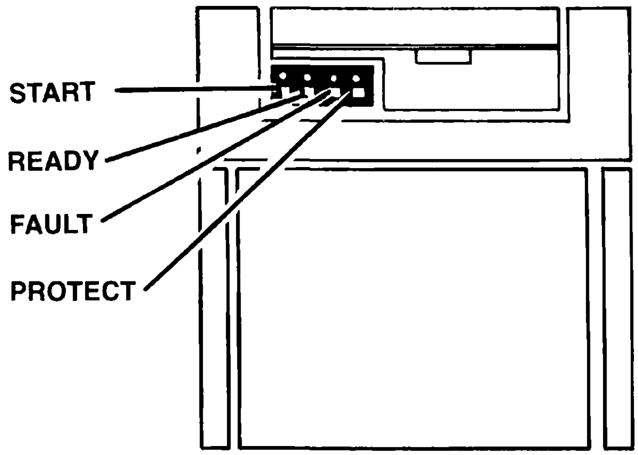
  The indicator light above this button lights when the drive is on line to the CPU.

- The density switch, located at the left of the panel, is a three-position switch that sets tape density at either 800 bpi or 1600 bpi. If the switch is left in the middle position, density is set by software.

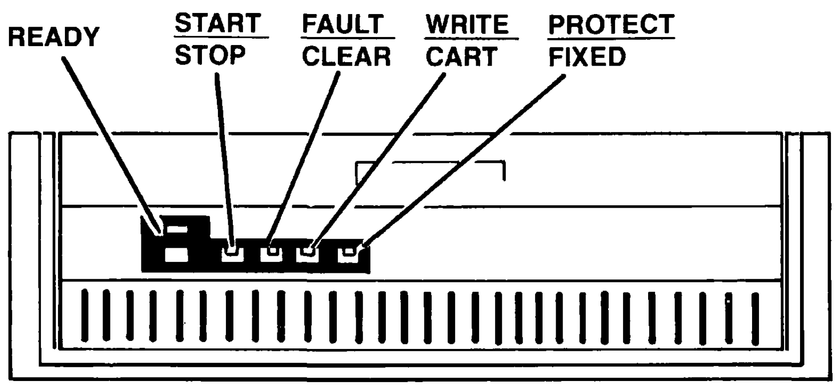
Four small status lights are grouped together:

- The SELECT light is on when the drive has been selected by the CPU.

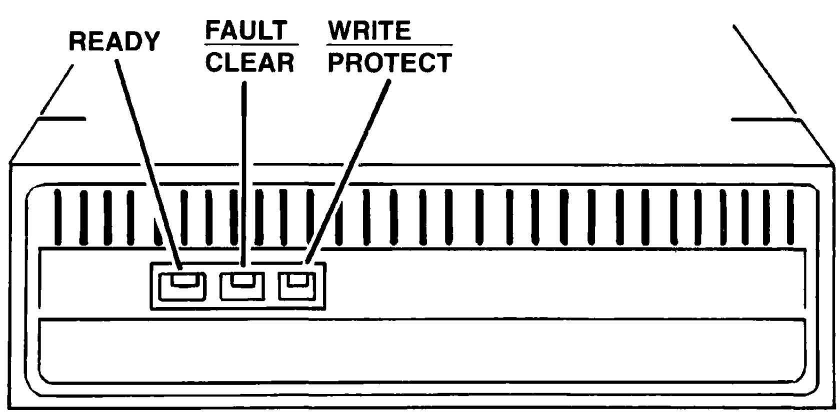
- The WRITE ENABLE light is on when the tape has a write ring mounted on it.



STORAGE MODULE



CARTRIDGE MODULE DEVICE



WINCHESTER DRIVE

Storage Module, Cartridge Module Device, and Winchester Drive
Figure 2-6

- The READ light is on when the tape is being read.
- The WRITE light is on when the tape is being written.

The small number at the upper left of the top panel indicates the tape drive number. The tape drive number can be set from 0 to 3 or from 4 to 7 (depending on the controller) with the small notched wheel at its left.

The tape drive's main circuit breaker or fuse is located at the back of the cabinet, in the lower left-hand corner.

See Figure 2-7 for an illustration of the 800/1600 bpi magnetic tape drive.

The 6250 BPI Magnetic Tape Drive

This tape drive operates at 75 ips, with tape density settings of 1600 bpi and 6250 bpi. The drive is operated with six control buttons mounted vertically on the left side of the front panel, and by a formatter power button on the front panel's lower right side. The drive is monitored through the use of indicator lights.

- The RESET button is pushed to cancel instructions given by the other buttons and to reset tape mount errors.

  The LOAD CHECK light adjacent to this button lights to indicate an error in tape loading has occurred.

  The next light, labeled SELECTED, lights when the tape is accessed.

- The LOAD/REWIND button is pushed to load and rewind tapes.

  The adjacent LOAD POINT light indicates that a tape is at the load point.

- The ON LINE button is pushed to enable transmission of data between the CPU and the tape drive.

  The ON LINE indicator light next to this button lights when the drive is on line to the CPU.

- The DENSITY button is pushed to select tape density. Tape density may also be set by a software command, in which case the button is not pressed.

  The adjacent light, labeled 1600 DENSITY, is lit in the 1600 bpi mode, and unlit in the 6250 bpi mode.

- The UNLOAD button is pushed to unload a tape.

The adjacent light, labeled FILE PROTECT, lights to indicate that the mounted tape is write protected.

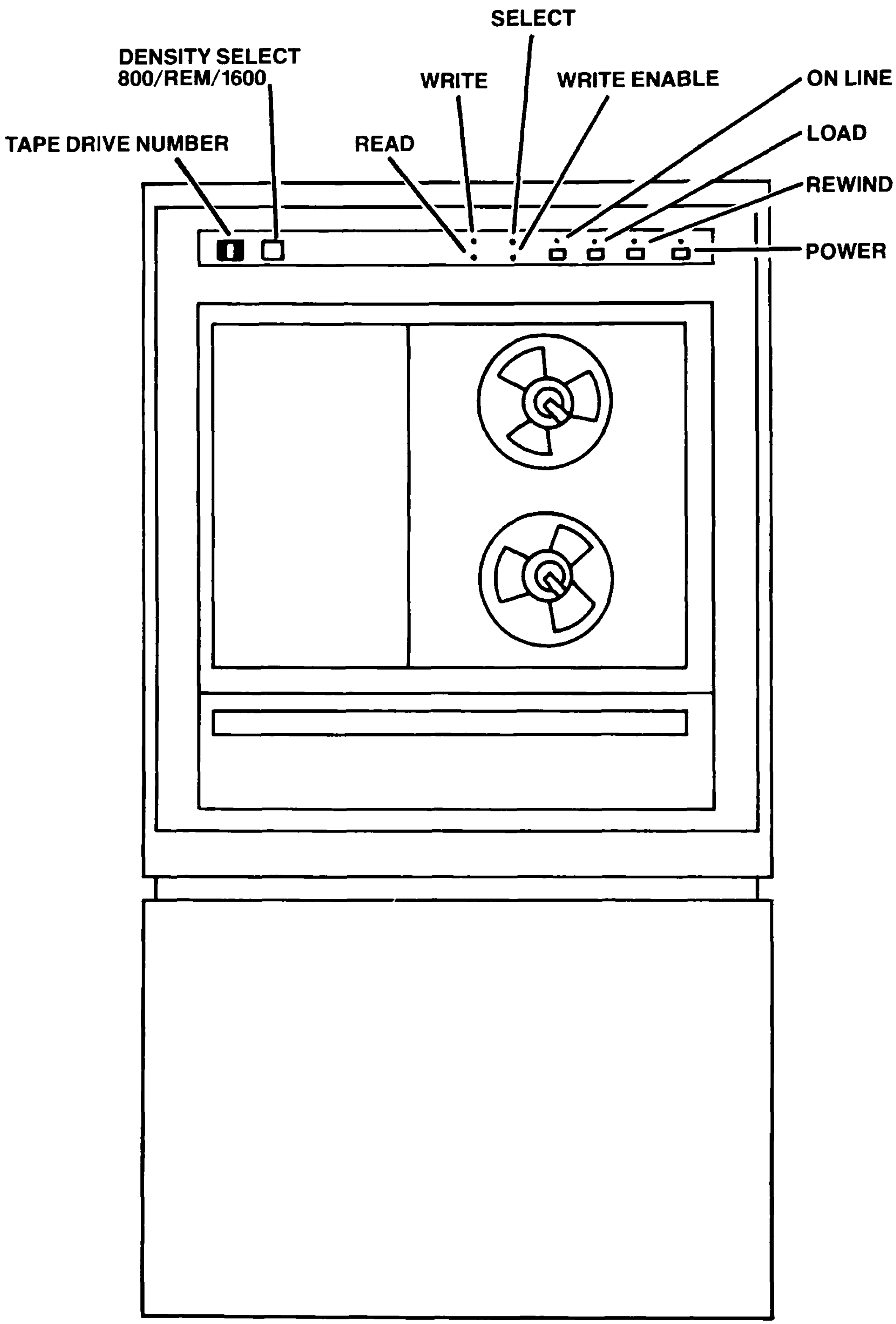
- The POWER ON button provides power to the unit. This button must be pushed prior to loading a tape.

  The indicator light adjacent to this button lights when power is on.

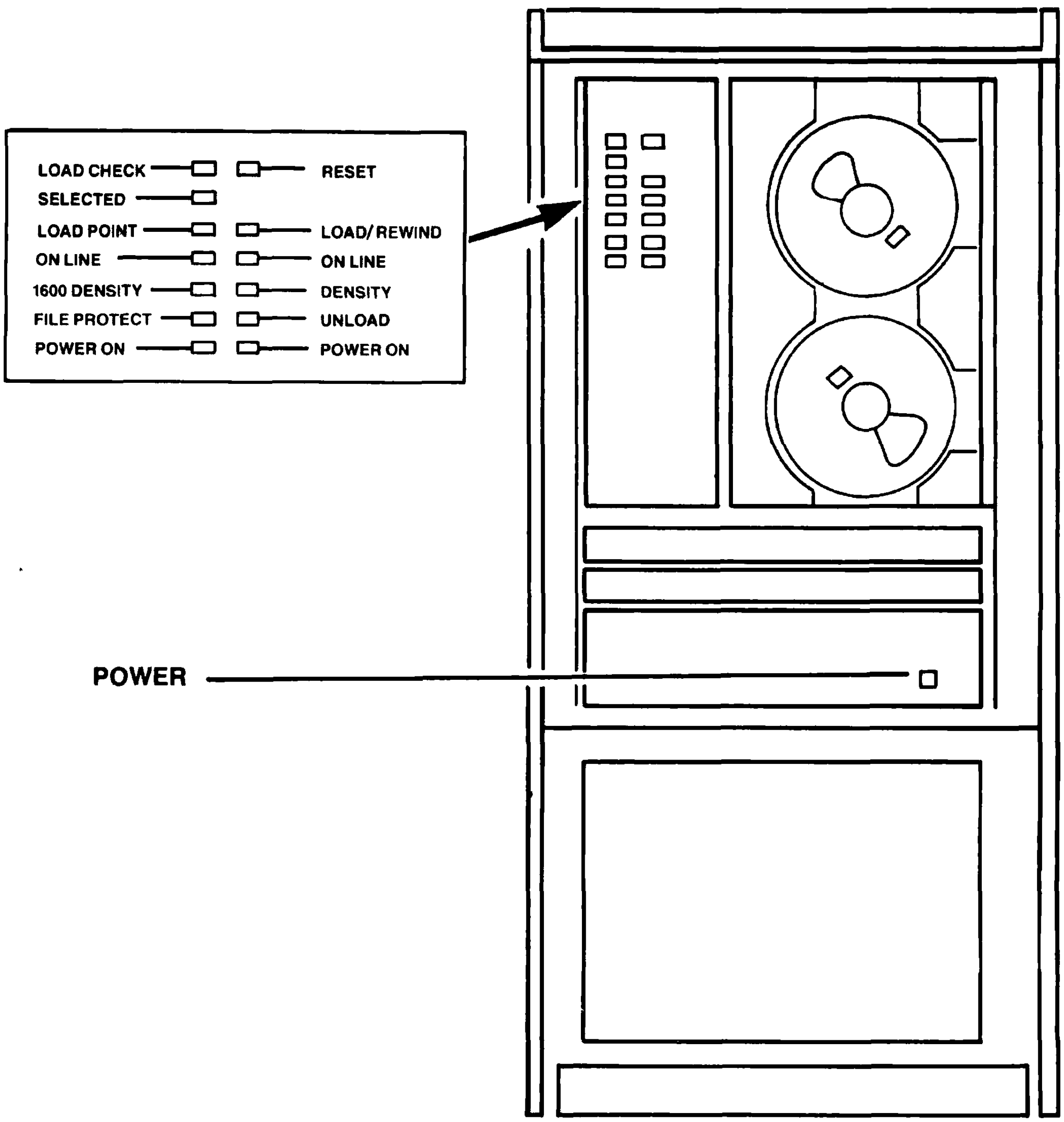
- The formatter POWER ON button turns on the tape formatter unit. The formatter interprets signals from the CPU, and must be turned on before the tape drive is assigned.

The tape drive's main circuit breaker or fuse is located at the back of the cabinet, in the lower left-hand corner.

See Figure 2-8 for an illustration of the 6250 bpi magnetic tape drive.



The 800/1600 BPI Magnetic Tape Drive
Figure 2-7



The 6250 BPI Magnetic Tape Drive
Figure 2-8

THE PRINTER

Several types of printers are used with Prime equipment. Typically, machines located in the computer room will be geared to large quantity (rather than letter quality) output. Printers that print an entire line (rather than a single character) at a time are frequently referred to as line printers.

Prime's standard line printers produce 300, 600, or 1000 lines per minute, and are controlled by six buttons and a special indicator light. Each button lights when engaged.

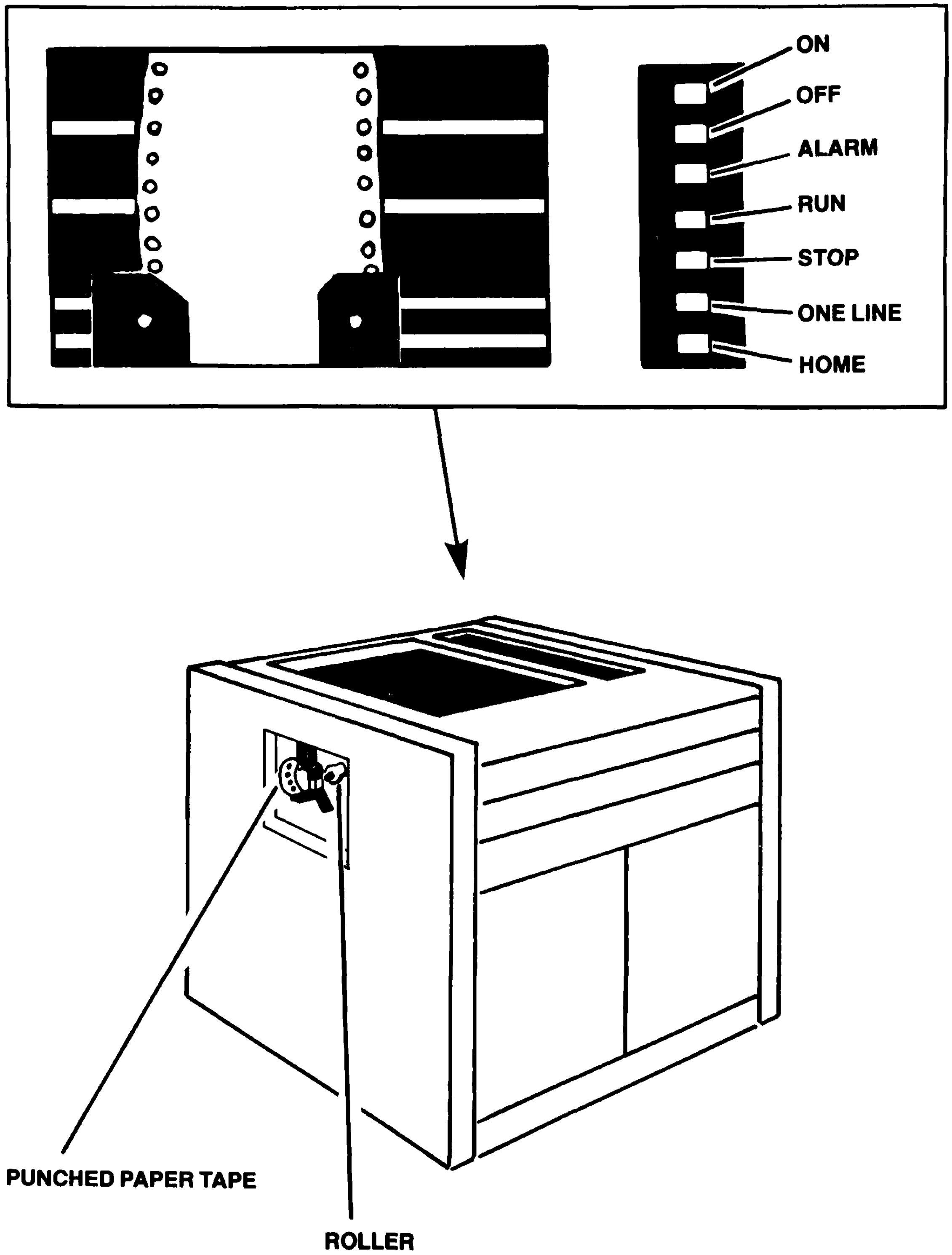
- Pushing the ON button turns the power on.
- Pushing the OFF button turns the power off.
- The ALARM light does not move. When lit, it indicates that a problem exists with the printer. To determine the specific problem, open the printer's hood and look at the labeled lights on the Alarm Status Indicator Panel.
- Pushing the RUN button enables the printer.
- Pushing the STOP button takes the printer off line. The power remains on.
- Pushing the ONE LINE button advances the paper one line.
- Pushing the HOME button advances the paper to the top of the next page.

Inside a sliding window on the left side of the printer are two controls:

- A punched paper tape that controls setting the top of a printing form.
- A roller for advancing the paper when lining up a form. The roller has a lock latch on the end. This must be unlocked to line up forms, and locked during normal operation.

The printer's circuit breaker is accessible through a door located in the back, on the lower left-hand side.

See Figure 2-9 for an illustration of a printer.



The Line Printer
Figure 2-9

3
The Operator's Tasks

INTRODUCTION

The role of the operator is a crucial one in any data processing installation. While specific assignments and responsibilities may vary from one computer room to another, there are a number of tasks routinely performed by the operator that will be a part of your job no matter where you work. Typically, your function is defined by the System Administrator.

Each operation you perform will generally fall within one of the following task groups:

- General monitoring of hardware and computer room conditions
- Starting up and shutting down the system (see Chapter 4)
- Allocating and monitoring system and subsystem resources (see Chapter 5)
- Preparing disks for use by the system (see Chapter 6)
- Performing backups (see Chapter 7)
- Repairing the partitions on a disk (see Chapter 8)
- Monitoring line printers and replacing printer paper and ribbons (see Chapter 9)
- Assisting users with magnetic tape assignments (see Chapter 10)

- Monitoring the Batch subsystem (see Chapter 11)
- Making sure data communications systems are functioning properly (see Chapter 12)
- Preventing and recovering from system halts (see Chapter 13)
- General system responsibilities

In this chapter you will find a brief outline of your functions as they relate to these tasks. For more specific information, refer to the appropriate chapters in this guide.

GENERAL MONITORING OF THE COMPUTER ROOM

Operators generally monitor all aspects of activity in the computer room, making sure that hardware is functioning properly so that it can efficiently serve the user community. Particular attention is paid to:

- Air conditioning units, to ensure that the temperature and humidity of the computer room are within acceptable limits
- The CPU, to make sure that it is operational at all times
- Disk drives, to make sure that they are online and functioning normally
- Tape drives, which must be cleaned on a regular basis to ensure proper operation
- Printers, which must be cleaned on a regular basis and have paper and ribbons replaced frequently to ensure that legible printouts are available to users
- Modems and multiplex units, to make sure that data can move throughout the network

Specific tasks in a data center will be defined by the System Administrator.

SYSTEM STARTUP AND SHUTDOWN

Starting the computer and its peripheral devices is the most basic of a computer operator's responsibilities. This function is covered in Chapter 4.

ALLOCATING AND MONITORING SYSTEM RESOURCES

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 15). 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 a MAKE, FIX_DISK, or COPY_DISK operation (among others). 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 15).

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 15.

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 15 for complete details.

Monitoring Disk Quotas and Space Utilization

Size limits (quotas) may be set on a system's UFDs to ensure a fair sharing of disk storage space. These quotas may be set, examined, and reset by the operator according to rules established by the System Administrator. (The commands used to do this are listed in Chapter 5.)

Although UFDs may have specific quotas limiting their size, it is possible for users to exhaust the number of free records on a partition. When this occurs, users may be temporarily unable to perform necessary tasks. Thus, the operator should make sure that there is always usable space on a partition.

The operator uses the AVAIL command to determine the amount of room still left on a partition. When disk partitions become very full (98% and up), the operator should establish the amount of space used by each UFD on the partition. (The commands used to do this are listed in Chapter 5.) The list of UFDs that is generated should be distributed and posted. Then, those individuals using the most disk space can be asked to reduce the number of records they are using.

When more disk space is needed immediately, a solution is to use the MESSAGE command to request that users delete unnecessary files. If this does not achieve satisfactory results, it may become necessary to increase the number of records on the partition.

Whenever the "Disk-Full" condition is encountered, the System Administrator should be informed.

Monitoring User Status

The STATUS command, described in Chapter 5, 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.

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.

Keeping System Records

An important part of the operator's duties is the maintenance of records regarding important events that affect the system.

Using the Event Logger: The PRIMOS event logger automatically records major system events (such as warm starts, cold starts, disk startups, etc.) and writes them to a file, named LOG.mm/dd/yy, in the UFD LOGREC*. The contents of this file are examined with the LOGPRT command. A detailed description of the event logging system can be found in Chapter 5; details regarding the logging system's messages are located in Appendix J.

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 Network Event Logger: The PRIMOS network event logger automatically records major network events (such as operator shutdowns, event buffer overflows, out of sequence packets, etc.) and writes them to a file, NET_LOG.mm/dd/yy, in UFD PRIMENET*. The contents of this file are examined with the LOGPRT command. Details of the network event logging system are in Chapter 5; details regarding network event messages are in Appendix K.

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

If You Use a Video Display Unit: 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 Chapter 4. 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 RWLOCK of UPDT (allowing multiple readers plus one writer). 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.

Keeping a Logbook: The logbook is used to record significant system conditions or changes. These include hardware configuration, disk partitions, system halts, backups, etc. A major purpose of the logbook is to allow reconstruction of the system history. Complete logbook procedure is detailed in Chapter 5. In logging events:

- All entries should have time and date (including year).
- All entries should be signed. The Administrator or system analyst will then be able to check with the operator for clarification or additional information.

DISK FORMATTING

Before a disk pack can be used on the system, it must be formatted by 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.

BACKING UP

Backing up is the process of copying information from system disks onto other storage media (disks or magnetic tape). Backups are performed to provide copies of files or programs for use if the versions active on the system are damaged or lost. Such damage or loss may be due to user or operator error, system halt (from power failure, etc.), or faulty peripherals.

Backing up should never be done during peak hours of system usage. Typically, backups are done early in the morning, after a cold start, late at night, or on weekends.

The principal commands associated with the backup procedure are COPY_DISK, PHYSAV, and MAGSAV, described in Chapter 7. The FIX_DISK utility (see Chapter 8) is also frequently used.

Backup procedures may be performed by CPL files or COMINPUT files.

Backed up files are normally recovered after bringing up the system but before allowing users to log in. Usually this entails simply starting up a disk and using the COPY command. The recovery procedures from disk (COPY_DISK) and magnetic tape (MAGRST and PHYRST) are described in Chapter 7.

MAINTAINING FILE SYSTEM INTEGRITY

File system and directory integrity are checked and repaired, if necessary, by the FIX_DISK utility. In addition to checking integrity, FIX_DISK performs general file housekeeping by compressing directories. FIX_DISK should be run on a regular basis by the operator. It has been found to be convenient to perform the FIX_DISK operations in conjunction with disk-to-disk system backups. (See Chapter 7 for a sample schedule.) A complete description of FIX_DISK usage may be found in Chapter 8.

The FIXRAT utility may be used in place of FIX_DISK with pre-Rev. 19 disks. This is <u>not</u> recommanded. A complete description of FIXRAT usage is located in Chapter 14.

> **Caution**
>
> FIXRAT may not be used with Rev. 19 disks.

OPERATING LINE PRINTERS

The operator can use the PROP command to control printers and their environments. 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. (The PROP command is explained in detail in Chapter 9.)

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 5).
- 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 twice a day, or more often when necessary.

WORKING WITH MAGNETIC TAPE

The extended ASSIGN command for magnetic tapes provides for better utilization of magnetic tape resources. It is designed primarily for use in CPL programs and COMINPUT files. It allows such programs (or any user who wishes to use the command) to request operator assistance in assigning a drive or mounting a tape. The operator may assign drives based upon user-supplied attributes.

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.

ASSIGN Process

Each mag 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 operator may unassign a tape drive from any user. This is useful when the user's job "hangs" but the user does not wish to be force logged out.

Assignment Functions

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 SETMOD, ASSIGN, and UNASSIGN commands, as well as a discussion of the operator's responsibilities regarding magnetic tape, can be found in Chapter 10.

CONTROLLING THE BATCH SUBSYSTEM

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.

These tasks are explained in Chapter 11.

DATA COMMUNICATIONS

If your system is part of a network, you will have a special set of network-related tasks. The most common responsibilities associated with network communications are:

- Adding and shutting down remote disks
- Monitoring the File Transfer Service

Adding and Shutting Down Remote Disks

The ADDISK and SHUTDN commands are used for starting up and shutting down both local and remote disks. These two commands are described in general in Chapters 4 and 15. Specific reference to their use with systems or networks is found in Chapter 12.

Monitoring the File Transfer Service (FTS)

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.

HANDLING SYSTEM HALTS

The procedures for handling a system halt vary with the reason for the halt. The reason can usually be determined by obtaining the halt location from the control panel. The operator should become familiar with the Virtual Control Panel operations described in Appendix B or the control panel operations described in Appendix C. For information on the handling of halts, halt location meanings, and restart procedures, see Chapter 13 and Appendix E.

GENERAL SYSTEM RESPONSIBILITIES

The operator may frequently be called upon to do the following:

- Send messages to users via the system
- Add new UFDs to the system
- Set, modify, and remove quotas on directories
- Add software to the CMDNC0 and LIB UFDs
- Control logins and logouts

Sending Messages

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 (e.g., system to be shut down, new utility available), communicating with a single user (e.g., answering questions, requesting action), or for passing information between nodes (e.g., remote disk availability). For information on the format of the MESSAGE command see Chapter 15.

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.

If you create a new directory, its quota will initially be set to zero; that is, it has no maximum quota. If a maximum quota is set on a UFD, it will limit the actual storage allowance on any subdirectory within the directory. If no limit exists 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 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 may create a new UFD that is a user's initial attach point.

Because access to the MFD is not available to users, the operator must also perform all requests for UFD name changes (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 an 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). The format is:

$$\left\{\begin{array}{l}\text{SET\_QUOTA}\\ \text{SQ}\end{array}\right\}\ \text{pathname -}\underline{\text{MA}}\text{X number}$$

pathname — The pathname of the directory having its quota set. If you want to set a quota on the current directory, you must use the full pathname. If you want to set a quota on a subdirectory within the current directory, you need specify only the simple name (the final element of the full pathname). A quota cannot be set on the current directory, or on a sub-UFD that is a non-quota directory (see note below).

-MAX number The maximum number of records the directory can use.

For example:

```
SET_QUOTA UFD.1 -MAX 300
```

Note

When you try to set a quota on a directory under a certain set of conditions, you will receive the error message:

```
File in use. directory-name (set_quota)
ER!
```

This message will appear when both of the following conditions are true:

- When the directory has no current quota (that is, quota = 0)
- When there are attached users or open files in the directory or its subtree

You may be able to remedy this situation. If you are the only user of the directory subtree, attach to another UFD or to a level higher in the tree than the directory whose quota you wish to set. The SET_QUOTA command should now execute.

If you are not the only active user of the directory subtree, you must wait for all other users to attach elsewhere or to log out before you can set the quota.

Modifying Quotas on Directories

Once a quota exists on a directory, you may raise, lower, or remove it with a new SET_QUOTA command.

Raising or Lowering a Quota: The format for raising or lowering a quota is the same as for establishing a quota on a non-quota directory:

```
SET_QUOTA pathname -MAX number
```

Removing a Quota from a Directory: You may remove an existing quota from a directory by setting the quota to zero. Any of the following command formats will set a directory quota to zero:

```
SET_QUOTA pathname
SET_QUOTA pathname -MAX
SET_QUOTA pathname -MAX 0
```

Adding/Replacing Software in CMDNC0 and LIB

CMDNC0 and LIB are ACL or password-protected directories under operator control. 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.

For best performance, all names of runtime (R-mode) programs should end with the .SAVE suffix. All names of CPL programs must end with the .CPL suffix. Users do not have to type these suffixes when invoking the commands. If a user types command-name, the command processor looks in CMDNC0 for command-name.SAVE. If it doesn't find that, it looks for command-name.CPL. If it doesn't find that, it looks for command-name.

The R-mode loader, LOAD, adds the .SAVE suffix automatically when it creates a file. The System Administrator may change the names of already existing runtime programs to add the .SAVE suffix.

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.

Installation of Programs in the Command UFD (CMDNC0): Runtime programs and CPL programs in the command UFD (CMDNC0) can be invoked by keying in the program name alone. This feature of PRIMOS is useful if a number of users invoke this program. Only one copy of the program need reside on the disk in UFD CMDNC0. Space is saved during execution by multiple users if the program uses shared code.

Installation of commands in the command UFD is extremely simple.

Runtime and CPL Programs: Use COPY to copy the program into CMDNC0. Once placed in CMDNC0, the program will be invoked when users type the program name as a command.

Segmented Runfiles Saved by SEG's Loader: As the PRIMOS command processor cannot directly handle SEG runfiles, a segmented program cannot be run directly from UFD CMDNC0. The segmented program may be invoked by means of a nonsegmented interlude program in CMDNC0.

The procedure for creating an interlude is:

1. Create the desired SEG runfile.

2. Run the command file SEG>CMDSEG using COMINPUT; it will ask for runfile pathname as the new SEG runfile name. This command file will create the interlude program under the name *TEST.

3. If you did not give a pathname for the runfile, make a copy of the SEG runfile in UFD SEG using COPY or FUTIL's TRECPY command. The name of the new SEG runfile should be the name by which it will be invoked.

4. A copy of *TEST should be placed in UFD CMDNC0 using the COPY command. The filename should be that by which the program will be invoked.

When filename is entered at the user terminal, the filename interlude program in CMDNC0 is executed. This program attaches to the SEG UFD, restores the segmented runfile filename, reattaches to the user's home directory, and begins execution of the SEG runfile.

If the SEG runfile requires only one segment of loaded information (procedure, link frames, and initialized common) in user space (segment '4000 and above) it is possible to include the interlude in the SEG runfile. This is discussed in the LOAD and SEG Reference Guide.

Controlling Logins and Logouts

In certain circumstances, it may be necessary for the operator to limit users' access to the system.

Limiting Logins: The maximum number of users allowed to be logged in to the system can be decreased from the configured value by the command: MAXUSR n. Here, n is an octal integer that is less than the configured number of users. This number specifies the maximum allowed number of logged in users. If the number of users logged in exceeds n, no users will be logged out but no new users may log in and no phantoms may be started. This feature is useful when:

- Modifications are to be made to PRIMOS at a cold start. Before setting the system time and date, set MAXUSR to 1 or 2, preventing users from logging in until the operator resets MAXUSR.

- The system is to be shut down. Set MAXUSR to 0 to prevent any new logins just prior to shutdown. (See also the section on forcing logouts, below.)

Forcing Logouts: From the supervisor terminal, the operator may forcibly log out any user by the LOGOUT command.

The connection of a process to a terminal over the network is forcibly broken by the LOGOUT command. Individual processes are logged out with the command: LOGOUT -user-number. user-number is the decimal number of the terminal or phantom 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 users--remote, local, and phantom--can be forcibly logged out by the LOGOUT ALL command. The only exceptions are user 1 (the supervisor terminal) and NETMAN, which are never affected by the LOGOUT command. No users may log in until MAXUSR is reset. LOGOUT ALL can be issued just prior to a SHUTDN ALL command to allow a more orderly shutdown of PRIMOS.

Notes

Before logging out or disconnecting users, the operator, as a matter of courtesy, should use MESSAGE to notify the users in advance, allowing them to reach a reasonable stopping place in their work.

The operator can make sure that all users and phantoms are logged out by issuing the command STATUS USERS.

User 1 (the supervisor terminal) and NETMAN are not affected by the LOGOUT command.

4
System Startup and Shutdown

INTRODUCTION

This chapter outlines procedures involved in the startup and shutdown of your Prime computer system. The section on system startup describes how to perform a standard boot from disk. Refer to Appendix A for information on:

- Booting from magnetic tape
- Booting from nonstandard devices
- Limiting the memory available for a PRIMOS II boot
- Interpreting error codes that appear in the control panel lights

Principal discussions in this chapter focus on:

- General startup procedures
- System startup for machines with a virtual control panel
- System startup for machines with a control panel
- What happens during system configuration
- General shutdown procedures

If Your Equipment Has a VCP: The VCP (Virtual Control Panel) allows the operator to perform most command functions at the supervisor terminal. For a general guide to the VCP, see Appendix B.

If Your Equipment Does Not Have a VCP: Some earlier Prime computers do not have a VCP. Such machines may be identified by a rotary switch and row of toggle switches on the front panel. If yours is an earlier system, many commands must be entered at the computer itself rather than via a terminal. Refer to the section of this chapter on system startup on machines without VCPs. For a guide to the general operation of this type of control panel, see Appendix C.

GENERAL STARTUP PROCEDURES

The startup procedure should be performed in this order:

1. Make sure all circuit breakers are turned on.
2. Turn on power to equipment.
3. Bootstrap in PRIMOS.
4. Set system time and date.

Turning on Equipment

Turn on power to equipment in this order:

1. Turn on power to the CPU.
2. Turn on all disk drives; wait for ready lights to go on.

> **Caution**
>
> If your equipment does not have a VCP, an attempt to boot in PRIMOS before the disk drives are ready will fail. No message will be printed at the supervisor terminal. If this happens, wait for the drives to become ready (ready light on), push the MASTER CLEAR button, and repeat the boot procedure.

3. Turn on magnetic tape drives that are to be used.
4. Turn on all dataphone sets; dataphones should be set to automatic answer.

5. Other peripheral devices, such as line printers, may also be turned on at this time.

Note

A system may be connected to more than one tape drive or line printer through controllers. All tape drives or line printers on any controller must be turned on in order for any tape drive or line printer on that controller to function.

SYSTEM STARTUP ON MACHINES WITH VCP

After the CPU has been turned on, the VCP executes self-verify routines to insure its own integrity. The progress of these tests is indicated in the cabinet's REMOTE ENABLE and REMOTE PRIVILEGE lights as follows:

01 Dynamic memory test

10 Serial ports test

11 High speed buffer and mapper test

The STOP light will be on during this prodecure.

The VCP will attempt to bring the processor on line. If the processor has passed its internal self-verify routines, the VCP is ready to accept command input. The following message is printed at the supervisor terminal:

```
*** CPU VERIFIED ***
```

The CP> prompt, indicating control panel mode, then appears.

Notes

If an error occurs while the VCP is verifying itself, the lights will indicate which test failed and an error message will be printed on the supervisor terminal:

```
$$$ VCP DID NOT VERIFY $$$
error message
```

If the VCP fails to print any message at the supervisor terminal after the machine has been turned on, this is an indication that the machine is hung. A hung machine will continue to wait for the processor to verify itself. This activity may be terminated by typing a CONTROL-P or pushing the MASTER CLEAR button. Either action will cause the supervisor terminal to print the CP> prompt, indicating VCP command level.

Do not push MASTER CLEAR if you anticipate taking a tape dump.

If an error message has appeared, enter the contents of the error message in the system logbook. If the system has hung, make an appropriate entry in the logbook. Attempt to start the system again. If the same error occurs, contact your System Administrator.

Automatic Boot

In the standard procedure for booting PRIMOS, the peripheral device is started automatically and PRIMOS is automatically brought up. Type in:

CP> BOOT 14114 (or BOOT 14134 if booting from controller 1)

The system will respond with:

```
PRIMOS II REV. 19.0 10/31/81 (AT 170000)
STARTING UP DISK nnnnnn
A PRIRUN
CONFIG
```

PRIMOS then boots.

Note

This automatic booting procedure may be used only with a Rev. 19 top partition on a device.

Manual Boot with Automatic Device Startup

When operations in PRIMOS II are required, type:

CP> BOOT 10114 (or BOOT 10134, if booting from controller 1)

The partition's physical device number will automatically be entered. The system will respond with:

```
PRIMOS II Rev. 19.0 10/13/81 (AT 170000)
STARTING UP DISK nnnnnn

OK:
```

The "OK:" prompt indicates that PRIMOS II is ready to accept command input.

PRIMOS may now be booted (see section on starting up PRIMOS below).

Manual Boot with Manual Entry of the Physical Device Number

The following startup technique is rarely required. However, if it is necessary to manually enter the physical device number of the bootstrap device, type in:

CP> BOOT 114 (or BOOT 134, if booting from controller 1)

The system responds with:

PHYSICAL DEVICE=

Type in the physical device (partition) number of the command device. (This number should be written on the cover of the disk pack.)

The system then responds with:

```
PRIMOS II Rev. 19.0 10/13/81 (AT 170000)
STARTING UP DISK nnnnnn

OK:
```

The first line gives the revision number of PRIMOS II, the date of the revision, and the restart location of PRIMOS II. The second line gives the command device number just specified by the operator. PRIMOS II now starts up the command device.

The "OK:" prompt indicates that PRIMOS II is ready to accept command input.

PRIMOS may now be booted (see section on starting up PRIMOS below).

Starting Up PRIMOS

If PRIMOS has not been automatically booted, use the following procedure to boot PRIMOS once PRIMOS II is running. If PRIMOS is being started for the first time and is located in the UFD PRIRUN, or if PRIMOS is being restarted from the directory previously used, issue the command:

OK: PRIMOS

If operating system files are not located in the directory PRIRUN or in the boot directory that was used previously, you must either indicate a pathname that is a top-level UFD or attach to the appropriate directory prior to giving the command RESUME PRIMOS.. For example, to start up PRIMOS when operating system files are located in the directory OPSYS, type:

OK: PRIMOS OPSYS

Once the PRIMOS command has been entered with a pathname, the pathname is retained and the command PRIMOS is sufficient for subsequent starts from the same directory.

If the pathname is not to be retained for reuse, use this method:

OK: A OPSYS
OK: R PRIMOS

SYSTEM STARTUP ON MACHINES WITHOUT VCP

Some earlier Prime computers are equipped with control panels instead of virtual control panels. This section describes the procedure for starting up the system from a storage module disk or a cartridge module disk (CMD) using a control panel.

The Initial Procedure

The first three steps in booting the system are:

1. Turn the CPU front panel rotary selector switch to STOP/STEP.

2. Press the MASTER CLEAR switch.

3. Turn the ADDRESS/DATA switch to ADDRESS.

The additional front panel switch settings and procedures are described below.

Note

On machines with VCP, if the supervisor terminal fails to print any message after the machine has been turned on, this is an indication that the machine is hung. Type CONTROL-P to allow the control panel to examine and print data.

Automatic Boot (non-VCP)

In the standard procedure for booting PRIMOS, the peripheral device is started automatically and PRIMOS is automatically brought up.

4. If booting from controller 0, set sense switches 4, 5, 10, 13, and 14 up ('14114). If booting from controller 1, set sense switches 4, 5, 10, 12, 13, and 14 up ('14134).

5. Turn the rotary switch to LOAD.

6. Press START.

7. Turn the rotary switch to RUN.

8. Reset all sense switches.

The system responds:

```
PRIMOS II REV. 19.0 10/31/81 (AT 170000)
STARTING UP DISK nnnnnn
A PRIRUN
CONFIG
```

PRIMOS then boots.

Note

This automatic booting procedure may be used only with a Rev. 19 top partition on a device.

Manual Boot with Automatic Device Startup (non-VCP)

The following startup technique should be used when operations in PRIMOS II are required. The partition's physical device number will automatically be entered. The procedure is:

4. If booting from controller 0, set sense switches 4, 10, 13, and 14 up ('10114). If booting from controller 1, set sense switches 4, 10, 12, 13, and 14 up ('10134).

5. Turn the rotary switch to LOAD.

6. Press START.

7. Turn the rotary switch to RUN.

8. Reset all sense switches.

The system will respond:

```
PRIMOS II Rev. 19.0 10/13/81 (AT 170000)
STARTING UP DISK nnnnnn

OK:
```

The "OK:" prompt indicates that PRIMOS II is ready to accept command input.

PRIMOS may now be booted (see section on starting up PRIMOS, non-VCP, below).

Manual Boot with Manual Entry of the Physical Device Number (non-VCP)

This startup technique is rarely required. However, if it is necessary to manually enter the physical device number of the bootstrap device, follow this procedure:

4. If booting from controller 0, set sense switches 10, 13, and 14 up ('00114). If booting from controller 1, set sense switches 10, 12, 13, and 14 up ('00134).
5. Turn the rotary switch to LOAD.
6. Press START.
7. Turn the rotary switch to RUN.
8. Reset all sense switches.

If the bootstrap is successful, the system will respond with:

```
PHYSICAL DEVICE=
```

Enter the physical device (partition) number of the command device. The system will respond:

```
PRIMOS II REV. 19.0 10/31/81 (AT 170000)
STARTING UP DISK nnnnnn

    OK:
```

The OK: prompt indicates that PRIMOS II is ready to accept command input.

PRIMOS may now be booted (see section on starting up PRIMOS, non-VCP below).

Starting Up PRIMOS (non-VCP)

Once PRIMOS II is running, PRIMOS may be started. If PRIMOS is being started for the first time (and is located in the UFD PRIRUN) or is being restarted from the directory previously used, use the PRIMOS command:

OK: PRIMOS

If operating system files are not located in the directory PRIRUN or the previously used boot directory, you must either indicate a pathname that is a top-level UFD or attach to the appropriate directory prior to giving the command. For example, to start up PRIMOS when operating system files are located in the directory OPSYS, type:

OK: PRIMOS OPSYS

Once the PRIMOS command has been entered with a pathname, the pathname is retained and the command PRIMOS is sufficient for subsequent starts from the same directory.

If the pathname is not to be retained for reuse, use this method:

OK: A OPSYS
OK: R PRIMOS

WHAT HAPPENS DURING SYSTEM CONFIGURATION

Two files usually residing in the directory CMDNC0, the CONFIG and C_PRMO files, are of particular importance at system startup time. These files are created by the System Administrator; they are appropriately protected and should never be modified without the prior approval of the Administrator.

The CONFIG file holds the data for system configuration. C_PRMO is a command file that invokes CONFIG, sets system parameters, and performs a number of other operations necessary to configure the system.

After the PRIMOS command is given, the operating system attaches to UFD CMDNC0 and looks for the file C_PRMO.

Note

It is strongly recommended that the CONFIG and C_PRMO files permanently reside in the directory CMDNC0. The CONFIG file may have any legal name. In the example below, it is named "CONFIG".

If C_PRMO exists, the system responds with:

```
GO
CONFIG -DATA CONFIG
PRIMOS 19.0

nnnnK BYTES MEMORY IN USE
```

The C_PRMO file is started. The first command of the C_PRMO file invokes the CONFIG file.

If C_PRMO does not exist, the system will print the message:

```
PLEASE ENTER CONFIG
```

Commands must then be entered from the supervisor terminal. If commands are to be entered manually, specific instructions regarding system configuration must be obtained from the System Administrator.

Whether commands originate at the supervisor terminal or are stored in a C_PRMO file, the first command must be the CONFIG command. (Details on the CONFIG command can be found in the System Administrator's Guide).

An Example of a CONFIG File: The first action C_PRMO takes is to invoke the CONFIG file. This is an example of a CONFIG file. The symbol "/*" indicates an explanatory comment that appears in the file. All numbers are in octal.

```
/* CONFIG DATA FOR SYSTEM SYB 19.0 01/16/82
ASRATE 1010           /* Set baud rate of supervisor terminal
TYPOUT YES            /* Print the following CONFIG commands at
                      /*  supervisor terminal
AMLTIM 2 3410 1130    /* Set variable event timers
NTUSR 76              /* Set number of terminal users
TYPOUT NO             /* Do not print the following CONFIG
                      /*  commands at supervisor terminal
COMDEV 4463           /* Specify command device
PAGDEV 11060          /* Specify paging device
MAXPAG 10000          /* Set number of pages of physical memory to
                      /*  validate at system startup
NAMLC 2               /* Set number of assignable AMLC lines
NPUSR 25              /* Set number of phantom users
NSLUSR 20             /* Set number of slave users
NRUSR 25              /* Set number of remote users
NUSEG 360             /* Set number of virtual memory segments
                      /*  available
SMLC ON               /* Enable SMLC
NET ON                /* Specify that network is to be configured
LOGREC 0              /* Turn event logger on
LOGLOG YES            /* Enable LOGIN without LOGOUT
LOUTQM 74             /* Set forced logout interval
DISLOG NO             /* Set disconnect logout option
```

```
LOGMSG NO                 /* Suppress printing of LOGIN/LOGOUT
                          /*  messages at supervisor terminal
/*   BUFFER SIZES FOR LINES AT 9600 BAUD ARE NOW:
/*     INPUT BUFFER = 256 CHARS
/*     OUTPUT BUFFER = 1024 CHARS
/*     DMQ BUFFER  = 128 CHARS
AMLBUF  3   0  1000  200          /* Set terminal I/O buffers
AMLBUF  4   0  1000  200          /* Set terminal I/O buffers
AMLBUF  5   0  1000  200          /* Set terminal I/O buffers
AMLBUF  6   0  1000  200          /* Set terminal I/O buffers
AMLBUF  7   0  1000  200          /* Set terminal I/O buffers
AMLBUF  10   0 1000  200          /* Set terminal I/O buffers
AMLBUF  11   0  1000  200         /* Set terminal I/O buffers
AMLBUF  12   0  1000  200         /* Set terminal I/O buffers
AMLBUF  13   0  1000  200         /* Set terminal I/O buffers
AMLBUF  14   0  1000  200         /* Set terminal I/O buffers
AMLBUF  15   0  1000  200         /* Set terminal I/O buffers
AMLBUF  16   1500  1000   200     /* Set terminal I/O buffers
                                  /*  (OA/WP TERMINAL)
AMLBUF  17   0  1000  200         /* Set terminal I/O buffers
AMLBUF  20   0  1000  200         /* Set terminal I/O buffers
AMLBUF  21   0  1000  200         /* Set terminal I/O buffers
AMLBUF  22   0  1000  200         /* Set terminal I/O buffers
AMLBUF  23   0  1000  200         /* Set terminal I/O buffers
AMLBUF  24   0  1000  200         /* Set terminal I/O buffers
GO                                /* End CONFIG data file
```

The final line of the CONFIG file is always GO. This causes an automatic return to the command file C_PRMO.

An Example of a C_PRMO file: The command file C_PRMO is built by the System Administrator. An example of C_PRMO is given below.

In some installations the operator may be called upon to modify the C_PRMO file. For specific information regarding system configuration, see the System Administrator's Guide.

C_PRMO always begins with the command line

```
CONFIG -DATA filename
```

where "filename" is the name of the data configuration file. After this are command lines that start up devices, configure terminal lines, share memory segments, and perform other actions necessary to make the system functional. This is done both by invoking individual commands (command lines beginning with ADDISK and other PRIMOS operator commands) and by invoking other command input files (command lines beginning with CO).

Following is an example of C_PRMO:

```
CONFIG -DATA CONFIG       /* Invoke CONFIG file "CONFIG"
ADDISK 460 32060 71061    /* Start up local devices
```

```
COMO -NTTY                    /* Turn off terminal output
AMLC TTY 0 2213               /* Configure AMLC line
                              /*  (300 baud telephone line)
AMLC TTY 1 2213               /* Configure AMLC line
                              /*  (300 baud telephone line)
AMLC TTY 2 2213               /* Configure AMLC line
                              /*  (300 baud telephone line)
AMLC TTY 3 2413               /* Configure AMLC line
AMLC TTY 4 2413               /* Configure AMLC line
AMLC TTY 5 2413               /* Configure AMLC line
AMLC TTY 6 2413               /* Configure AMLC line
AMLC TTY 7 2413               /* Configure AMLC line
AMLC TTY 10 2413              /* Configure AMLC line
AMLC TTY 11 2413              /* Configure AMLC line
AMLC TTY 12 2413              /* Configure AMLC line
AMLC TTY 13 2413              /* Configure AMLC line
AMLC TTY 14 2413              /* Configure AMLC line
AMLC TTY 15 2413              /* Configure AMLC line
AMLC TTY 16 2413              /* Configure AMLC line
AMLC TTY 17 2413              /* Configure AMLC line
AMLC TTY 20 2413              /* Configure AMLC line
AMLC TTY 21 2413              /* Configure AMLC line
AMLC TTY 22 2413              /* Configure AMLC line
AMLC TTY 23 2413              /* Configure AMLC line
AMLC TTY 24 2413              /* Configure AMLC line
ASSIGN AMLC 77                /* *** PREVENT RANDOM USER FROM
                              /* CAUSING MANY INTERRUPTS ***
ELIGTS 1                      /* Set Eligibility Time Slice
OPR 1                         /* Enable SHARE command (requires OPR 1)
SHARE SYSTEM>ED2000 2000            /* Share the editor (ED)
SHARE SYSTEM>S2050 2050 700         /* Bring up shared libraries
R SYSTEM>S4000
SHARE 2050                          /* Share this segment
OPR 0                               /* Disable SHARE
/*
/*  Run command files in UFD SYSTEM to do the following
/*      (all opened on file unit 7):
/*
CO SYSTEM>BASICV.SHARE.COMI 7       /* Share BASICV compiler
CO SYSTEM>COBOL.SHARE.COMI 7        /* Share COBOL COMPILER and
                                    /*  library
CO SYSTEM>C_SHAREFED 7              /* Share FED
CO SYSTEM>C_SHAREFORMS 7            /* Share FORMS
CO SYSTEM>F77.SHARE.COMI 7          /* Share F77 compiler
CO SYSTEM>EMACS.SHARE.COMI 7        /* Share EMACS
CO SYSTEM>MIDAS.SHARE.COMI 7        /* Share MIDAS library
CO SYSTEM>PL1G.SHARE.COMI 7         /* Share PL1G compiler
CO SYSTEM>SPL.SHARE.COMI 7          /* Share SPL
CO SYSTEM>FTS.SHARE.COMI 7          /* Share File Transfer Service
CO SYSTEM>POWERPLUS.SHARE.COMI 7 /* Share POWER
CLOSE 7                             /* Close file unit 7
OPR 1                               /* Enable SHARE command
CO SYSTEM>PASCAL..COMI 7            /* Share PASCAL
* CO DBMSLB>C_SHARE 7               /* Bring up DBMS
```

```
CLOSE 7                         /* Close file unit 7
OPR 1                           /* Enable SHARE command
SHARE 2176 700                  /* Share this segment
SHARE SYSTEM>ED2034 2034        /* Share this segment
R SYSTEM>E4000 1/7
OPR 1                           /* Enable SHARE command
R SYSTEM>Q4000 1/11
SHARE 2026                      /* Share this segment
*
SHARE SYSTEM>S$2167 2167        /* Shared SPOOL$ subroutine
                                /*  library
COMO -TTY                               /* Turn on terminal output
A CMDNC0
CLOSE 7                                 /* Close file unit 7
OPR 0                                   /* Disable SHARE
ADD -ON SYS SYSSYS COCOA VANIL          /* Make remote disks on
ADD -ON SYA SYSSYA OSPRY IUDGR1         /*  SYA SYD SYE
ADD -ON SYM SYSSYM OSCR1 BBRD           /*  SYG SYI
ADD -ON SYD SYSSYD TRNS MRGFY           /*  SYM SYO SYP SYQ
ADD -ON SYE SYSSYE IUDGR1 IUDGR2 FTING  /*  SYS SY.CAD
ADD -ON SYG SYSSYG AAAA BBBBB CCCCCC    /*  RES.X1 RES.X2
ADD -ON SYP SYSSYP                      /*  available to users
ADD -ON SYQ SYSSYQ                      /*  of local system
ADD -ON SYA SPOOLA EUA LAPIS LJFHJ IOADHJ/*"        "
ADD -ON SYA QW1 QW2 QW3                 /*      "        "
ADD -ON SYI SYSSYI QW5 QW7              /*      "        "
ADD -ON SYO SYSSYO MYHRR FRCNSE         /*      "        "
ADD -ON SY.CAD SYSCAD                   /*      "        "
ADD -ON RES.X1 RESRCH                   /*      "        "
ADD -ON RES.X4 KYHJKJ                   /*      "        "
BATCH -START -RLV 2 -TS 99              /* Bring up BATCH monitor
FTOP -START_MNGR                        /* Start FTS manager
FTOP -START_SRVR FTP                    /* Start FTS server
COMO -END                               /* Close this COMO file
M ALL -NOW                              /* Inform all users that
System SYB is up and running!           /*  the system is up
CO -END                                 /* End command file
```

<u>COMINPUT files within C_PRMO</u>: Several COMINPUT files will be invoked when the above example of C_PRMO is run. Such files are built by the System Administrator.

The COMINPUT files invoked by the above example are located in the directory SYSTEM. Two examples of such files are:

```
/* C_SHAREEMACS, EMACS, MAM, 07/20/81
/* Share EMACS editor
/* Copyright (c) 1981, Prime Computer, Inc., Natick, MA 01760
OPR 1
SHARE SYSTEM>EM2141 2141
SHARE SYSTEM>EM2142 2142
SHARE SYSTEM>EM2143 2143
SHARE SYSTEM>EM2147 2147
```

```
OPR 0
CO -CONTINUE 6
CO -END

/* MIDAS.SHARE.COMI, MIDAS, LSH, 06/10/81
/* SHARE & INITIALIZE MIDAS FROM THE SUPERVISOR TERMINAL
/* Copyright (c) 1980, Prime Computer, Inc., Natick, MA 01760
/*
/*
OPR 1
SHARE SYSTEM>K2014A 2014
SHARE SYSTEM>K2014B 2014
R SYSTEM>K4000 1/2
SHARE 2020 700
OPR 0
R SYSTEM>IMIDAS
/*
CO -CONTINUE 6
CO -END
```

Additional System Startup Commands

To make the system fully operational, additional actions must be taken after C_PRMO is invoked. These actions are largely defined by what is (or is not) contained in the files CONFIG and C_PRMO. Generally, an operator may expect to perform the following operations:

- Enter time and date. Only then will users be able to log in.
- If the system is on a network, add to the system any remote disks not started in C_PRMO.

Other commands may be given at this time, depending on your system's startup procedure. The System Administrator should provide you with guidelines as to exactly what additional actions need to be taken. Some possibilities are indicated below.

Setting the Time and Date: On many systems, the SETIME command is a part of the C_PRMO command file. Thus, the time and date are set during the running of C_PRMO. If this is the case, a prompt will appear telling you to enter appropriate time and date information at the supervisor terminal.

Other systems may not have SETIME included in C_PRMO. A reason for this is that some subsystems require time and date to be set before monitors begin operation. The Batch monitor, BATCH_SERVICE, is an example of this (see Chapter 11).

If SETIME is not a part of C_PRMO, the SETIME command must be entered at the supervisor terminal. To set the system date and time, issue the command:

    SETIME -mmddyy -hhmm

where mmddyy are digits that represent the month, day, and year and hhmm are digits that represent the time in hours and minutes.

For example:

    OK, SETIME -110282 -1530

The system responds with the message:

    Login please.

at all user terminals. Users may then log in.

Adding Remote Disks to the System: Remote partitions not started by C_PRMO may be added to the system with the ADDISK command:

    ADDISK diskname-1 [diskname-2...diskname-9] -ON nodename

For example, to add the disks PAMPA and STEPPE, located on remote system FIELDS, enter:

    OK, ADDISK PAMPA STEPPE -ON FIELDS

These disks are now connected to the system.

COMINPUT or CPL files may also be invoked to add remote disks. For example, to run command input file REMOTE in UFD SYSTEM, enter:

    OK, CO SYSTEM>REMOTE

The previously written command file would then take over, adding specified disks.

For more information on working with remote disks, see Chapter 12.

Starting the Batch Monitor: In the above example of the C_PRMO file, the Batch monitor is started by the BATCH -START command in C_PRMO. The monitor, however, will not actually be started until the system time and date have been set with the SETIME command. For information on the BATCH subsystem, see Chapter 11.

Starting the File Transfer Service (FTS): In the above example of the C_PRMO file, the FTS servers were started with the commands FTOP -START_MNGR and FTOP -START_SERVER FTP. These servers should be started up only after the system date and time have been set with the SETIME command. For information on the File Transfer Service, see Chapter 12.

Additional Actions: Some additional tasks you may have to undertake at the time of system startup are:

- Issuing the command CHAP, to grant particular operations special priorities and time slices.
- Invoking MAXUSR, if it is necessary to place special limits on the number of users the system can accommodate.
- Starting special phantoms if needed.

Operator commands are described in detail in Chapter 15.

PRIMOS Messages at System Startup

During the startup procedure, PRIMOS prints messages at the supervisor terminal when certain requested operations have been performed. (These messages do not occur immediately after the requested operation.)

The File Access Manager: If your system is running FAM II, there will be no FAM messages. Messages that will appear on systems running FAM I are listed in Appendix L.

The Batch Subsystem: This message will be transmitted if the Batch subsystem has been started:

```
Monitor started up.
```

This will soon be followed by the message:

```
Monitor in operation.
```

The File Transfer Service: When the command FTOP -START_SERVER FTP is given, the system responds with the message:

```
[FTOP rev 1.0]
Server ftp phantomed, - user number is xx.
```

When the server has been successfully phantomed, the system responds with the message:

```
*** FTP (user xx on NODENAME) at hh:mm
hh.mm.ss: FTS server - FTP started up on day, date, year
```

When the command FTOP -START_MNGR is given, the system responds with the message:

```
[FTOP rev 1.0]
Manager YTSMAN phantomed, - user number is xx.
```

When the FTS manager has been successfully phantomed, the system responds with the message:

```
*** YTSMAN (user xx on NODENAME) at hh:mm
[YTSMAN rev 1.0] started.
```

Other Messages: Certain other messages may appear at system startup. Your system's message output at startup time will reflect changes made by the System Administrator to the C_PRMO command file. If messages appear with which you are unfamiliar, note them in the system logbook and ask the System Administrator about their meaning.

If Your Supervisor Terminal is a VDU

On systems where a Video Display Unit (VDU) is being used as a supervisor terminal, a COMOUTPUT (or COMO) file is used to store system messages. Commands that open this file should be placed in the C_PRMO file, immediately after the CONFIG command. The first three lines of such a C_PRMO file would read as follows:

```
CONFIG -DATA CONFIG                      /* Specify CONFIG file
COMO SYSTEM.COMO -CONTINUE -TTY          /* Continue COMO to file
STATUS SYSTEM                            /* Print system information
```

where SYSTEM.COMO is the name of this COMO file.

Note

If the disk partition on which CMDNC0 resides is an ACL partition, user SYSTEM must be given at least ALURW access to CMDNC0. Users must have LUR rights.

It is the responsibility of the System Administrator to maintain and edit the command output file so that it does not become so large that it is unmanageable. If you notice that the COMO file is becoming too large, notify your System Administrator.

The system COMO file is supported by a 384-character buffer provided to store all messages for User 1. All messages sent to User 1 by other processes are directed to this buffer.

If messages come in faster than they can be taken from the buffer, subsequent messages will be lost. (Multiple disk error messages could cause such a condition to occur.) PRIMOS indicates this with the message:

```
User 1 message buffer overflowed. nn characters lost.
```

SYSTEM SHUTDOWN

Before shutting down the system, it is good operational procedure to inform local users and operators at other PRIMENET nodes (if any) by using the MESSAGE command (see Chapter 15).

The shutdown procedure is as follows:

1. Send shutdown messages to inform local users and operators of any remote systems that a system shutdown is anticipated. Give users as much advance notice as is possible.

   Local users are contacted with MESSAGE ALL. Remote systems are contacted with the command format: MESSAGE -usernumber -ON nodename. For example, to contact all users on system SYB, and to contact the operators of systems SYH, SYD, and SYE, the command format would be:

   ```
   OK, MESSAGE ALL -NOW
   SYSTEM GOING DOWN IN 5 MINS. PLEASE LOGOUT

   OK, MESSAGE -1 -ON SYH
   SYB GOING DOWN IN 5 MINS.
   OK, MESSAGE -1 -ON SYD
   SYB GOING DOWN IN 5 MINS.
   OK, MESSAGE -1 -ON SYE
   SYB GOING DOWN IN 5 MINS.
   ```

2. Prevent new users from logging into the system with the MAXUSR command. Setting MAXUSR to zero prevents additional users from logging in, but does not interfere with users currently on the system.

   ```
   OK, MAXUSR 0
   ```

3. Give all users a final warning message.

   ```
   OK, MESSAGE ALL -NOW
   *** SYSTEM GOING DOWN IN 1 MINUTE *** (bells)
   OK,
   ```

Warning bells may be invoked by typing CONTROL-G.

Messages sent regularly may be run from command input files, for convenience.

4. Stop the Batch subsystem with the BATCH -STOP command to allow the Batch monitor to log itself out gracefully.

```
OK, BATCH -STOP
[BATCH rev 19.0]
Stop request issued.

*** BATCH_SERVICE (user nnn on SYSTEMNAME) at hh:mm
Operator stop.
```

When the message "Operator stop." is received, the monitor has logged out and the Batch subsystem is no longer running. Batch jobs that were running at the time of the BATCH -STOP command, however, continue running.

5. Stop the File Transfer Service server phantoms with the command FTOP -STOP_SRVR FTP to allow FTS to log itself out gracefully. Failure to do this may invalidate the FTS database.

```
OK, FTOP -STOP_SRVR
[FTOP rev 1.0]
Server notified to stop.

*** FTP (user nnn on SYSTEMNAME) at hh:mm
hh.mm.ss: FTS Server - FTP closed down on day, date, year

*** FTP (user nnn on SYSTEMNAME) at hh:mm
hh.mm.ss: Server shutdown by operator.

Phantom nn: Normal logout at hh:mm
Time used:  xxh xxm connect, xxm xxs CPU, xxm xxs I/O
```

Once the FTS servers have been closed down, stop the FTS manager phantom (YTSMAN). To do this, first determine YTSMAN's usernumber with the STATUS USERS command. Then type the command line:

```
LOGOUT -usernum
```

The following message is printed:

```
Phantom nn: Normal logout at hh:mm
Time used:  xxh xxm connect, xxm xxs CPU, xxm xxs I/O
```

6. Log out all users with the command line LOGOUT ALL.

```
OK, LOGOUT ALL
```

7. Use the STATUS USERS command to make sure all users have been logged out.

```
OK, STAT USERS

User                No   Line  Devices
SYSTEM               1    asr  <SYSA> SMLC00 SMLC01 AL077
NETMAN              84    NSP  <SYSA>
```

8. When you are sure nothing is still running, shut down the system with the SHUTDN ALL command. For example:

```
OK, SHUTDN ALL
REALLY? YES
WAIT,
PRIMOS NOT IN OPERATION

HALTED AT xxxxxx/yyyyyy: zzzzzz

CP>
```

When the "REALLY?" prompt is given, any answer except "YES" aborts the shutdown.

9. Shut down equipment in the reverse order of system startup:

   - Turn off peripheral devices.
   - Turn off dataphone sets.
   - Turn off all magnetic tape drives.
   - Turn off disk drives; wait until all disks have stopped.
   - Turn off the CPU.

5 Monitoring the System

INTRODUCTION

This section contains the following:

- A discussion of commands that give the operator information about system status, followed by a reference list indicating instances in which each command is used.
- A listing of messages printed at the supervisor terminal reflecting actions of users or of the system. These actions are not initiated by the operator.
- A discussion of the event logging mechanism in PRIMOS.
- A brief discussion regarding the system logbook.

The major tools available for system monitoring are the commands AVAIL, STATUS, and USAGE.

Command	Function
AVAIL	Displays information regarding available records on a disk.
STATUS	Prints information on the status of users, devices, and the network (if any) to which the local system is attached.

USAGE	Actively monitors demands on the various components of the system.

Certain other commands are available to monitor specific aspects of the system (the spooler, batch, etc.). Brief descriptions as they pertain to system monitoring are given below. These commands are described fully in the PRIMOS Commands Reference Guide.

MONITORING SYSTEM UFDS

The directories listed below are under the control of the operator.

Directory	Description
BATCH	Contains command files to initialize Batch subsystem.
BATCHQ	Contains command files to initialize Batch database.
CMDNC0	Contains all external commands available on the system. (External commands run in the user's address space.)
DOS	Contains the single-user operating environment PRIMOS II.
FAM	Contains a file access manager (FAM I) used for remote file access to pre-Rev. 18.2 systems. Must be installed. (See Appendix L.)
FORMS*	Contains files needed to run the Forms Management System (FORMS). Must be installed. See the FORMS Programmer's Guide.
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 mounted on logical disk 0.
LOGREC*	Contains system event logging files.
PRIMENET*	Contains all files needed to run networks, including FAM II and network event logging files.
PRIRUN	Contains PRIMOS runfiles used in booting PRIMOS.

SPOOLQ	Contains spool queue and files to be spooled.
SYSCOM	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.
SYSTEM	Contains all files for shared subsystems. Also contains DISCS file, an operator-generated file which holds a list of disk partitions. The DISCS file is used by the AVAIL * command.
PL1G>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. Periodically (about once per 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).

Examples of Monitoring System UFDs

System UFDs are monitored by using the ATTACH and LD commands, as illustrated in the two examples below. (For a complete discussion of these and related commands, see the Prime User's Guide.)

Example 1:

```
OK, ATTACH SYSCOM
OK, LD

<PEGSYS>SYSCOM (LUR), Records= 100, Quota= 100 / 1000

Files= 27.

A$KEYS              A$KEYS.INS.FTN      A$KEYS.INS.PL1      A$KEYS.PL1
BDKEYS              BDKEYS.INS.FTN      ERRD.F              ERRD.INS.FTN
ERRD.INS.PL1        ERRD.INS.PMA        ERRD.P              ERRD.PL1
KEYS.F              KEYS.INS.FTN        KEYS.INS.PL1        KEYS.INS.PLP
KEYS.INS.PMA        KEYS.P              KEYS.PL1            ONCODES.INS.PL1
ONCODES.INS.PMA     ONCODES.P           ONCODES.PL1         PARM.K
PARM.K.INS.FTN      PARM.K.INS.PL1      PARM.K.PL1
```

Example 2:

```
OK, LD LIB>@@

<PEGSYS>LIB (LUR), Records= 900, Quota= 900 / 4000

Files= 17.

APPLIB.BIN      FINLIB.BIN      IFINLB.BIN      LIBEDB.SAVE
MATHLB.BIN      MSORTS.BIN      NPFINLB.BIN     PFINLB.BIN
SHARE4.BIN      SPLLIB.BIN      SPOOL$.BIN      SRTLIB.BIN
SVCLIB.BIN      VAPPLB.BIN      VMSORT.BIN      VSPOO$.BIN
VSRTLI.BIN

OK,
```

MONITORING ACCESS CONTROL LISTS

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

Command	Function
LIST_GROUP	Lists the ACL groups to which you belong. Such groups may determine access rights to certain files. 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.

ACL access rights are indicated by symbols:

Symbol	Right	Applies To	Meaning
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.

The use of these commands is illustrated below.

Monitoring ACL Group Membership (LIST_GROUP)

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, LG
Groups are:  .OPERATIONS
```

Group membership is defined by the System Administrator.

Note

User 1 is never a member of any group.

Monitoring Access Rights on a File (LIST_ACCESS)

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, LAC AAA>BBB

ACL protecting "AAA>BBB":
        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 (LU), 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.

Monitoring Priority Access Rights on a Disk (LIST_PRIORITY_ACCESS)

System Administrators, and occasionally 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! LPAC PATCH
Priority ACL on partition "<PATCH>":
        SYSTEM:    ALL
        $REST:     NONE
```

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

```
OK, LPAC PATCH
Priority ACL not found.  <PATCH> (list_priority_access)
```

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, LAC

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 System Administrator's Guide.

MONITORING DISK QUOTAS

Introduction

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 perform this function. 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 3. 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.

Examining Quotas and Current Storage

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 the partition SYS.B, type:

```
OK, LQ <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, LQ 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 will also be 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.A>UFD.1 (ALL), Records= 47, Quota= 115 / 500
                  .
                  .
                  .
```

The number following "Records= " is the number of records used by this directory. The two numbers following "Quota= " are, respectively, the number of records used by the directory and its entire subtree followed by the maximum number of records permitted for use by the directory and its subtree. If the second 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. For example, to display information for a partition, type:

```
OK, LD <SYS.A>@@ -SIZE

<SYS.A>MFD (ALL)

Files= 3.

    1         sam   BADSPT
    2         sam   BOOT
    9         sam   SYS.A

Directories= 3.

  555   1000  dir   FLOPSY
   -1      0  dir   MFD
  433   1000  dir   MOPSY
 3412      0  dir   PETER

OK,
```

In this example, the files BADSPT and BOOT contain, respectively, 1 and 2 records. The file SYS.A (the DSKRAT) contains 9 records. The directories FLOPSY and MOPSY, with quotas of 1000 records each, contain 555 and 433 records. The nonquota directory PETER contains 3412 records. The MFD has no quota information.

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
```

The wildcard may be used. For example, to display information for a partition, type:

```
OK, LD <SYS.A>@@ -DETAIL

<SYS.A>MFD (LUR)

Files= 3.

LUR      1 sam              30 Mar 82 00:01:12  BADSPT
LUR      2 sam              30 Mar 82 00:01:12  BOOT
LUR      9 sam              30 Mar 82 00:01:12  SYS.A

Directories= 3.

LUR    555 dir    0         02 Jun 82 17:45:12  FLOPSY    (Specific)
LUR    433 dir    0 dmp     02 Apr 82 12:39:56  MOPSY
ALL   3412 dir    0         02 Jun 82 13:18:32  PETER     (Specific)

OK,
```

In this example, column 1 lists access rights; column 2 lists the number of records and the file or directory type; column 3 indicates if the dump flag has been set during a backup; column 4 lists the date and time of the last modification to the file or directory; column 5 lists the file or directory name; column 6 indicates if a specific ACL exists for that file or directory.

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 = 440 words) 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:

Object	Report
file	The size of the file in 1024-word records (440-word records if -NORM is specified). The number of words in the file 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>UFD1 (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, which provides more efficient use of the disk. If, however, 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:

Argument	Definition
packname	The name of the disk
*	Represents "all started partitions"
-LDEV nn	The logical device number, where nn is represented numerically, in octal (e.g., 2, 6, 12)

If AVAIL is given without arguments, information will be printed for the device currently attached to. 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. 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      TOTAL        FREE    %      COMMENTS
  ID        RECS         RECS    FULL
--------------------------------------------------
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.

In normalized form, the table appears as:

```
OK, AVAIL * -NORM

VOLUME      TOTAL        FREE    %        COMMENTS
  ID        RECS         RECS    FULL      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 packname 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 (U) on the MFD and read (R) 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 disknames,in column form, that has been created with the editor. In addition to the column listing disknames, other information may be included in separate columns. For example:

- The disk's logical device number
- The disk's physical device number
- Miscellaneous information

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
LKYTHS  1    460  WRITE PROTECTED
MISCEL  3  31460
```

MONITORING USER STATUS

System usage is monitored with the internal command STATUS. While this command is available to both operator and user, when given from the supervisor terminal the command provides some information not available at a user terminal. The format for this command is:

STATUS [argument]

The arguments and the descriptions of their display as seen from the supervisor terminal are:

Argument	Description
ALL	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.
DEVICES	Prints physical device number, user-id, user number, and logical device number of all currently assigned magnetic tape devices.

DISKS	Prints partition name, logical and physical device numbers, and nodename of all currently started disk partitions.
ME	Prints information on all users. Identical to STATUS USERS.
NETWORK	Prints information regarding the status of the full duplex, ring, and public data networks.
PROJECTS	Prints information regarding the project status and user number of all currently logged-in users.
SEMAPHORE	Prints all semaphores, their values, and their users.
SYSTEM	Prints the version of PRIMOS in operation and, if the command is given from the supervisor terminal, the amount of physical memory being used.
UNITS	Prints user-id, system name, and local nodename for each currently open file unit.
USERS	Prints usernumber, line number, and all partitions and assigned devices in use by each terminal currently logged into the system.

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 below.

```
OK, STAT ALL

System is currently running PRIMOS rev. 19.0

4096K bytes memory in use                                          I


User SYSTEM                              TEK1                      II

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

Device User name                          Usrnum Ldevice
MT0    BAOBAB                             22     MT0

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

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

Full duplex network

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

Ring network
                                                                   VI
      Node       State
    TEK1         ****
    TEK2          Up
    MKTG          Up
    MNFG.A        Up
    MNFG.B        Up
    RES1          Up
    RES2         Down
```

```
Public data network

     Node
   ATHNS                                                           VII
   CNBER
   RIO.A
   SNGPR

Pagdev = 11060 Comdev = 4463                                       VIII

User                        No  Line Devices
SYSTEM                       1  asr <PLEIST> SMLC00 SMLC01 AL077
SYSTEM                       5    3 <PLEIST>
ASH                          7    5 <PRECAM>
YGDRSL                       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 )       IX
ELEMEN                      65   33 KANSAN<PRECOL>
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 <ORDOVI> <PLEIST> PR0
```

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

Section	Information
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 nodename, if any. (Here, the nodename 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, i.e., 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 D). 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 and 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 device (Pagdev) 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.

<u>Note</u>

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 <u>usually</u> the line number plus 2. Column 3 (Line) is the AMLC line number of the user terminal (octal). Specially assigned (non-AMLC) line numbers are:

Line	Meaning
asr	User is the supervisor terminal using the USRASR command

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 initial attach point or current UFD, or if the user has opened any files 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 disks are shown by DISK physical device, and assigned AMLC lines by AL line 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 SYSTEM PERFORMANCE

USAGE is a system metering tool that allows operators and users to monitor several performance factors of PRIMOS's operation. Both manual and automatic sampling modes are available.

The command format is:

```
USAGE [options]
```

Options may be selected in any order from the list below:

Option	Meaning
-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 n seconds (n must be an integer in the range 1 to 32767). It is recommended that n 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 n sets of data have been printed. n 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.

Manual Sampling

If manual sampling is desired, do not specify the -FREQ option or the -TIMES option. USAGE will be invoked each time a START command is issued, printing the most recent differential values; after the sample is taken, USAGE will pause and go to the command level, allowing other commands to be entered.

It is recommended that manual sampling times be not less than 30 real seconds. No options are permitted with START.

An example of the command format for manual sampling is:

```
OK, USAGE -BRIEF
[USAGE 19.0]
Type "START" to continue.

OK, START
```

Automatic Sampling

For automatic sampling, the -FREQ option must be specified and the -TIMES option may be specified. An example of automatic sampling is:

OK, USAGE -FREQ 1800 -TIMES 10

PRIMOS will monitor the system 10 times, with an interval of 1800 seconds (30 minutes) between each sampling, for a period of 5 hours (1800 seconds x 10 times / 3600 seconds per hour). The long form of data display appears in the following format:

```
[USAGE 19.0]

 10/01/81 14:40:01.20  DTIME=   29.82  CPTOT= 9272.42  IOTOT= 2279.54
                                          CP=    6.15    I/O=    4.44

   %CPU    %IDL1   %IDL2    %ERR     %IO    %OVLP    IO/S     PF/S
  10.30    90.89   81.54    1.97    14.87   0.00     7.78     3.99

   %CLK    %FNT    %AML     %MPC    %PNC    %SLC    %GPPI    %DSK
   1.57    0.03    1.07     0.00    0.13    0.00     0.00    0.23

  LOCATE   %MISS    %FND   %SAME  %SHARE    LOC/S    LM/S
   1308    6.35    84.17    9.48    0.00    43.86    2.78

   DISK  QWAITS  %QWAIT  DMAOVR  %DMAOV    HANGS   %HANG
    232      0    0.00       0    0.00       0     0.00

 USR LOGNAM     MEM    CPTIME     DCP      %CP     IOTIME     DIO      %IO

   1 SYSTEM     662   528.035   0.225    0.755   161.476   0.294    0.986
   4 PEGSYS      36    36.712   0.017    0.058    30.558   0.000    0.000
  11 NINA         9     9.424   0.043    0.144     1.227   0.000    0.000
  16 SRG        252    67.214   4.060   13.613    27.500   3.909   13.107
  23 SHARON     154   199.009   1.397    4.683    46.012   0.000    0.000
  32 ROO         36     1.287   0.003    0.010     1.600   0.000    0.000
  49              3     0.372   0.154    0.515     0.000  -0.500   -1.676
  84              5    25.844   0.120    0.402     0.000   0.000    0.000
  88 SYSTEM      13    75.980   0.056    0.189    13.185   0.000    0.000
  98 SHINE       24     0.480   0.075    0.251     0.442   0.000    0.000

   DISK     I/O   %I/O    TIME   %TIME

   '26       97   41.81   2.09   47.13
     0       97   41.81   2.09   47.13

   '27      135   58.19   2.35   52.87
     0      135   58.19   2.35   52.87
```

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 LOGNAM column.

USAGE SCREEN DISPLAYS

Definitions for USAGE's screen displays are listed below. All percentages are based on elapsed, or CPU, time in the last sampling interval, unless otherwise stated.

Definitions of System Meter Displays

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

Display	Definition
DTIME	The number of real seconds elapsed between the previous sample time and the current sample time.
CPTOT	The number of CPU seconds charged to all user processes since cold start.
IOTOT	The number of I/O (disk) seconds charged to all user processes since cold start.
CP	The number of CPU seconds charged to all _user_ processes in the current sampling interval.
I/O	The number of I/O (disk) seconds charged to all _user_ processes in the last sampling interval.
%CPU	The percentage of real time during which CPU time was charged to _user_ 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.

%ERR	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 process has been overcharged with respect to CPU time.
%IO	The percentage of DTIME 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.
IO/S	The I/O (disk) request rate in operations per second, over the last sampling interval.
PF/S	The page fault frequency in faults per second, over the last sampling interval.
%CLK	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.
%AML	The percentage of CPU time used by the AMLC 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.
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 (i.e., the percentage of LOCATE misses).

%FND	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.
%DMAOV	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.

Display	Definition
USR	The usernumber.
LOGNAM	The first six characters of the user-id.
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.
CPTIME	The CPU time, in seconds, used by this user since login.
DCP	The CPU time, in seconds, used by this user during the last sampling interval.
%CP	The percent of total CPU time used by this user during the last sampling interval.
IOTIME	The I/O (disk) time, in seconds, used by this user since login.
DIO	The I/O (disk) time, in seconds, used by this user during the last sampling interval.
%IO	The percent of realtime (over the last sampling interval) during which I/O (disk) was in progress for this user.

Definitions of Disk I/O Displays

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

Display	Definition
DISK	The octal controller I/O address and the disk drive unit number.
I/O	The number of disk I/O operations for that controller or unit in the last sampling interval.
%I/O	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.
%TIME	The percentage of disk I/O time spent performing I/O on that controller or unit during the last sampling interval.

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.

MONITORING THE SPOOL QUEUE

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

SPOOL -LIST[arguments]

Arguments that may accompany SPOOL -LIST are:

Argument	Action
ALL	Lists all files on the current system.
DEFER	Lists deferred files.
FORM type	Lists files queue with FORM specified by type. (For default type specify ' ').
OWN	Lists files spooled under user's user-id.
PLOT	Lists files in plot queue.
PRINT	Lists files in print queue.

If no option is requested, the output is the same as for SPOOL -LIST ALL.

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 19.0]

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

See Chapter 9 for more details on the spooler.

MONITORING THE NUMBER OF USERS

The total number of system users (terminal, phantom, and remote users, but not including SYSTEM) can be obtained with the internal command USERS. The dialog is:

```
OK, USERS

USERS= 23
```

SYSTEM AND NETWORK EVENT LOGGING

An event logging mechanism is provided to record information about significant system events (cold/warm starts, machine checks, disk errors, etc.) in an internal buffer, periodically dump this buffer to a disk file, and format and print the disk recording file. Similarly, a network event logging mechanism is provided for systems that are connected to networks. Both logging mechanisms are controlled at system startup time by the CONFIG directives LOGREC and NETREC (see 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 Appendix J for a list of system event logger messages, and Appendix K for a list of network event logger messages.

The EVENT_LOG Command

The command EVENT_LOG is used to turn system or network event logging off or on. The command format is:

$$\text{EVENT\_LOG [-NET]} \begin{bmatrix} \text{[-ON]} \\ \text{[-OFF]} \end{bmatrix}$$

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. Turning system 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.

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

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

Interrogating the Event Logging File

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. The command line to invoke LOGPRT is as follows:

```
LOGPRT [outtreename] [options]
```

Note

If the System Administrator has not included LOGPRT as an external command in CMDNC0, LOGPRT can be invoked by the command line:

```
R TOOLS>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.)

Option	Meaning
-HELP	Prints a list of LOGPRT options. The LOGPRT command must be reissued after the list of options is printed.
-CENSUS	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.
-CONTIN	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.
-DELETE	Deletes the output file (after spooling). This option is ignored if TTY is specified for outtreename. May only be used with the -SPOOL option.
-DUMP	Dumps each entry in octal.
-FROM mmddyy [hhmm] TODAY	Prints entries from the specified date mmddyy or today's date to the latest entry.
-INPUT [pathname]	Specifies the pathname of the input system or network event logging file to be processed as pathname. 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 mm/dd/yy 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).
-NET	Processes the input network event logging file. Must precede all other options.

-PURGE	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.
-REMARK	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.
-SPOOL	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 outtreename.
-TYPE 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:

Type	Meaning
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
SETTIM	SETTIME command issued
QUIET	Quiet machine check mode

REMARK	Operator message
BADENT	Bad entries not of types 0-20

Network Event Types: Network event types that may be specified with LOGPRT -NET's -TYPE option are:

Type	Meaning
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 dims 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 start
POWERF	Power fail checks
INCREQ	Incoming call requests for FAM debug
OUCREQ	Outgoing call requests for FAM 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
BADENT	Bad entries not of types listed above

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
```

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: CHMMRA 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: M.PYTHN ON 001064
20:06:28 MON 10 MAY 1982
DISK MOUNT: M.PYTHN ON 001064
20:07:28 MON 10 MAY 1982
DISK MOUNT: M.PYTHN 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

TYPE   NUMBER
COLD   2
WARM   3
TIMDAT 13
SHUTDN 2
DSKNAM 14
***** END OF FILE -- 21 ENTRIES, 21 PROCESSED *****
OK,
```

COMMANDS REFERENCE LIST

Following is a list of system parameters and values whose status the operator may wish to know. The command to print that information (and usually more) is given. Check the individual command for specific details.

Status Item	PRIMOS Command
Access groups	LIST_GROUP
ACL protection	LIST_ACCESS
Active Batch jobs	JOB -STATUS or JOB -DISPLAY
AMLC (user) line	STATUS USERS
Assigned devices, user	STATUS USERS
Assigned mag tape drives	STATUS DEVICES
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 units in use	STATUS UNITS
Free records	AVAIL
Line, user (AMLC)	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
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 [ALL]
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 (AMLC)	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

SUPERVISOR TERMINAL MESSAGES

In addition to messages printed at the supervisor terminal in response to operator-initiated actions (e.g., bringing up the spooler), other messages are printed to inform the operator of system status changes.

Note

If the supervisor terminal is a Video Display Unit (VDU), messages should be saved in a COMOUTPUT file. The COMOUTPUT file should be opened as part of the system startup; see Chapter 4 for instructions on how to do this.

Typical informative messages appearing at the supervisor terminal are:

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

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

- TIMED OUT

PRIMOS has logged out an inactive user.

- 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 NODENAME) 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 H. They are easily identified, since they all begin with *BATCH*.

- DISK xx ER - other information.

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

- DISK xx ERROR - other information.

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

MONITORING THE FILE TRANSFER SERVICE (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. These tasks are explained 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. You can always expect the directory to contain 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 users' files that are being transferred. Thus, these log files should be periodically reviewed and archived to offline storage.

THE SYSTEM LOGBOOK

The System Administrator will establish a logbook for each system in the computer room. Although the precise design of the logbook will vary from one computer facility to another, logbooks generally are a repository for the recording of all events that have a bearing on system operation. As an operator, maintaining this logbook will be one of your responsibilities. Therefore, you should familiarize yourself with your site's logbook procedures as soon as possible. Logbook entries should always be signed.

A logbook should contain sufficient information about system operation and history to allow a thorough analysis of any unusual or undesirable occurrences. Such unusual operational events include: system crashes, hardware changes, operator errors, or external happenings such as power failure. Logbook records should be adequately detailed so that the administrator or operator can restore all or part of the system to normal status. In extreme cases, the logbook may serve as a reference for a Systems Analyst. Thus, as a general rule: When in doubt, enter it in the logbook.

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 in the process of reading, writing or updating information, it must conform to your system's software addressing method. Bringing a disk into conformance with your system's requirements is called "formatting" a disk. Formatting a new disk is often referred to as "creating" a disk. Additionally, your System Administrator may often decide it is advisable to divide a disk's total 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 will format both user disks (those areas of the disk pack utilized for the actual storage of a user's work) and paging disks (those areas of the disk used for the temporary storage of data when paging occurs). MAKE will create and structure any form of disk storage supported by PRIMOS. After a disk is formatted, 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 all directories and files on the partition.
- The BOOT file, used in bootstrapping the disk.

- 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 name of the partition.
- If badspots exist, a badspot file (BADSPT), used to indicate the location of any physical defects on the disk.
- The directory CMDNC0, containing the run modules for PRIMOS commands.
- The directory DOS, containing the run module for PRIMOS II.

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 must determine the number of the physical disk (or partition of the disk) that will be added to the system. The physical disk number gives the system the following information about the disk: the type of storage device being used, the drive unit on which the disk is mounted, and, for partitions, the size of the partition and its location on the disk pack. This physical device number must be added to the table of assignable disks with the DISKS command at the supervisor terminal.

Physical disk numbers may be determined by referring to the tables in Appendix D.

Procedure

The following actions should be taken before running MAKE:

1. Determine the physical disk number (see Appendix D).
2. If you are working in PRIMOS, enter the DISKS command from the supervisor terminal.
3. To prevent accidental erasure of data on a disk because a physical device number was mistyped, use the following procedure:
    - Under PRIMOS, only the disk to be created by MAKE should be assigned to the terminal.
    - Under PRIMOS II, all running disks should be write protected except the disk to be created by MAKE. (Most disk drives have a switch labeled 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.

4. If you are working in PRIMOS, add the device to the table of assigned disks with the DISKS command. This must be done at the supervisor terminal. The format for this command is:

   DISKS physical-device-number

   Then, assign to your terminal the disk to be formatted. To do this, use the command ASSIGN DISK plus the physical device number. The format for this command is:

   ASSIGN DISK physical-device-number

RUNNING MAKE

The MAKE utility is initialized with the MAKE command. The operator is then called upon to respond to a number of prompts, each of which is explained below. The MAKE command format is:

MAKE [-options]

The options are:

Option	Description
-AUTO	Automatically verifies badspots. There is no verification query in the user dialog.
-OLD	Creates a pre-Revision 19 badspot file. This option must not be specified when MAKE is used in conjunction with diskettes or any fixed-disk medium.

The MAKE Procedure

After MAKE is initialized, a dialog will ensue. YES/NO queries are answerable by YES, OK, or NO. (These terms may not be abbreviated.)

1. Initialize the MAKE utility: Type the command MAKE and any desired option, generating the response:

   OK, MAKE -AUTO
   MAKE <Rev. 19.0>

2. Enter the physical disk number: MAKE now asks you to enter the physical disk number (see Appendix D).

   PHYSICAL DISK: physical-disk-number

MAKE computes the number of records on the disk pack from the disk number.

3. Specify the storage device: MAKE must now determine the storage device being used. The following prompts will appear in the indicated order until a YES response is entered:

   ```
   Please answer YES or NO to the following questions.
   STORAGE MODULE OR CMD?
   34 MB FIXED MEDIA?
   68 MB FIXED MEDIA?
   160 MB FIXED MEDIA?
   600 MB FIXED MEDIA?
   40 MB STORAGE MOD?
   ```

4. Split/do not split the disk: When the size and type of the storage device have been determined, MAKE asks:

   ```
   SPLIT DISK?
   ```

   Answer NO if all or none of the disk is to be used for paging.

   If only part of the disk is to be used for paging, or badspot information must be maintained for the paging partition, answer YES. This will cause a record of badspots on paging partitions to be kept in the badspot file.

   MAKE will ask for the number of records that are to be reserved for paging:

   ```
   PAGING RECORDS (DECIMAL):
   ```

   The optimal number of paging records for your system is a function of system use, but in all cases most records should be reserved for this use. A maximum of 20 to 25 records for the file system partition is all that is needed for the badspot file. The actual size of the partition is a decision generally made by the System Administrator.

   MAKE then prints the disk number, file records, and paging records at the terminal:

   ```
   DISK        FILE-RECORDS   PAGE-RECORDS(DECIMAL)
   xxxxxx      yyyyyy         zzzzzz
   ```

5. Check the parameters: MAKE then asks:

   ```
   PARAMETERS OK?
   ```

   If the number is correct, type YES. If not, type NO, and MAKE will return to the PHYSICAL DISK: question (see step 2).

6. Name the partition: You will now be asked to name the partition. MAKE asks:

   PACK NAME?

   Packnames have the same naming requirements as filenames, but may be a maximum of six characters long. Enter a valid filename. This will serve as the name of both the disk pack and the file containing the disk record availability table (DSKRAT). Make sure each disk pack has its own unique packname.

   Notes

   The name of the DSKRAT file (packname) may be changed later, by the CNAME command, or with ADDISK's -RENAME option.

   For split disk paging devices, the packname must be PAGING or ALTPAG unless the command device and the paging device are the same.

   For non-split disk paging devices, the packname may be any legal filename except PAGING or ALTPAG.

7. Set the baud rate: Once a valid packname has been entered, MAKE asks:

   BAUD RATE (DEFAULT = 300)?

   Enter the baud rate of the supervisor terminal to set the character transmission speed for the new disk. Valid baud rates are: 110, 300, 1200, 9600. Enter a carriage return to set the baud rate to the default value of 300.

8. Identify any badspots: You will now be asked to identify the location of any badspots (parts of a disk pack that cannot hold data) so that a record of their locations can be written. This file will be used by the utility commands FIX_DISK, PHYSAV, and COPY_DISK, and will allow the use of disk packs even though they contain badspots. MAKE asks:

   BADSPOTS ON DISK?

   If there are no known badspots on the disk, type NO. MAKE will create a BADSPT file if badspots are encountered.

   If badspots have been identified on the disk, answer YES. You will then be asked to identify the locations of the badspots. On some disk packs, this information is recorded on the pack's inside 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 record numbers. If a list exists, the operator must respond to the BADSPOT question with YES. MAKE then prints:

```
ENTER AS RECORD BASED BADSPOTS?
```

A NO response will cause badspot mapping to be done by tracks (as described below) rather than by records. As a result, if a track contains a single badspot, the entire track will be marked in the BADSPT file as being bad.

A YES response will cause badspot mapping to be done by records. This is a more efficient method as it allows optimal utilization of disk space. Thus, in most cases, the response to this question should be YES, indicating that individual records are to be mapped (leaving the remaining records on the track available for use).

The badspot record dialog now asks you to enter the record number of each known badspot:

```
RECORD NUMBERS ARE OCTAL
USE A RECORD NUMBER OF 0 TO END INPUT.
RECORD =
```

Enter the octal number of each badspot. Enter a 0 to end the dialog.

Note

If a NO response was given to the question "ENTER AS RECORD BASED BADSPOTS?", the dialog will be:

```
TRACK=
```

Respond by typing the track (decimal) of the first badspot. MAKE then prints:

```
HEAD=
```

Respond by typing the head (decimal) of the first badspot. This dialog continues, allowing the operator to record the track and head of all badspots on a given disk.

To terminate the badspot dialog, input 0 after both the TRACK= and HEAD= prompts. MAKE then prints a list of the badspot HEAD and TRACK numbers at the terminal.

9. Check the badspot parameters: This step will be ignored if the response to the BADSPOTS ON DISK? prompt (step 8) was NO. If the disk does contain badspots, MAKE will ask:

    PARAMETERS OK?

    Check the list of badspot locations printed by MAKE. If the listing is correct, type YES. MAKE will now write 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.

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

    Note

    PRIMOS supports a maximum of 16 badspots on paging partitions. If the sum of the badspots on the primary paging partition (PAGDEV) and the alternate paging partition (ALTDEV) exceeds 16, the partition may not be used for paging.

10. Initialize/verify/format the disk: MAKE then asks:

    VIRGIN DISK?
    VERIFY DISK?

    A virgin disk is one that has never been formatted. All formatting data is written to it during the MAKE operation.

    If the operator answers YES to the first question, MAKE writes the records of the disk or partition. A worst case test pattern is also written in the data partition. There is no need to initialize records if this has been done by a previous run of MAKE; however, it is strongly recommended that the operator answer YES to the VIRGIN DISK? question at each invocation of MAKE. This is because a YES response to both the VIRGIN and VERIFY questions allows the operator to find out immediately if there is any problem in the file structure part of the disk pack.

    If the operator answers NO, MAKE does not initialize the records but writes the BOOT file, the DSKRAT file, and the MFD, and then proceeds to verification.

    Unless the disk is a storage module, MAKE immediately begins disk initialization. For a storage module MAKE asks:

    FORMAT DISK?

A NO answer begins disk initialization. A YES answer causes the disk tracks to be correctly formatted. At the start of this process the following message will appear:

    BEGINNING FORMAT

On machines without a VCP, the current track number is displayed on the control panel lights during this process if operating under PRIMOS II. Upon completion, MAKE prints the following message and begins disk initialization:

    FORMAT COMPLETED

MAKE creates and writes to the disk the bootstrap (BOOT), the DSKRAT, and the MFD. If requested by the operator (with a YES response to the VIRGIN DISK? prompt), MAKE initializes all remaining records on the disk. At the start of this process it prints:

    BEGINNING WRITE

and upon completion:

    WRITE COMPLETE

If the disk was not specified as a virgin, this step is omitted.

If the verification question was answered YES, MAKE prints the message:

    BEGINNING VERIFY

MAKE reads every record in the file system part of the disk or partition to verify that each record can be read. On early model machines (without VCP) the track number being processed is displayed in the data lights. Upon completion of the verification procedure, MAKE prints the message:

    VERIFY COMPLETE

If the answer to VERIFY DISK? was NO, MAKE will skip this procedure. However, as most badspots are identified during the VERIFY stage it is always advisable to answer YES to the VERIFY DISK? query.

When verification has finished, the MAKE utility has completed its function. MAKE prints:

    DISK CREATED

and returns the operating system prompt (OK, if in PRIMOS; OK: if in PRIMOS II).

Special Messages from MAKE

- CANNOT HANDLE BADSPOTS ON RECORDS<16. (MAKE)

A badspot has been found on records 0-15. Records 0 to 15 contain the bootstrap and badspot file and cannot contain any badspots. A disk with flaws in these locations is not usable by PRIMOS. MAKE aborts.

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

A disk write error has occurred. MAKE retries writing the record nine times. An error message is printed for each unsuccessful write attempt. 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.

- ILLEGAL DEVICE NUMBER (MAKE)

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

- INVALID, RETYPE LAST PAIR

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

- LOST RECORDS xxxxxx

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

- NOT IN THIS PARTITION, IGNORED

The operator has specified as a badspot a record that is not within the defined limits of the partition.

- READ ERROR, RECORD = record-number

A read attempt was unsuccessful. If the -AUTO option was not specified, MAKE then asks:

    ADD TO BADSPOT FILE?

If you respond by typing YES, MAKE prints the location of the badspot and modifies the BADSPT file and DSKRAT appropriately.

- UNRECOVERED ERROR

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

WHAT TO DO AFTER RUNNING MAKE

After the disk has been created, the operator should unassign it 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
```

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

If PRIMOS is to be bootstrapped from this disk, use COPY or FUTIL to copy the UFD DOS on a master disk to UFD DOS on the newly created disk. The BOOT file expects these files to be in UFD DOS in order to bootload 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 these files.

EXAMPLES OF DISK FORMATTING SESSIONS

Following is an example of the MAKE command with no options specified, illustrating badspot detection:

```
OK, DI 4465
OK, ASSIGN DI 4465
OK, MAKE
***MAKE***  <Rev. 19.0>

PHYSICAL DISK: 4465
Please answer YES or NO to the following questions.
STORAGE MODULE OR CMD? YES
SPLIT DISK?: NO
DISK  FILE-RECORDS PAGE-RECORDS (DECIMAL)
004465      140773          0
PARAMETERS OK? OK
PACK NAME? PARSLY
BAUD RATE (DEFAULT = 300 BAUD)?  <CR>
BADSPOTS ON DISK? NO
VIRGIN DISK?YES
VERIFY DISK?YES
FORMAT DISK?YES
BEGINNING FORMAT
FORMAT COMPLETED
BEGINNING WRITE
WRITE COMPLETE
BEGINNING VERIFY
VERIFY COMPLETE
READ ERROR, RECORD = 000000242213
ADD TO BADSPOT FILE? YES
DISK  RD ERROR   004465 000001 042213 000000 000000 110000 000012
UNCORRECTABLE
TRACK =    485 HEAD =      16
     1 LOST RECORDS
DISK CREATED
OK, UNASSIGN DI 4465
OK, DI NOT 4465
OK,
```

The example below shows the MAKE command with the -AUTO option, illustrating badspot detection:

```
OK, DI 71064
OK, ASSIGN DI 71064
OK, MAKE -AUTO
***MAKE***  <Rev. 19.0>

PHYSICAL DISK: 71064
Please answer YES or NO to the following questions.
STORAGE MODULE OR CMD? YES
SPLIT DISK?: NO
DISK  FILE-RECORDS PAGE-RECORDS (DECIMAL)
071064       29628            0
PARAMETERS OK? OK
PACK NAME? SGE
BAUD RATE (DEFAULT = 300 BAUD)?  <CR>
BADSPOTS ON DISK? NO
VIRGIN DISK? YES
VERIFY DISK? YES
FORMAT DISK? YES
BEGINNING FORMAT
FORMAT COMPLETED
BEGINNING WRITE
WRITE COMPLETE
BEGINNING VERIFY
VERIFY COMPLETE
READ ERROR, RECORD = 00000042112
TRACK =    485 HEAD =    16
DISK  RD ERROR   071064 000000 042112 000000 000000 110000 000012
UNCORRECTABLE
     1 LOST RECORDS
DISK CREATED
OK, UNASSIGN DI 71064
OK, DI NOT 71064
OK,
```

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

```
OK, DI 71064
OK, ASSIGN DI 71064
OK, MAKE -AUTO
***MAKE***  <Rev. 19.0>

PHYSICAL DISK: 71064
Please answer YES or NO to the following questions.
STORAGE MODULE OR CMD? YES
SPLIT DISK?: NO
DISK  FILE-RECORDS PAGE-RECORDS (DECIMAL)
071064       29628            0
PARAMETERS OK? OK
PACK NAME? RSMRY
BAUD RATE (DEFAULT = 300 BAUD)?  <CR>
BADSPOTS ON DISK? YES
ENTER AS RECORD BASED BADSPOTS? YES
RECORD NUMBERS ARE OCTAL.
USE A RECORD NUMBER OF 0 TO END INPUT.
RECORD = 456
RECORD = 567
RECORD = 4332
RECORD = 44556
NOT IN THIS PARTITION, IGNORED.
RECORD = 443
RECORD = 34
RECORD = 0
VERIFY DATA? YES
RECORD = 00000000456
RECORD = 00000000567
RECORD = 00000004332
RECORD = 00000000443
RECORD = 00000000034
PARAMETERS OK? OK
BEGINNING FORMAT
FORMAT COMPLETED
BEGINNING WRITE
WRITE COMPLETE
BEGINNING VERIFY
VERIFY COMPLETE
DISK CREATED
OK, UNASSIGN DI 71064
OK, DI NOT 71064
OK,
```

The example below shows the MAKE command with the -OLD option, illustrating badspot detection:

```
OK, DI 71064
OK, ASSIGN DI 71064
OK, MAKE -OLD
***MAKE*** <Rev. 19.0>

PHYSICAL DISK: 71064
Please answer YES or NO to the following questions.
STORAGE MODULE or CMD? YES
SPLIT DISK?: NO
DISK   FILE-RECORDS PAGE-RECORDS (DECIMAL)
071064      29628              0
PARAMETERS OK? OK
PACK NAME? THYM
BAUD RATE (DEFAULT = 300 BAUD)?  <CR>
BADSPOTS ON DISK? NO
VIRGIN DISK? YES
VERIFY DISK? YES
FORMAT DISK? YES
BEGINNING FORMAT
FORMAT COMPLETED
BEGINNING WRITE
WRITE COMPLETE
BEGINNING VERIFY
READ ERROR, RECORD = 00000042112
ADD TO BADSPOT FILE? YES
DISK  RD ERROR   071064 000000 042112 000000 000000 110000 000012
UNCORRECTABLE
TRACK =    485 HEAD =      16
     9 LOST RECORDS
DISK CREATED
OK, UNASSIGN DI 71064
OK, DI NOT 71064
OK,
```

The example below illustrates the MAKE command with the -AUTO and -OLD options:

```
OK, DI 71064
OK, ASSIGN DI 71064
OK, MAKE -OLD -AUTO
***MAKE***  <Rev. 19.0>

PHYSICAL DISK: 71064
Please answer YES or NO to the following questions.
STORAGE MODULE or CMD? YES
SPLIT DISK?: NO
DISK  FILE-RECORDS PAGE-RECORDS (DECIMAL)
071064       29628             0
PARAMETERS OK? OK
PACK NAME? ALLSPC
BAUD RATE (DEFAULT = 300 BAUD)?  <CR>
BADSPOTS ON DISK? NO
VIRGIN DISK? YES
VERIFY DISK? YES
FORMAT DISK? YES
BEGINNING FORMAT
FORMAT COMPLETED
BEGINNING WRITE
WRITE COMPLETE
BEGINNING VERIFY
VERIFY COMPLETE
DISK CREATED
OK, UNASSIGN DI 71064
OK, DI NOT 71064
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 classifications of backups, full and incremental. A full backup is one where all specified files are copied onto offline storage. An incremental backup is one where only files modified since the last backup are copied to offline storage.

If desired, 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.

Benefits of backing up include:

- Aiding in maintaining the integrity of user data
- Providing recovery in case of loss of online data

- Giving the user the ability to restore earlier versions of files or programs

- Allowing inactive files to be removed from the system without losing them

File losses may be major (many files or entire disk of data lost) or minor (one or a few files lost).

Causes of major losses include:

- Physical damage to disks due to head crashes, being dropped, fire, etc.

- Operator error (copying from the target disk back to the source disk, thereby losing all the originally copied information; running MAKE on a disk that contains current data; etc.)

Causes of minor losses include:

- User deletion or truncation of the wrong file by mistake (the most common cause)

- Power failure during write operations which may cause pointer mismatch, discrepancies with DSKRAT file, etc.

Scheduling

The exact scheduling of backups depends upon the installation. Decisions regarding backup procedures and scheduling are made by the System Administrator, who considers how often files are changed, what equipment is available, and how important current information is. Once the backup schedule has been established it must be rigidly maintained to ensure the maximum protection of data.

Two Examples of Backup Scheduling

One development system at Prime consists of a P750, a mag tape drive, and two 300MB disk packs. As the maintenance of up-to-the-minute information is critical, all backups done on this system are full backups. On Monday evenings, both 300MB modules are backed up onto storage modules; on Wednesdays, they are backed up onto magnetic tape; on Fridays, both storage modules are again backed up, this time using a second set of backup disk packs.

The use of two separate sets of backup disks provides coverage between the weekly magnetic tape backups, and functions as an additional safety factor.

After each disk-to-disk backup, FIX_DISK (a disk "clean-up" procedure) is run. (See Chapter 8 for a discussion of FIX_DISK.)

Each weekend the entire system is copied onto magnetic tape. Weekly tapes are kept for two months; each month's first set of tapes is kept for two years. This degree of backup protection is probably well in excess of that needed by the average system.

A less intensive (and less expensive) procedure is followed at another Prime installation. Here all backups are done onto magnetic tape, mid-week backups are all incremental, and a complete copy is done only one weekend per month. This procedure saves time and expense, and takes fewer tapes.

Note

> To protect the system data in the event of a fire, some level of backup should be stored away from the computer room, preferably in another building or in a fireproof vault.

BACKING UP: DISK OR MAGNETIC TAPE?

Backups may be disk-to-disk (using the COPY_DISK command) or disk-to-tape (using the MAGSAV or PHYSAV commands). Each method offers particular advantages, as described below.

Disk-to-Disk Advantages

The advantages of disk-to-disk backups are described below.

- This method provides a generally faster data transfer rate and less operator intervention. Typically, when the system is configured for one user, a fully used 300MB disk pack can be copied in about one hour. Smaller devices would require proportionately less time. This may be contrasted with backups done on magtape. Using MAGSAV, backing up the same 300MB disk pack would take about an hour and a half (with tape density set at 1600 bpi), and would require the operator to mount and unmount nine reels of tape. Using PHYSAV with tape density set at 6250 bpi, data transfer would take about 45 minutes and would require the use of two and one-half tape reels. Additional time would be required if data were to be restored to a disk pack.

- Individual files on a disk can be accessed rapidly for recovery by PRIMOS using the directory tree structure.

Disk-to-Tape Advantages

Following is a discussion of the advantages of disk-to-tape backups.

- This backup method provides a generally lower relative cost. Although it takes about nine magnetic tapes (at 1600 bpi) to back up a full 300MB module using MAGSAV, the cost of these tapes is substantially less than the cost of another module. Additionally, disk-to-tape backups are useful for a small system, which may have a very limited number of disk packs and/or disk drives.

- Magnetic tapes can be stored under a wider range of environments than storage modules. Disk packs must be handled more carefully than tapes since the mechanical tolerances of the module surfaces, with respect to head alignment, are exacting.

DISK-TO-DISK BACKUP (COPY_DISK)

COPY_DISK is an external command that copies and verifies data from one physical device (disk or partition) to another. After COPY_DISK is used, FIX_DISK should be run for badspot handling and to ensure file system integrity.

Note

> Although COPY_DISK can be used under the PRIMOS II operating system, operation under PRIMOS is the recommended procedure. This chapter assumes that COPY_DISK is run under PRIMOS.

COPY_DISK is invoked by the command:

```
COPY_DISK [options]
```

The options are:

Option	Function
-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.

-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 P750 and P850 will slow down the disk copying operation.

What to Do Before Running COPY_DISK

To ensure a system's efficient operation with minimum inconvenience to users during the COPY_DISK operation, certain procedures should be executed prior to running COPY_DISK. The specific procedure to be followed depends on whether COPY_DISK is to be run under PRIMOS or PRIMOS II.

COPY_DISK operations should be conducted under PRIMOS except when:

- The disk pack containing the command device (COMDEV) is being backed up
- The disk pack containing the paging device (PAGDEV) must be removed

Running COPY_DISK Under PRIMOS: In this example, assume a backup is to be performed on the partition ZAPHOD (physical device number 10460). ZAPHOD resides on drive 0; the backup disk pack is on drive 1. An appropriate preparation for running COPY_DISK in this situation would be:

1. Using the MESSAGE command, contact all users to inform them that a backup procedure will be starting. This is best handled with a series of messages, the earliest of which will provide users with plenty of advanced warning:

   ```
   OK, MESSAGE ALL -NOW
   BACKUP TONIGHT AT 11:00 -  PARTITION: <ZAPHOD>

   OK, M ALL -NOW
   BACKUPS IN 1 HR. <ZAPHOD> WILL BE UNAVAILABLE FROM 11-12:30

   OK, M ALL -NOW
   BACKUPS IN 5 MIN.: <ZAPHOD> WILL BE UNAVAILABLE UNTIL 12:30

   OK, M ALL -NOW
   ** SHUTTING DOWN PARTITION <ZAPHOD> NOW!  BACK AT 12:30 **
   ```

   Users should also be warned about the shutdown of other partitions (e.g., those on the backup drive).

2. Shut down the partition being backed up (the "source," or FROM, partition):

   ```
   OK, SHUTDN 10460
   ```

3. Shut down any active partitions on drive 1; stop the drive; and remove the current disk pack.

4. Insert the new disk pack that will hold the backup (known as the "target" disk), and restart the drive. Remember that, except when backing up a cartridge module device (CMD), the target disk must be equal in size to the source disk.

5. Add the source and target disks to the table of assignable disks with the DISKS command:

   ```
   OK, DISKS 10460
   OK, DI 10462
   ```

6. Assign both disks:

   ```
   OK, ASSIGN 10460
   OK, AS 10462
   ```

The COPY_DISK utility may now be invoked.

Running COPY_DISK Under PRIMOS II: In this example, assume a backup is to be performed on the partition ZAPHOD (physical device number 10460). ZAPHOD resides on drive 0, as does COMDEV; the backup disk pack is on drive 1. An appropriate preparation for running COPY_DISK in this situation would be:

1. Using the MESSAGE command, contact all users to inform them that a backup procedure will be starting. This is best handled with a series of messages, the earliest of which will provide users with plenty of advanced warning:

   ```
   OK, MESSAGE ALL -NOW
   SYSTEM COMING DOWN FOR BACKUPS TONIGHT AT 11:00 PM.

   OK, M ALL -NOW
   SHUTDOWN FOR BACKUPS IN 1 HOUR...PLS LOGOUT BY THEN...

   OK, M ALL -NOW
   BACKUPS IN 5 MIN ...PLS LOGOUT

   OK, M ALL -NOW
   ** SHUTTING DOWN NOW FOR BACKUPS! FORCED LOGOUT COMING! **
   ```

2. Logout all users on the system by executing the LOGOUT command:

   ```
   OK, LOGOUT ALL
   ```

3. Shut down the system by executing the SHUTDN command:

   ```
   OK, SHUTDN ALL
   REALLY? YES
   WAIT,
   PRIMOS NOT IN OPERATION

   HALTED AT xxxxxx/yyyyyy: zzzzzz
   ```

4. Write protect the disk pack being backed up (the "source," or FROM disk) by pushing the PROTECT button on its drive unit.

5. Stop the backup drive (drive 1) and remove the current disk pack.

6. Insert the "target disk" (the disk pack that will hold the backup) and start its drive. Remember that, except when backing up a cartridge module device (CMD), the target disk must be equal in size to the source disk. The target disk must not be write protected.

7. When the control panel prompt appears, verify the CPU and load PRIMOS II into memory:

    CP> SYSCLR

    *** CPU VERIFIED ***

    CP> BOOT 114

The COPY_DISK utility may now be invoked.

Running COPY_DISK

To run COPY_DISK, enter the COPY_DISK command and the desired options at the terminal. Prior to the actual copying session, you must respond to several questions from COPY_DISK. These questions concern device specification, and have four purposes:

- Identifying the unit to be backed up
- Identifying the backup unit
- Clearing up ambiguities and incorrect operator input
- Verifying the operator's input

The operator must first indicate the device to be copied from (FROM), and the device to be copied to (TO). Each device must be specified by its physical device number. Information on physical device numbers is located in Appendix D.

```
FROM PHYS DISK= physical_device_number
TO PHYS DISK= physical_device_number
```

If there is any ambiguity, COPY_DISK will follow the operator's input with the questions:

```
1.5M WORD PACK?
     or
40MB STORAGE MOD?
```

The questions require a YES or NO response.

If disks/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 disks/partitions are of equal size, the operator's input is verified with the message:

```
FROM, TO, RECORDS = pdn, pdn, 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 from the operator. If the new parameters are acceptable, COPY_DISK initiates the copy operation.

A typical setup of COPY_DISK might look like this:

```
OK, COPY_DISK

COPY_DISK 19.0
FROM PHYS DISK=  460
40MB STORAGE MOD?  NO
TO PHYS DISK=  462
40MB 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
GO
COPY_DISK 19.00

FROM PHYS DISK= 460
40MB STORAGE MOD? NO
TO   PHYS DISK= 462
40MB STORAGE MOD? NO
FROM, TO, RECORDS= 0000460, 000462,        14814
PARAMETERS OK? YES

OK, * Now Copy PERSON
OK, COPY_DISK
GO
COPY_DISK 19.00
```

```
FROM PHYS DISK= 110061
TO   PHYS DISK= 110063
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 operator input. The COMINPUT file used in the above example looks like this:

```
GO
* In this example COPY_DISK is run from a COMINPUT file
        *
* This COMINPUT file copies MODO to a backup pack on MOD1
* We are copying the partitions SALES and PERSON
* Copy SALES
COPY_DISK
460
NO
462
NO
YES
* Now Copy PERSON
COPY_DISK
110061
110063
YES
* That's it!  We're done!
CO TTY
```

Caution

When COPY_DISK is being run under PRIMOS the TO disk number must not be a disk connected to PRIMOS by the ADDISK command.

Special Events During the Copying Procedure

COPY_DISK copies disk records from the FROM disk to the TO disk. When the copy is finished, COPY_DISK 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 aborts, 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 I for a complete list of status words, and Appendix F for a listing of COPY_DISK error messages.)

Handling Badspots

COPY_DISK can copy FROM a disk or partition that has badspots. To do this, COPY_DISK looks for the file BADSPT in the MFD (see Chapter 6). This file contains a record of all badspots on the disk. If the COPY_DISK command finds the file BADSPT in the MFD, it will ignore any tracks that are listed as bad in the BADSPT file. If the BADSPT file does not exist (or is not appropriately marked in the MFD), no badspot handling can take place.

COPY_DISK can copy TO a disk or partition having badspots. No information will be written on those tracks that have badspots. To use a disk containing badspots as the TO disk, do the following:

1. If the BADSPT file does not exist, specify MAKE to the badspots on both the FROM and the TO disks.

2. Perform the backup.

3. If the information backed up in step one is to be restored to its original disk, retrieve it using the COPY command. (See the section later in this chapter on restoring data from disk.)

Special Events During Badspot Handling

Following is a discussion of events which 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 F). 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, the operator receives the message:

  ```
  NO FREE RECORDS AVAILABLE ON 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.

- COPY_DISK will not handle badspots on pre-Rev. 19 partitions. When a pre-Rev. 19 partition is encountered, the operator receives 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 partition prior to its being used again.

Restoring Data from Disk (COPY)

Files, trees, and UFDs may be restored to an active disk from a backup disk by using the COPY utility. COPY operates under PRIMOS. To familiarize yourself with this command, see the PRIMOS Commands Reference Guide.

The backup disk must be physically mounted on one of the system drive units. This disk should be write-protected. After the disk is ready, the desired partition should be connected to the system by the ADDISK command at the supervisor terminal. (Use the tables in Appendix D to construct the physical device number for the partition.)

Example: The system disk, containing the two partitions SOFTWR and DBTEST, is mounted on unit 1. It is necessary to restore the tree structure of the directory DBADM in the MFD on the DBTEST partition.

The backup disk is mounted on unit 2. To determine the logical device numbers, use the STATUS DISKS command:

```
OK, STATUS DISKS

Disk    Ldev  Pdev  System
SOFTWR   0    3462
DBTEST   1   71061
DBBAK    2   71063

OK,
```

Invoke COPY, define the pathname from which information will come and to which information will go, then restore the desired files and/or directories with COPY_DISK or COPY, as appropriate. Exit to PRIMOS when restoration is completed:

```
OK, COPY <1>MFD>DBADM <2>MFDBAK>DBADM
OK,
```

The backup disk is disconnected from PRIMOS by the SHUTDN command at the supervisor terminal; it can then be dismounted and returned to storage.

Note

At Rev. 19, it is not possible to add two disks with the same name.

Using COPY_DISK with Cartridge Module Device (CMD)

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:

```
SRWREC 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 under PRIMOS II, 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.

DISK-TO-TAPE BACKUP (MAGSAV AND MAGRST)

MAGSAV and MAGRST are PRIMOS subsystems that move files from any disk including storage modules to a seven- or nine-track magnetic tape and vice versa. The files are moved logically, and may be SAM, DAM, segment directories, UFDs, or MFDs. Whenever you specify a directory, the directory and all of its components (the subtree) are transferred. Because of their wide capabilities, MAGSAV and MAGRST can be used by both operators and other system users.

A discussion of the messages generated by MAGSAV and MAGRST may be found in the Magnetic Tape User's Guide.

The MAGSAV Subsystem

MAGSAV is the PRIMOS disk-to-tape backup subsystem. It allows you to copy files and directories from any PRIMOS-supported disks and disk partitions onto seven- or nine-track magnetic tape.

Backups and Archiving

MAGSAV is used by both users and operators. By alternating full and incremental backups, the operator 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 -SUFD, if you wish.) The -INC option causes MAGSAV to check the DUMPED bit on each file before saving it. Only if the DUMPED bit is 0 will the file be saved. (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 -SUFD 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 -SUFD 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.

> **Caution**
>
> Users who are archiving their own files or directories will probably not want to use the -UPDT or -INC options, as this may prevent their files from being saved during system backups. Only users who know that MAGSAV is never used for system backups on their system should use the -UPDT or -INC options to MAGSAV.

Running MAGSAV

There are five steps to running MAGSAV:

1. Assign a tape drive and mount the tape. (The ASSIGN command is explained in Chapter 10.)

2. Invoke MAGSAV, as explained below.

3. Tell MAGSAV what you want saved, by responding to its prompts (as explained below in the section on the MAGSAV dialog).

4. Allow MAGSAV to write the tapes, mounting additional tapes if MAGSAV asks for them.

5. Remove the tapes, and unassign the tape drive.

Invoking MAGSAV

The command line format for invoking MAGSAV is:

MAGSAV [options]

Command Line Options: You may specify one or more options in any order on the MAGSAV command line. These options and their functions are:

Option	Function
-7TRK	Specifies seven-track tape format. The default is nine-track.
-LONG	Specifies a 1024-word record size. The default is 2048-word variable length records.
-P300	Specifies 512-word records. Also suppresses ACLs.

-VAR Writes variable-length records up to 4096 bytes. (This option is the default.) -VAR is useful for large files, since 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 dump. Only files and directories with a reset (=0) DUMPED switch are saved. If you do not specify -INC, 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. 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.

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.

Request	Response
TAPE UNIT:	Supply the physical or logical tape unit number (0-7).

Note

When MAGSAV encounters the physical END OF TAPE (EOT), a message appears on your terminal and a new tape unit is requested. The new unit may be the same as the old unit.

ENTER LOGICAL TAPE NUMBER: Supply 1 for the first logical tape, 2 for the second, and so on. MAGSAV rewinds the tape, then positions it correctly. If you enter the value 0, MAGSAV assumes that your tape is already positioned correctly.

TAPE NAME: Supply any name, no more than six characters long.

DATE: Supply the date in the format MM DD YY. PRIMOS checks the date and rejects it if it is not valid. If you reply to this request with a carriage return (CR), the current date is used. (Under PRIMOS, this is the default.)

REV. NO: Supply any number.

NAME OR COMMAND: Supply the name of a file or directory you wish to save, or supply an action command.

A <u>filename</u> identifies the specific file or directory you wish to save on tape. If you specify an MFD, you are identifying an entire disk to be saved. (You must be attached to the specified MFD.) If you specify an asterisk (*) you wish to save your current directory. If MAGSAV encounters an access problem during a save operation, then that file is abandoned, and MAGSAV continues with the next file/UFD (if there is one).

Possible action command responses are:

$A ufd [password] [ldisk] [key] — Changes the home UFD. If you do not specify <u>ldisk</u>, only the local disk is searched for the specified UFD. This is the default. $A does not accept pathnames.

$I [filename] n — Prints, at your terminal, an index of files and directories saved from disk to tape. This is the default. If you specify a <u>filename</u>, the index is written into that file. <u>n</u> indicates the number of levels to be included in the index. (The default is two levels.) $I does not accept pathnames.

$Q — Terminates the logical tape and returns you to PRIMOS.

$R — Terminates the logical tape, rewinds the physical tape, and returns you to PRIMOS.

$UPDT ON — Turns on update and sets the DUMPED switch for all saved files and directories. This action command is the same as the -UPDT command line option.

$UPDT OFF — Turns off update and does not set the DUMPED switch for saved files and directories. This is the default.

$INC ON — Turns on incremental dump and saves only those files and directories with a set DUMPED switch. This action command is the same as the -INC command line option.

$INC OFF — Turns off incremental dump and saves files and directories whether or not their DUMPED switch is set. This is the default.

Sample MAGSAV Session: The following example illustrates a terminal session during which a disk file, DFILE, was saved on tape. If a carriage return (CR) is given in response to the DATE and REV NO prompts as shown below, the system supplies the current date and zero Rev number. Note that, as in this example, you must supply a logical device number (ldn) as a response to the TAPE UNIT prompt.

```
OK, AS MT1 -ALIAS MT7
Device MT1 Assigned.
OK, STAT DEV

DEVICE  USRNAM  USRNUM  LDEVICE
MT1     ADLEY     50    MT7

OK, MAGSAV
REV. 19.0
TAPE UNIT (9 TRK): 7
ENTER LOGICAL TAPE NUMBER: 0
TAPE NAME: DATATAP
DATE (MM DD YY): (CR)
REV NO: (CR)
NAME OR COMMAND:   DFILE
NAME OR COMMAND:  $Q
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: If a UFD is saved, all Access Categories within the UFD are saved. They are written onto the tape before any other files or directories. Each object protected by an Access Category is followed on the tape by the information that it is so protected.

If individual files are being saved by name, then any desired category ACLs must also be saved by name. They are not saved automatically, as specific ACLs are.

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 a UFD.

THE MAGRST SUBSYSTEM

MAGRST allows you to restore information from a magnetic tape created by MAGSAV on any Prime-supported disks. This information is saved from and readily restored into the PRIMOS file system. All restore operations take place in your initial attach point. MAGRST can read tapes of any record size, with fixed- or variable-length records (up to 4096 bytes), making it compatible with MAGSAV.

Running MAGRST

The MAGRST command line format is as follows:

```
MAGRST [options]
```

Type MAGRST after the PRIMOS prompt (OK,). At this point, there are several command line options you can specify. 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 following the command line options.

Command Line Options: You may specify one or more options in any order on the MAGRST command line. These options and their functions are:

Option	Function
-7TRK	Specifies seven-track format. The default is nine-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.

The MAGRST Dialog: After you invoke MAGRST and specify any options on the command line, the MAGRST dialog begins. MAGRST requests information from you in the following order. Appropriate user responses are shown.

Request	Response
TAPE UNIT:	Supply a logical device number (0-7). If you do not specify the -7TRK option on the MAGRST command line, the default is nine-track. Note When MAGRST encounters the physical END OF TAPE (EOT), a message appears on your terminal and a new tape unit is requested. The new unit may be the same as the old unit.
ENTER LOGICAL TAPE NUMBER:	Supply a logical tape number from 1 to n (1 for the first logical tape, 2 for the second, and so on), if your tape is divided into several logical units. This positions your tape to the specified logical tape. If you enter the value 0, MAGRST assumes that your tape is already positioned correctly.

Note

A "runaway" tape condition can occur if there is only one logical tape on the currently mounted reel and you supply a number greater than 1 in response to this request. If this happens, MAGRST searches endlessly for the nonexistent logical tape(s) and is not able to read the EOT (end-of-tape) marker. You must unassign your drive to abort the unsuccessful search.

MAGRST does not have to search all logical tapes when it restores sequential ones. After MAGRST returns you to PRIMOS, the tape is not rewound. Instead, it is positioned at the location before the beginning of the next logical tape in sequence. For sequential logical tapes, run MAGRST again and supply 0 as the response to the LOGICAL TAPE NO: request. Then, the next logical tape is restored without rewinding and reading through the preceding logical tapes.

NAME: No user response is necessary. MAGRST displays the name of the logical tape at which you are currently positioned. This is the name provided during the MAGSAV dialog.

DATE: No user response is necessary. MAGRST displays the date that the tape was recorded. This is the date provided during the MAGSAV dialog.

REV. NO. No user response is necessary. MAGRST displays the number provided during MAGSAV.

REEL NO: No user response is necessary. MAGRST displays the appropriate tape reel number.

READY TO RESTORE: Supply one of the following options:

YES — Restores the entire tape and returns you to PRIMOS. If MAGRST encounters an access problem during a restore operation, then that file is abandoned and MAGRST continues with the next file/UFD (if there is one).

NO — Requests a different tape unit and logical tape. (MAGRST does not restore the previously specified tape.)

$I [filename] n — Prints, at your terminal, an index of all files and directories restored. This is the default. If you specify a filename, the index is written into that file. n indicates the number of levels to be included in the index. (The default is two levels.) $I does not accept pathnames.

NW [filename] n — Prints, at your terminal, a tape index, but files and directories are not restored. If you specify a filename, the index is written into that file. n indicates the number of levels to be included in the index. (The default value is 100.) This option is useful if you wish to determine the tape's contents. NW does not accept pathnames.

$A UFD [password] [ldisk] [key] — Attaches you to the specified UFD. Pathnames are not allowed. Supply a password, if needed. If you do not specify a number for ldisk, MAGRST searches all disks for the specified UFD. (The STATUS DISK command gives ldisk numbers.) Supply a numerical value for key to attach to a sub-UFD. If you wish to attach to a sub-UFD in the current UFD, then set ldisk equal to 0 and key to 2.

PARTIAL — Restores only certain files and directories. Supply pathnames in response to the TREE NAME: request.

TREE NAME:

This request prints only if you supply the PARTIAL option. In response to TREE NAME:, supply the pathname of the file or directory you wish to restore. In the pathname, do not include the name of the UFD from which the file or directory was saved. You may specify multiple pathnames for a partial restore. In this case, the TREE NAME: request is repeated after each restoration until you enter a null line (CR), which signals the end of restoration. For a partial or full restore, files with bad records are omitted. The pathnames of these files are printed, along with an error message. A maximum of ten treenames per restore is permitted.

After each file or directory is restored, the message FILE COMPLETE prints at your terminal. The message RESTORE COMPLETE prints when the end of logical tape is reached.

Sample MAGRST Session: The following example illustrates a terminal session during which a file is restored from tape to disk. The file (DFILE) saved in the previous MAGSAV sample session is also used in this example.

```
OK, MAGRST
[REV. 19.0]
YOU ARE NOT ATTACHED TO AN MFD
TAPE UNIT (9 TRK):0
ENTER LOGICAL TAPE NUMBER: 1
NAME: DATATAP
DATE(MM DD YY): 09-02-81
REV NO:       0
REEL NO:      1
READY TO RESTORE: PARTIAL
TREE NAME: DFILE
TREE NAME:(CR)
*** STARTING RESTORE ***
*** END LOGICAL TAPE ***
*** RESTORE COMPLETE ***
OK,
```

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.

DISK-TO-TAPE BACKUP (PHYSAV AND PHYRST)

PHYSAV and PHYRST are PRIMOS subsystems that create a disk image backup on magnetic tape and restore that image back to disk from tape. Both subsystems support the following hardware:

- Nine-track magnetic tape
- 40MB, 80MB, 160MB, 300MB, and 600MB storage module disks
- 32MB, 64MB, and 96MB cartridge module disks

Although PHYSAV and PHYRST are usually used solely by the system operator, both subsystems can be used by all other users as well.

A discussion of the messages generated by the PHYSAV and PHYRST subsystems is located in the Magnetic Tape User's Guide.

The PHYSAV Subsystem

PHYSAV copies the contents of one or more disk partitions 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 5212 and 8340 bytes.

Since no attention is paid to logical file structure, it is not possible to restore a single file. The smallest unit which 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.

Specifying Disk Partitions to PHYSAV

PHYSAV can save one or more partitions in a single logical tape. (A logical tape may be part of a physical tape; or, it may contain several reels of tape.) All partitions to be saved on one logical tape must be on the same controller and unit.

PHYSAV needs to know the physical device number of the partition to be saved. You can find this number with the STATUS DISKS command. For example:

OK, <u>STAT DI</u>

Disk	Ldev	Pdev	System
GHRKIN	0	4463	
SWEET	1	460	
SOUR	2	31460	
HLFSR	3	60460	
DILL	4	71060	

If you wanted to save the partition "GHRKIN", you would give PHYSAV the physical device number 4463.

<u>Note</u>

Several partitions may be saved at once with PHYSAV. However, doing so requires computing the new physical device number for the combined partition. It also prevents use of the RAT, and hence prevents disk compression and badspot handling.

<u>Using the RAT</u>: 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 which 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.

<u>Using FIX_DISK</u>: 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, PHYRST will not be able to restore the partition.

Invoking PHYSAV

The PHYSAV command line format is as follows:

```
PHYSAV [-TTY] [-LOWEND] [-UNMOD]
```

The options are:

Option	Function
-TTY	PHYSAV asks for the magnetic tape unit number at the terminal, even if it is running from a CPL or COMINPUT file. Do not 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 or 850.
-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.

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.

Request	Response
UNIT NO:	Supply the logical tape unit number (0-7), or you may type QUIT. (Reenter the subsystem by typing REN.)
LOGICAL TAPE:	Specify 1 for the first logical tape, 2 for the second, and so on. You may use 0 to specify the current (or next) logical tape. **Note** There is no check for the previous existence of logical tape 1; the tape will be written from BOT.
COMMENT	Supply a comment up to 80 characters long.

PHYS.DEV.NO:	Specify the physical device number of the partition to be saved (40460, for example).
USE THE RAT (YES/NO)?	YES — Saves only tracks with records in use by the file system. NO — Saves all records of all tracks. **Note** If you declare a split partition by its true physical device number, then the paging portion will not be saved (even if you do not specify the RAT option). However, if it is part of several partitions declared as one, the entire partition is saved.
40MB STORAGE MOD? 600MB FIXED MEDIA?	PHYSAV asks these questions if any of the following conditions exists: 1. The user is saving multiple partitions, and there is not enough information to distinguish the type of disk. 2. The RAT is bad, and the user is not using the RAT. 3. The partition is not a file system partition, and the user is not using the RAT.
PARAMETERS OK (YES/NO)?	YES — Begins the save operation. NO — Exits from the subsystem. You may reenter by typing REN.

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 QUIT or 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 one of the partitions 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 last physical device number, but PHYSAV will have remembered partition numbers you already entered.
- If you exit from the subsystem once the save operation has begun, you may continue from the exit point by typing S, 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 on a Prime machine.
PHYS.DEV.NO: 30462
USE THE RAT (YES/NO)? YES
PHYS.DEV.NO: 40462
USE THE RAT (YES/NO)? YES
PHYS.DEV.NO: (CR)

DISK                        HEAD OFFSET, #HEADS,  SAVING
030462   NEWSYS                  6           2      RECORDS USED
040462   OLDSYS                  8           2      RECORDS USED

PARAMETERS OK (YES/NO)? YES
REEL:       1  LOG.TAPE:     1  SECTION:      1
SAVE COMPLETE
WRITE NEXT LOG.TAPE (YES/NO)? NO
OK,
```

The PHYRST Subsystem

PHYRST restores one or more partitions saved with PHYSAV to assigned partitions of the same size. It is not necessary to restore all saved partitions in one run. PHYRST runs in R-mode under either PRIMOS or PRIMOS II. Operation under PRIMOS is strongly recommended.

A verify facility is available to verify the readability of a magnetic tape. The tape is read (as for the restore operation), but the information is not written to disk. The verify facility does not verify that all the data has been written to the tape. It only verifies that all tape blocks are readable and in sequence.

Running PHYRST

PHYRST restores assigned partitions from a magnetic tape written with PHYSAV. Partitions being restored do not have to be on the same controller or unit. However, a saved partition can only be restored to another partition of equal size. You do not have to restore all partitions saved on your tape. You can restore only one partition if you wish. You can also, at PRIMOS level, restore a command partition.

The PHYRST command line format is as follows:

```
PHYRST [-TTY] [-NO_BADS] [-UNMOD]
```

The options are:

Option	Function
-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, below. (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.

The PHYRST Dialog: After you invoke the subsystem, PHYRST responds with a series of questions. PHYRST requests information from you in the following order. Appropriate user responses are shown.

Request	Response
UNIT NO:	Supply the physical tape unit number (0-7), or you may type QUIT. (Reenter the subsystem by typing S 1000.)
LOGICAL TAPE:	Specify 1 for the first logical tape, 2 for the second, and so on. You may specify 0 for the current (or next) logical tape.
CORRECT TAPE (YES/NO)?	YES — Continues PHYRST.
	NO — Prompts you again for a unit number.
	After the requested logical tape is found, PHYRST prints out the details of the saved information.
RESTORE ALL PARTITIONS TO ORIGINAL POSITIONS (YES/NO)?	
	YES — Restores all partitions to their original positions.
	NO — Asks if each partition is to be restored as it is saved.
RESTORE PARTITION xxxxxx (YES/NO)?	
	NO — Prompts with the next partition saved.
	YES — Generates the next dialog question.
AS PARTITION:	Supply either a physical device number or type a (CR):
	pdn -- The physical device number to which the partition is to be restored.
	(CR) — Causes the partition to be restored to its original position.
PARAMETERS OK (YES/NO)?	NO — Exits you from PHYRST. (Type S 1000 to reenter.)
	YES — Begins the restore operation.

Badspot Handling

When PHYRST is restoring material onto a Rev. 19 partition (that is, a partition which has been formatted by Rev. 19 MAKE and which is running under Rev. 19 PRIMOS), it attempts to ensure that no records are written onto any badspots the disk might contain.

PHYRST does not handle badspots if any of the following conditions are true:

- You are copying the tape onto a Rev. 18 (or earlier) partition.
- You are running a Rev. 18 version of PHYRST.
- You are copying onto a disk partition which has no BADSPT file.
- You are copying either to or from a partition which has a bad BADSPT file.
- You give the -NO_BADS option when you invoke PHYRST.

How PHYRST Handles Badspots: PHYRST performs badspot handling in the following manner:

1. Before writing a record (r) onto the disk, PHYRST checks the disk's BADSPT file to see whether the target record (t) is good or bad. (Rev. 19 BADSPT files mark individual records, not entire tracks.)
2. If record t is good, PHYRST writes record r onto it.
3. If record t is bad, PHYRST looks in the DSKRAT file copied from the source disk for a free record.
4. If it finds a free record (f) available, PHYRST copies record r onto it. It then adds an EQUIVALENCE block to the BADSPT file, showing that the record originally intended for record t has been written onto record f.
5. If there is no free record available, PHYRST prints 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. (If you do continue the copy, there will be no BADSPT file left on the target disk.)

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

PHYRST runs in R-mode. Consequently, you cannot reenter it 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 QUIT during a restore operation, PHYRST restarts from the beginning of the latest section (logical tape or current reel, whichever was most recently started). Typing S always continues 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 on a Prime machine.
PARTITIONS SAVED
030462  NEWSYS
040462  OLDSYS

CORRECT TAPE (YES/NO)? YES
RESTORE OR VERIFY (RE/VE)? RE
RESTORE ALL PARTITIONS TO ORIGINAL POSITIONS (YES/NO)? NO
RESTORE PARTITION 030462 (YES/NO)? NO
RESTORE PARTITION 040462 (YES/NO)? YES
AS PARTITION: 100462
```

```
DISK                                TO BE RESTORED AS
040462  OLDSYS                                 100462

PARAMETERS OK (YES/NO)? YES
REEL:   1 LOG.TAPE:    1 SECTION:   1
RESTORE COMPLETE
RESTORE/VERIFY NEXT LOG.TAPE (YES/NO)? NO
OK,
```

8

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 G 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
- Truncate/delete defective files
- Replace a defective DSKRAT file

However, FIX_DISK has other uses as well. It can:

- 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 normal procedure accompanying COPY_DISK 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

HOW IS FIX_DISK USED?

The format for the FIX_DISK command is:

FIX_DISK -DISK pdn [options]

Note that "-DISK pdn" must be included in the command line. pdn is the physical device number of the partition on which FIX_DISK is to be run.

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
- An option that specifies conversion to the Rev. 19 format

To Specify Repair Tasks

To specify repair tasks, use the following options:

Option	Function
-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.
-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.

-DELETE_UNKNOWN_FILE_ENTRY
Eliminates all inconsistent file entries or entries of unknown type. 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. Abbreviation: -DUFE.

Use this option to remove all file types unknown to this version of FIX_DISK.

-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.
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

-MAX_NESTED_LEVEL [n]
Sets n (decimal) as the maximum depth that directories are allowed to 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.

-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.

-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:

Option	Function
-LEVEL [n]	Sets n (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_FILE	Prints filenames in all directories. Abbreviation: -FILE.

To Specify the Command Device as the Target

To perform operations on the command device, use the following option:

Option	Function
-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.

To Convert a Disk to Rev.19 Format

To convert a pre-Rev. 19 disk to the Rev. 19 format, use the following option:

Option	Function
-CONVERT_19	Converts the current partition to a PRIMOS Rev. 19 style disk. Converts the BADSPT file to the Rev. 19 format. Initializes all quota information, and disables printing of all warning/error messages related to quotas. Creates a new rev stamp. The -FIX option must be used with -CONVERT_19.

Examples of FIX_DISK

An example of FIX_DISK follows. Here, the partition SLSRCD is specified by its physical device number (21462), 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 21462 -LEVEL 0 -CMPR -FIX
***FIX_DISK*** <REV 19.0>  03/22/82 0:11

DISKPACK ID is SLSRCD

BEGIN MFD
END   MFD  48965

51849 records in partition
48965 records used
    0 records lost
 2884 records left
    0 records compressed


DSKRAT OK
FIX_DISK finished
OK,
```

FIX_DISK has identified the partition name, signified all important data regarding records; 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 data line regarding "records lost" refers to records lost due to badspots. It does not indicate file truncation or record compression.

It is quite possible, however, that FIX_DISK will find problems on the partitions with which it is working. Such problems would be indicated by an error message (a complete listing of FIX_DISK's error messages is located in Appendix G).

The next two examples illustrate FIX_DISK's error handling technique. In the first example, 0 levels of directories will be printed. The use of the -COMDEV option indicates that the disk being worked upon is the command disk.

```
OK, FIX_DISK -DISK 460 -FIX -CMPR -LEVEL  -COMDEV
***FIX_DISK*** <REV 19.0> 01/02/82 0:15

Partition name is CMDDSK

Partition not shut down properly during the previous session!
The quota system may be incorrect.
BEGIN MFD
ACL at word 4746 is not pointed at by object it points to!
ACL is deleted!
ACL at word 5555 is not pointed at by object it points to!
ACL is deleted!
The Directory Used count is bad.  It should be 142 instead of 146.
MFD>SPOOLQ

The Directory Used count is bad.  It should be 4 instead of 10.
MFD>BATCHQ

The Directory Used count is bad.  It should be 7 instead of 8.
MFD
END    MFD  13869

14814 records in partition
13869 records used
    0 records lost
  945 records left
    1 records compressed
DSKRAT UPDATED!
FIX_DISK finished
OK,
```

In the above example, the following is indicated: disk 460 is identified as CMDDSK; 2 ACLs in the MFD were deleted; the Directory Used counts in both BATCHQ and SPOOLQ were found to be in error and were 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. The -NQ option indicates that this is not a quota partition.

```
OK, FIX_DISK -DISK 460 -LEVEL 9 -NQ -CONVERT_19 -CMPR -FIX
***FIX_DISK*** <REV 19.0> 11/02/82 0:20

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

File truncated
MFD>OEDPUS>ATE

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 this example, 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 (in this case, the file was deleted) and reported; a record was freed by compressing a UFD, thus removing empty space on the partition; information regarding records was printed; and the disk's record availability table (DSKRAT) was updated. The disk was converted to the Rev. 19 format.

On a single-partition system, running FIX_DISK with the -COMDEV option would look like this:

```
OK, FIX_DISK -DISK 460 -LEVEL 1 -CMPR -FIX -COMDEV
***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 DIRECV
      END   DIRECV  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.

HOW DOES FIX_DISK WORK?

Locating DSKRAT Errors: In order to read every used physical record, 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 quota disks (disks where the directory used count is not set to zero), 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.

Handling the BADSPT File: All badspots encountered are added to the badspot file if it exists. 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.

9
Working with Printers and Plotters

INTRODUCTION

This chapter focuses on the use of printers (also called "spoolers"). The most important command associated with controlling printers is PROP, a utility that allows the operator to control spooler operations by defining and modifying the spooler phantom environment, and by starting, stopping, and monitoring the phantoms that control the spoolers.

The format of the PROP command is:

```
PROP phantom-name [option]
```

The options of the PROP command fall into two groups: those that control the environment (that is, the parameters) of a phantom, and those that control the actions of a phantom.

This section explains:

- The nature of spooler phantoms.
- How to create and modify a spooler phantom.
- How to control spooler phantoms.

SPOOLER PHANTOMS

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 printer or plotter. Each spooler phantom runs the program SPPHN.SEG under the control of one particular "environment", or parameter set. A spooler phantom drives one device.

The environment is created using the PROP command (see below). It tells the phantom which 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/plots them.

The phantom program, the environments, the queue file, and the print files are all located in the SPOOLQ UFD.

The SPOOLQ Directory

The spooler uses the SPOOLQ UFD to store files in the print queue. If multiple systems are connected over a network, a SPOOLQ directory should be created for each system.

The SPOOLQ UFD contains two types of files that are of specific interest at this point: spool queue files and environment-controlling files. A typical SPOOLQ directory might look like this:

```
<MEGASY>SPOOLQ (Owner), Records= 362, Quota= 362 / 0

Files= 17.

E.PR0          E.PR1          L.DEST          L.DFLT
L.FORM         NETWORK_INFORMATION.SPOOL      O.PR0
O.PR1          O_CSM          O_PR0           O_PR1
P.CTRL         PRT001         PRT002          PRT003
PRT004         Q.CTRL

Segment Directories= 1.

SPPHN.SEG
```

Spool Queue Files: A spool queue file holds the contents of an item on the spool queue. One such file exists for each item on the queue. The file is placed in the directory by the spooler whenever a user inputs the SPOOL command to print a file, and is eliminated (thus removing the file from the spool queue) when the file has been printed. 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.

Environment-controlling files: Some environment-controlling files are created by the operator. Others are created by the operating system. These are the files that control the various environments of the individual spooler phantoms. Each running device has its own phantom, and hence, its own environment file. In fact, a single device may have several environments stored as files in this UFD.

Filenames that affect spooler operation will always contain the name of the phantom, plus a prefix that specifies the role of the file in the spooler's operation. Environment files take their names according to the following pattern:

Filename	Function
A.phantom-name	Contains the last operator request received by the phantom phantom-name. If A.phantom-name is open, it means that the request has been received but not yet acted on.
E.phantom-name	Contains the environment definition for the phantom phantom-name.
O.phantom-name	Contains the COMINPUT file created by PROP for the phantom phantom-name. Indicates that phantom phantom-name is running.
R.phantom-name	Contains all operator requests that have not yet been received by the phantom phantom-name.
O_phantom-name	Contains a COMOUTPUT file for phantom phantom-name, listing all files spooled.

In addition to the files mentioned above, three additional files may have been created by the System Administrator to define allowable form and destination names for the system. These files, which appear in the previous example, are as follows:

Filename	Function
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 used.
L.DFLT	This file contains one destination name (maximum 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.

Other Contents of the Directory SPOOLQ: A control file, Q.CTRL, used by the queuing mechanism, is also 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.

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 from 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
```

Note

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.

Phantom Environments

An "environment" is a collection of all the decisions that have to be made concerning a spooler phantom. For example:

- What printer or plotter does the phantom control?
- What type of paper is mounted on the spooler?
- How many lines per page should the spooler print?
- How many columns per line should the spooler print?
- Should the phantom keep a COMOUTPUT file of the spooler's activities?

The value of the spooler phantom is that decisions such as those listed above, instead of being made each time a printer is started up, may be made in advance and stored in the file E.phantom-name in the SPOOLQ UFD. Usually several phantoms are created at once, so that an environment will be on record for most of the possible spooler configurations. Then, when a printer is started or when paper is changed, the appropriate phantom can be invoked quickly and easily.

At any time, the operator (or any user) can see how the current environment is set up by using the command:

    PROP phantom-name -DISPLAY

Figure 9-1 shows the -DISPLAY option used to display a default environment (that is, an environment composed entirely of default values).

The commands that create and modify these environments are explained later in this chapter, in the section on how to create or modify the spooler phantom.

How Environments Work: An Example

As many environments as are needed can be set up and kept on file in the SPOOLQ UFD for each printer. For example, printer PR0 might have four phantoms on file: phantom PR0, which uses standard paper and a standard printout format; phantom WHITE, which uses white paper; phantom SHIP, which prints shipping labels on special forms; and phantom RUSH, which uses standard paper and prints only very short jobs, thus creating a quick-service queue.

Upon arriving, the operator may:

1. Power up printer PR0.
2. Mount white paper.
3. Give the command, PROP WHITE -START.

Phantom WHITE then controls printer PR0. Later on, seeing that printer PR0 is needed to serve as a "rush" printer, 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 then running under its new environment, with phantom RUSH in control.

```
OK, PROP DEFAULT -CREATE
> DISPLAY
[PROP rev 19.0]

DEVICE: PR0
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 9-1

HOW TO CREATE OR MODIFY THE SPOOLER PHANTOM

The PROP Command's Environment-related Options

The PROP command is used to create, modify, and display the environment of a spooler phantom. The correct command format is:

```
PROP phantom-name [option]
```

The options listed below define how PROP will be used:

Option	Function
-CREATE	Sets up a new phantom environment with the name given in the PROP command. Subcommands (explained in the next section) are used to define the environment.
-MODIFY { NOW / FINISH / IDLE }	Modifies the phantom environment named in the PROP command. If -MODIFY is used on a started phantom, use the NOW, FINISH, or IDLE argument (explained below, in the section on controlling the spooler phantom) to determine when the changes will be made. As with -CREATE, the subcommands shown below define the changes to be made.
-DELETE	Deletes the phantom environment named in the PROP command. The phantom must be stopped in order to do a -DELETE.
-DISPLAY	Prints a detailed description of the specific phantom environment named in the PROP command.
-STATUS	Prints a list of the currently defined environments and indicates which ones are currently running (i.e., are being used by a phantom).

As can be seen above, the options fall into two general categories. -CREATE, -MODIFY, and -DELETE are used to construct, change, and eliminate environments. -DISPLAY and -STATUS are monitoring tools that do not affect the environments.

Defining an Environment

Spooler environments are defined by using the -CREATE and -MODIFY options. -CREATE constructs a new environment. -MODIFY makes changes to a pre-existing environment. The created or modified environment is stored in a file E.phantom-name.

The use of either option initiates PROP's environment definition mode. PROP prints an angle prompt (>) and waits for subcommands. You may now define a spooler's environment through use of the subcommands shown below:

Subcommand	Definition
COMOUT {ON OFF}	ON specifies that a COMOUTPUT file will be kept of all phantom actions. The file, named SPOOLQ>O_phantom-name, is created the first time the phantom is activated. 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 assumed by this environment. Users specify logical destinations with the SPOOL command's -AT option.
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 AMLC line will be used as configured on the supervisor terminal. (Default is PR0.)
EVFU {-ON -OFF -NAME filename}	Indicates whether or not the printer associated with an EVFU filename 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 with a 300 lpm printer/plotter without a "forms length" switch. -OFF is the default setting, and is used with printers that do not contain EVFUs (for example, standard 300 lpm printer/plotters with a "forms length" switch). With the switch OFF, the phantom driving the printer will treat the printer normally.

	-NAME filename is used when a special EVFU file, explicitly defining channels, has been constructed in advance. filename is the filename of the specific EVFU file designed to handle the particular form being used on the spooler. This file must reside in the SPOOLQ directory.
	For further details, see the section on the Electronic Vertical Format Unit, below.
FILE	Exits the environment definition mode. If CREATE was used to enter environment definition mode, the environment file E.phantom-name is created and defaults are used whenever the user did not specify a value. If MODIFY was used to enter environment definition mode, the environment file E.phantom-name is updated.
FORM synonym	Defines a paper form as synonym.
HEADER {0 | 1 | 2}	Sets the number of header pages. A setting of 2 gives a trailer page. (Default is 1.)
LARGE n	Prints files whose length in records is less than n before larger files. n may be from 0-32767. (Default is 30.)
LENGTH n	Prints n lines per page. n may be from 10-32767. (Default is 38.)
LIMIT n	Prevents printing of any files bigger than n disk records. n may be from 0-32767 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 table.
LOWER n	Looks for SPOOLQs starting at logical disk n, where n is a decimal number. n may be from 0-63. (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 is from 1-6 characters. (Default is blank.) **Note** Use of this option deletes all currently existing FORM synonyms.
PLOT {ON | OFF}	If ON, scans the queue for PLOT files. If OFF, ignores PLOT files in the queue. (Default is OFF.)
PRINT {ON | OFF}	If ON, scans the queue for PRINT files. If OFF, ignores PRINT files in the queue. (Defalt is ON.)
QUIT	Exits environment definition mode. Any parameter changes are ignored; the file E.phantom-name remains unchanged.
TYPE {0 | 1}	0 indicates a 300 lpm printer/plotter. 1 indicates a band printer.
UNDEST synonym	Deletes a logical destination named synonym.

UNFORM synonym	Deletes a paper form named synonym.
UPCASE {ON / OFF}	If ON, converts all lowercase characters to uppercase before printing. (Default is OFF.)
UPPER n	Prevents search for SPOOLQs on logical disks with numbers higher than n. n is a decimal number from 0-63. (Default is 63.)
WIDTH n	Sets the number of physical columns on a page. Used for formatting header and trailer pages. n may be from 10-140. (Default is 108.)

Notes

If you modify the environment of a stopped or hung phantom, the modified environment takes effect when the phantom is started up again.

If you modify the environment of a running phantom, modifications take place immediately if the NOW argument is used, and when the next file is printed if the FINISH argument is specified.

All subcommands except DISPLAY, QUIT, and FILE define some parameter of the environment. DISPLAY shows the environment you have created, but does not leave environment definition mode. QUIT leaves environment definition mode without updating (or creating) the new environment. FILE updates the environment and then leaves environment definition mode.

Figure 9-2 shows a terminal session which creates an environment. See Figure 9-3 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 RIPORT
> DEST SALES
> DEST MARKETING
> MESSAGE This is a sample environment.
> DISPLAY
DEVICE: PR0
PAPER:  WHITE
FORM:
  WH
  DOC
  RIPORT
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 9-2

```
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 9-3

Synonyms: The phantom environment may contain synonyms for both the form type (the PAPER and FORM subcommands) and the printer name (the DEST subcommand) which 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 a parameter that is not identical to the value currently contained by the spooler phantom. 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. If WHITE paper is mounted, and if it has synonyms REPORT, DOC, and 8_x_11, then any request with any of those names after the spooler -FORM option is eligible for printing. For example:

OK, SPOOL TESSERACT -FORM 8_x_11

prints 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.

The Electronic Vertical Format Unit: Some 300 lpm printer/plotters and band printers now use the Electronic Vertical Format Unit (rather than the forms-length switch and paper tape loop) to define a form's length. 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 to handle a particular special form, the only steps necessary are to make sure the LINES parameter is set equal to the number of lines in the EVFU file, and to issue the subcommand:

  ```
  > EVFU -NAME filename
  ```

In most cases, all necessary EVFU files will have been established in advance by the System Administrator. If such is not the case, a special EVFU file must be constructed. Instructions for building such a file are located in the System Administrator's Guide.

- The phantom is started by issuing the command:

    PROP label -START

Notes

If the power to the printer goes off, the EVFU must be reloaded by stopping and restarting the printer using PROP.

The paper must be aligned to its top of form before the environment is started with PROP.

It is particularly important to use correct paper forms when using EVFU. Also, when the SPOOL command is given, it must be accompanied by the correct -FORM option.

CONTROLLING THE SPOOLER PHANTOM

To control the printer phantoms, the operator uses another set of PROP options. Printer phantoms are normally run from the UFD SPOOLQ, and are initiated as part of the normal system startup procedure. The format is the standard format for the PROP command:

    PROP phantom-name [option]

Options that control the spooler phantom are:

Option	Function
-START	Starts the spooler. Used to start the spooler initially or after STOP.
-STOP {NOW / FINISH / IDLE}	Causes the spooler to stop printing or scanning the queue. NOW aborts the spooler immediately. FINISH stops the spooler after it finishes the file which is currently printing. IDLE stops the spooler when it has no more work to do. The spooler is logged out.

-HANG { NOW | FINISH | IDLE }	Causes the spooler to stop printing and/or scanning the queue. NOW stops the spooler immediately. FINISH stops the spooler after it finishes the currently printing file. IDLE stops the spooler when it has no more work to do. The spooler is not logged out.
-CONTINUE	Takes the spooler out of HANG mode so it can continue printing or scanning. Spooler operation recommences where it left off.
-RESTART	Restarts the spooler after it has been halted because paper has run out, paper has jammed, or similar cause. Printing of the file restarts from the beginning of the file.
-LINEUP [lines]	Used to check alignment of paper. Functions like -RESTART, except that the phantom does an implied -HANG after printing lines number of lines. (Default for lines is OFF, meaning 0.) If the alignment of the paper is correct, give the -CONTINUE command to continue execution. Otherwise, realign the paper and give another -LINEUP command.

Note

-LINEUP may be specified as part of the -START command. For example:

```
PROP PR1 -START -LINEUP 5
```

In this case, the lineup will be done on the first file eligible for printing.

-BACK	Restarts printing of the file 128-256 lines prior to the current line. Note that the spooler will not necessarily restart on an even page boundary.
-ABORT	Causes the spooler 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 to occur immediately.
-DROP	Causes the spooler 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.)

Changing the Spooler Phantom: 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 currently running spooler is a two-step process: the old phantom must be stopped, and the new phantom must be started. In the example below, phantom PR0 will be stopped and phantom WIDE will be started. If paper needs to be changed or any other physical adjustments made to the spool mechanism, these procedures would be done during the time between the stop and the start instructions.

```
OK, LOGIN SYSTEM
OK, PROP PR0 -STOP FINISH
[PROP REV 19.0]

Wait... Acknowledged.
*** SYSTEM (user 92) at 16:08
*SPOOL* PR0          — 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]

PR0              stopped
PR1              started
RUSH             started
WIDE             started
OK,
```

The login-id under which the spooler phantom is run (usually SYSTEM) is the login-id the operator must use to utilize the operator commands of the spooler. Any necessary passwords must be included at login time. If the operator has logged in under a different name and attempts to perform operator functions, the PROP program will return an error message and ignore the commands.

When the -START option is given, PROP attaches to the first available local SPOOLQ directory and creates a file R.phantom-name. There may be a brief time delay between the time the stop or start instruction is given and the actual stopping or starting of the phantom. The phantom has actually stopped when the screen display shows:

```
*SPOOL* phantom-name — STOPPING
```

Similarly, the phantom has actually started when the screen displays:

```
*SPOOL* phantom-name -- Starting
```

As illustrated in the example 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, log in to UFD SYSTEM (use the owner password). Restart the spooler phantom by typing:

   ```
   PROP printername -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 printername -DISPLAY
   ```

5. If your system uses FAM I, make sure that user FAM has had its priority set to 2. Check this with STATUS USERS. If FAM is not running at priority 2, set it using the CHAP command (see Chapter 15).

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.

10
Working with Magnetic Tape

INTRODUCTION

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 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 Chapter 7)

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

Command	Function
SETMOD	Sets user access to drives
ASSIGN	Allocates tape drives
REPLY	Communicates with user
STATUS	Gives status of system use
UNASSIGN	Releases tape drive from user

For more information regarding magnetic tape and magnetic tape drives, refer to the Magnetic Tape User's Guide.

SETTING THE MODE OF ASSIGNMENT

The SETMOD command defines how users will gain access to the tape drives. There are three assignment modes:

Mode	Description
User mode	Users can assign tape drives without operator assistance unless options require special intervention. This is the default mode.
Operator intervention mode	Users must channel all assignment requests through the system operator.
No-assignment-allowed mode	Users are not permitted to assign tape drives at all.

The SETMOD command can be issued only from the supervisor terminal.

SETMOD has three arguments, each of which corresponds to a mode described above:

$$\text{SETMOD} \quad \begin{Bmatrix} \text{-USER} \\ \text{-OPERATOR} \\ \text{-NOASSIGN} \end{Bmatrix}$$

The arguments and the assignment modes they establish are:

Argument	Mode Definition
-USER	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 (e.g., MTX, -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 usernumbers 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.

THE ASSIGN 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.

The format of the ASSIGN command is:

```
ASSIGN { MTpdn  [-ALIAS MTldn]  [options] }
       { MTX    -ALIAS MTldn [options]    }
```

Explanations of these arguments are as follows:

Argument	Description
MTpdn	Assigns a specific tape drive, pdn. Magnetic tape (MT) units are numbered from 0 to 7, inclusive. pdn is the physical device number assigned to each drive at system startup.

MTX	Assigns an available drive. Must be accompanied by -ALIAS MTldn, which assigns a number (alias) to the drive for reference purposes. Legitimate values for ldn 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:

Option	Description
-ALIAS MTldn	Allows the user to specify tape drives with logical device numbers, from 0 to 7 inclusive. Any physical device may be used as long as the drive is specified by the requested ldn. 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, etc.). Identifiers may not begin with a hyphen (-), which is a reserved character indicating the next control argument on the ASSIGN statement line.

-RINGON
-RINGOFF
: Operator intervention required. Specifies protection 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.

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 each user terminal. It allows the operator to:

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

The REPLY command format is:

$$\text{REPLY -usernum -TAPE} \left\{\begin{array}{l} \text{ABORT} \\ \text{GO} \\ \text{pdn} \\ \text{RESEND} \end{array}\right\}$$

The command must include the usernumber (-usernum) and the -TAPE argument, or an error message will result. Only one of the following listed arguments can be specified at a time:

Argument	Description
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 not 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.

Three additional REPLY options are available for the operator's use:

Option	Description
-ALL -RESEND	This repeats all outstanding (unanswered) requests on the operator's terminal.
-usernum -RESEND	This repeats all outstanding requests from the specified usernumber.
-REPEAT seconds	This allows the operator to set how frequently (in seconds) the RESEND message is repeated. The seconds parameter is a decimal number. The default repeat frequency is 180 seconds.

Note

Improper use of the REPLY command results in an error message. The error message specifies the improper command, then demonstrates correct formats. For example, the improper command:

    REPLY -7 -TAPE -MT1

results in the message:

```
"MT1" not implemented or improper use of argument.  (REPLY)
Usage:  REPLY -usernum -TAPE [RESEND | ABORT | GO | pdn]
        REPLY -TAPE RESEND
        REPLY [ -usernum | -ALL ] [ -RESEND ]
```

FULFILLING REQUESTS FROM USERS

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

Receiving User Requests

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

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

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

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

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

The user-id and usernum 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. Sending the appropriate REPLY command.

If the assignment mode is SETMOD -USER, you will have to respond only when options are requested, or when drive assignments are requested by ldn. If the mode is SETMOD -OPERATOR, you must respond to all requests.

Some Examples of How to Fulfill ASSIGN Requests

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

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

This indicates that user-id SHANIN (designated by PRIMOS as usernumber 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 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 transmitting:

```
REPLY -7 -TAPE ABORT
```

If the request can be approved, type:

```
REPLY -7 -TAPE GO
```

This indicates that SHANIN now "owns" physical device MT1.

Responding to ASSIGN MTpdn 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 Send the message: 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. Send the message: 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 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 STAT DEV command, described below.

DETERMINING THE CURRENT STATUS OF USERS AND DEVICES

The system operator can obtain a quick list of the physical devices currently in use by typing STAT DEV. The information returned might look something like this:

```
OK, STAT DEV

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

Only currently assigned ("owned") magnetic tape devices are listed.

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

```
OK, STAT US

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
```

RELEASING TAPE DRIVES

Tape drives are released with the UNASSIGN command. Either the physical device number (pdn) or logical device number (ldn) can be specified:

```
UNASSIGN { MTpdn      }
         { -ALIAS ldn }
```

In all modes, the operator can release any tape drive owned by any user. This is done by issuing the UNASSIGN command from the supervisor terminal. The operator must use the MTpdn option to unassign a user-held device; the -ALIAS ldn option can be used by the operator only when the drive to be released was actually assigned with an alias from the supervisor terminal.

Only the user who assigned an alias can use this number when unassigning a drive.

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:

Usernum	Physical Device Number	Logical Device Number
17	MT1	MT2
17	MT2 (formerly MTX)	MT0

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

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

In -OPERATOR 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.

DRIVE STARTUP AND MOUNTING TAPES

Specific procedures for starting up tape drives and mounting tapes will vary according to the model of the drive. The general procedure is:

1. Depress the POWER button. The indicator light will go on.
2. Open the drive door and place the supply reel on the upper spindle. Lock the tape in place.
3. Unwind approximately 8 feet of tape and thread it.

Notes

The tape itself must never be touched. Handle the leader (usually the first 15 feet) only.

A threading diagram is usually located on the inside of the drive door.

On self-threading models: Place tape on upper spindle, close door, and press LOAD button.

4. Manually wind tape onto take-up reel.

5. Press the LOAD button. The tape will advance to the load point.

6. Press the ON LINE button.

> **Caution**
>
> If you are using a 6250 bpi tape drive, the formatter POWER button must also be pressed. Failure to do so may hang the system.

UNMOUNTING TAPES AND DRIVE SHUTDOWN

To shut down a tape drive, the procedure is:

1. Take the unit off line by pressing the ON LINE button.

2. Press the REWIND button to rewind the tape to the load point. Press REWIND again to unload the tape.

3. Open door and remove the rewound tape (upper reel).

4. Close door and press the POWER button. Make sure the indicator light is off.

Note

Many installations may prefer to leave tape drives powered up during the operation period.

11
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

- Starting, stopping, and "pausing" the Batch monitor
- Monitoring the Batch subsystem
- Controlling users' jobs when requested to do so

The System Administrator may also request that the operator:

- Modify or add queues, using the BATGEN utility
- Deal with any problems that might affect the database, using the FIXBAT utility

CONTROLLING THE BATCH MONITOR

The Batch monitor is controlled by the BATCH command. Its format is:

    BATCH options

The options are:

Option	Function
-START	Starts the monitor.
-STOP	Stops the monitor.
-PAUSE	Tells the monitor not to start up jobs, but to finish processing jobs that are currently executing.
-CONTINUE	Takes the monitor out of a paused state allowing it to begin processing jobs again.

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.

STARTING THE BATCH MONITOR

Start the Batch monitor from the supervisor terminal by giving the command:

```
BATCH -START -RLEVEL rlv [-TIMESLICE ts]
```

This command usually is included in the C_PRMO command file (see Chapter 4), 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. Default is 1. The value for TIMESLICE must be between 1 and 99. Default is 20. (Numbers are decimal.) The options -START, -RLEVEL, and -TIMESLICE may be given in any order.

In order to keep Batch jobs at the same or lower priority than interactive jobs, it is recommended that you use the default values of 1 for the monitor's RLEVEL (or priority) and 20 for time slice. The monitor's priority and time slice represent the greatest priority and time slice any Batch job can have. Job priority and time slice can be lowered for each queue with BATGEN's RLEVEL and TIMESLICE subcommands (explained in the System Administrator's Guide).

Notes

- The Batch system runs under PRIMOS only. It cannot run under PRIMOS II.
- 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 happens when the BATCH -START command is included in the C_PRMO file), the monitor does nothing until the SETIME command is given.

Once 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 do this, 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 an already paused monitor, 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 stopped the phantom process which started the job is logged out without being signaled that the job is completed. Thus, the restarted monitor notices nothing until it does its regular check of the queues, which may take as long as 10 minutes.)

Force 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; rather 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.

Caution

Do not issue the SHUTDN ALL command or invoke SHUTDN for the partition on which BATCHQ resides while the Batch monitor is running; either action may interrupt the monitor while it is updating queue information and thus invalidate the database.

If this occurs, the database will be inaccessible to users until it is repaired. The repairs can usually be made by FIXBAT, when it is run by the BATCH -START command. If FIXBAT cannot repair the database, the System Administrator must run INIT. (Details on running FIXBAT and INIT are given in the System Administrator's Guide.)

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.
```

MONITORING BATCH

The operator monitors the status of the Batch subsystem by two methods: specifically requesting information, and reading messages spontaneously displayed by Batch.

Requesting 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 queuename for each currently executing job
- The number of queues which contain waiting or held jobs

An example of this is:

```
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]
5 batch jobs; 3 waiting or held jobs in 2 queues; 2 executing jobs.
```

Spontaneous Messages

Whenever the monitor starts 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 monitor. When the job is completed (or aborts), the monitor sends the message:

```
*BATCH* Job jobname for user username (job-id) completed.

    or

*BATCH* Job jobname for user username (job-id) aborted.
```

These messages help the operator monitor Batch usage and load without having to make inquiries.

The System Administrator can prevent the printing of these messages by changing the command "RESUME MONITOR" in the file BATCHQ>START_BATCH_MONITOR.COMI 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.) These messages are listed and explained in Appendix H.

Monitoring User Jobs with the JOB Command

Two JOB options 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 the user-id, job-id, status, external name and queue of all active (i.e., executing, held, or waiting) jobs. The JOB -DISPLAY command returns full information on all active jobs in the system.

For example, a JOB -STATUS command might return the following display:

```
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
```

By giving the job-id in the JOB -STATUS or -DISPLAY command (for example, JOB #10003 -DISPLAY), the operator can monitor a specific active job, no matter what user submitted it. (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
```

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:

```
JOB #00003 -ABORT
[JOB rev 19.0]
Job MORIA.MAP(#00003) cancelled.

OK, DATE
27 May 82 14:02:20 Thursday
```

```
***BATCH_SERVICE (user 104 on SYS.A) AT 14:03
Job MORIA.MAP for BALROG(#00003) aborted.

***BATCH_SERVICE (user 104 on SYS.A) AT 14:03
*BATCH* Executing BRIDGE for BALROG(#00004).
```

SUMMARY OF THE JOB COMMAND

The format of the JOB command is:

JOB job-id[options]

The options are as follows:

Option	Function
-CANCEL	Cancels a held or waiting job.
-ABORT	Aborts a running, held, or waiting job.
-RESTART	Terminates, then restarts a job.
-HOLD	Holds a job in the queue.
-RELEASE	Releases a held job so that it can run.
-STATUS	Displays the status of a job.
-DISPLAY	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 resource the job needs (such as magnetic tape, disk space, or the line printer) is not available. When the resource is available, the job can be released by the operator with the command:

JOB job-id -RELEASE

HANDLING BATCH QUEUES

Monitoring 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); to unblock a queue (opening it to jobs again); or to change queue defaults or parameters.

Creating Batch Queues (BATGEN)

Either the operator or the System Administrator can create from one to sixteen queues via the BATGEN command. The operator must be logged in as SYSTEM or BATCH_SERVICE, or must be a Batch Administrator. The system date and time must have been set via the SETIME command for BATGEN to be executed. This command has the form:

    BATGEN pathname

Usually, pathname will be BATCHQ>BATDEF, as the BATDEF file is the only file the Batch monitor reads in its search for queues in which to place jobs. It is possible, however, to create queues in other files and then transfer them into the BATDEF file. Do this by:

1. Typing "BATGEN pathname", where pathname is something other than BATCHQ>BATDEF
2. Doing whatever work you want within BATGEN
3. Exiting BATGEN with the command "FILE BATCHQ>BATDEF"

For an example of this, see the section on cleaning up queues, below.

> **Caution**
>
> Only BATGEN can copy new queue configurations correctly into BATCHQ>BATDEF. If you try to copy new configurations with COPY or with FUTIL, you will disturb the ACLs set on BATDEF by INIT. If this happens, users will be unable to use the Batch subsystem. (They will probably get the error message: "Insufficient access rights. BATDEF missing.") In order to remedy this situation, you must use BATGEN to recopy the desired queues into BATCHQ>BATDEF.

Once pathname 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:

ADD queuename

MODIFY queuename

DELETE {queuename | ALL}

BLOCK {queuename | ALL}

UNBLOCK {queuename | ALL}

DISPLAY {queuename | ALL}

STATUS

FILE pathname

QUIT

A queuename is an alphanumeric name of up to 32 characters. It is created by the ADD command and is the only name by which the queue may be referenced. Queuenames must conform to standard PRIMOS filename rules. The name ALL is illegal, as it would cause ambiguity in commands such as BLOCK ALL or DELETE ALL, where "ALL" means "all queues."

Note

The name of the queue has nothing to do with the queue's number, or with the order in which queues are searched for jobs. The id number (which becomes the first digit after the "#" of the job number for jobs executing from that queue) is assigned by the Batch system and reflects the order in which queues are used when they are first defined. The search order reflects the order in which the queues are created, or added to the Batch subsystem. To establish a queue as the number-one queue for searching, add it first; add the number-two queue second, and so on.

The BATGEN commands, and their subcommands, are defined on the following pages.

Command	Function
ADD queuename	Instructs BATGEN to create a new queue. If queuename is acceptable, ADD returns the message "Enter queue characteristics:", prints a prompt ($), and waits for subcommands. If queuename is already in use, BATGEN returns a fatal error message, "Queue queuename already exists." (ADD subcommands are discussed immediately following this list of commands.)
MODIFY queuename	Instructs BATGEN to modify an existing queue. If queue queuename exists, MODIFY responds "Enter queue characteristics:", prints a prompt ($), and waits for subcommands. If queue queuename does not exist, or if it is flagged for deletion, MODIFY sends a fatal error message. (MODIFY subcommands are discussed immediately following this list.)
DELETE queuename ALL	Flags an existing queue (or all queues) for deletion. The queue will accept no more jobs and will be deleted when all currently waiting jobs have been run.
BLOCK {queuename ALL}	Sets flag in status control block of an existing queue (or of all queues) to disallow submission of further jobs to the queue.
UNBLOCK {queuename ALL}	Resets flag to allow submission of jobs to a previously blocked queue (or to all queues). Default status for queues is "unblocked".
DISPLAY {queuename ALL}	Displays name, status, and characteristics of the named queue (or of all queues). Omitting the optional argument displays information for all queues.
STATUS	Shows name and status of all queues in tabular form.
FILE pathname	Modifies file named pathname to include commands given during this session. If pathname is not given, current file is modified (the usual situation).

QUIT — 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.)

BATGEN Subcommands for ADD and MODIFY

Subcommands for BATGEN's ADD and MODIFY commands are identical. Six of them—CPTIME, ETIME, FUNIT, PRIORITY, RLEVEL, and TIMESLICE—define queue characteristics. Two others—RETURN and QUIT—tell BATGEN to save or ignore the preceding subcommands. Following is a description of the ADD/MODIFY subcommands. (All numeric values must be decimal integers.)

CPTIME default maximum — Sets CPU time limits for jobs run in this queue. The default limit will be placed on any job whose user does not specify a CPTIME limit. The maximum is an absolute limit: jobs asking for greater CPTIME than the maximum will not be allowed into the queue.

The values for CPTIME are given in decimal seconds. The word NONE may also be used, to signify that no time limit is to be set. Thus, the subcommand "CPTIME 30 NONE" would cause jobs submitted without CPU limits to be limited to 30 seconds of CPU time, but would allow unlimited time to those requesting it.

The default value may exceed the maximum. For example, "CPTIME NONE 60" is a legal command. Its effect is to close the queue to jobs which do not specify CPTIME limits of 60 seconds or less, since these jobs would be given the queue's default limit of NONE and then denied admission to the queue because their CPTIME limit was greater than the queue's maximum. If you wish to demand that users define their own time limits, this is the way to do it.

As delivered, the system has default values of "NONE" for default and maximum CPTIME. Unless both CPTIME limits are explicitly given, they will be set to "NONE" when the queue is created.

(When modifying existing queues, one or both limits may be changed. In this case, the command "CPTIME default maximum" would change both values, while the command "CPTIME default" would change only the default value.)

ETIME default maximum — This subcommand sets elapsed time limits. It acts exactly as CPTIME does, except that its values are given in minutes rather than seconds. Its system defaults are both NONE.

FUNIT number — This subcommand sets a default file unit for command input for any non-CPL job in the queue which has not specified its own file unit number. Numbers range from 1 to 126. (The maximum is dependent on the number of file units set by the System Administrator.) System default is 6.

PRIORITY value — Sets the default value for a job's priority within the queue itself--that is, its priority in relation to other jobs in the same queue. Any job not specifying its own priority will be given this default value. Permissible values are from 0 to 9, with 9 being the highest priority and 0 the lowest. System default is 5. An interactive user must specify a priority less than 4 if not logged in as SYSTEM. Priorities 5-9 may be used by the operator to specify important jobs. (Note that this priority affects only the order in which jobs within a single queue are initiated. It does not determine how fast they run. Use RLEVEL and TIMESLICE to determine runtime priority.)

RLEVEL delta-value — This subcommand does not set the runtime priority for jobs in the queue. Rather, it determines the amount their priority will be lowered from the priority of the Batch monitor. (The monitor's priority is set with the -RLEVEL option of the BATCH -START command.) Delta-values may range from 0 to 7, with 0 meaning that the queue's jobs will run on the same priority as the monitor does, and 7 representing the maximum lowering. (Note that this is one value the user may not specify.) System default is 0.

PRIMOS currently allows a process to have a priority from 0 to 3. Therefore, if the Batch monitor is running at priority 3, RLEVEL values from 3 to 7 are identical. If the monitor is running at priority 1, RLEVEL values from 1 to 7 are identical.

TIMESLICE value — This subcommand sets the time slice value for jobs in the queue. A queue's time slice may be smaller than the monitor's time slice and be effective; but if it is larger, it will be ignored, and the monitor's time slice will be used for each job in the queue. (Again, the user has no control over this value.) Time slice values represent tenths of a second. These values may range from 1 to 99, but they probably should not go above 20 unless job priority is unusually high. System default is 20, equaling 2 seconds.

RETURN — Saves the new characteristics for future display and/or filing. The use of RETURN terminates the subcommand session and returns you to BATGEN command level.

QUIT — Throws away the work done at subcommand level. If you were modifying an old queue, QUIT leaves that queue unchanged. If you were adding a new one, QUIT throws away the new queue's name as well as its characteristics. If you modified anything before quitting, BATGEN asks "Queue definition modified, ok to quit?" If it does not receive an answer of "yes" (or a carriage return), it prompts you to save work with "Please return." The use of QUIT terminates the subcommand session and returns you to BATGEN command level.

BATGEN Example

In the following sample BATGEN session, two Batch queues are defined that might be useful in a university environment. The first, QUICK.QUEUE, is intended for the use of a large number of students submitting short jobs. The second queue, PAYROLL, is intended solely for the processing of a payroll. Except for the illustrative error, all input could come from a command file:

```
OK, BATGEN BATCHQ>BATDEF
[BATGEN rev 19.0]
> ADD QUICK>QUEUE
```

```
Illegal queue name.QUICK>QUEUE (BATGEN)
> ADD QUICK.QUEUE
Enter queue characteristics:
$ CPTIME 2
$ ETIME 5
$ PRIORITY 4
$ RETURN
> ADD QUICK.QUEUE
Queue QUICK.QUEUE already exists (unblocked).
> ADD PAYROLL
Enter queue characteristics:
$ CPTIME NONE
$ ETIME NONE
$ FUNIT 126
$ PRIORITY 9
$ RLEVEL 2
$ RETURN
> DISPLAY

Queue name = QUICK.QUEUE, unblocked.
Default cptime=2, etime=5, priority=4;
Maximum cptime=None, etime=None; Funit=6;
Delta rlevel=0; Timeslice=20;

Queue name = PAYROLL, unblocked.
Default cptime=None, etime=None, priority=9;
Maximum cptime=None, etime=None; Funit=126;
Delta rlevel=2; Timeslice=20;

> MODIFY QUICK.QUEUE
Enter queue characteristics:
$ CPTIME 3
$ RLEVEL 1
$ TIMESLICE 10
$ RETURN
> DISPLAY QUICK.QUEUE

Queue name = QUICK.QUEUE, unblocked.
Default cptime=3, etime=5, priority=4;
Maximum cptime=None, etime=None; Funit=6;
Delta rlevel=1; Timeslice=10;

> FILE

OK,
```

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 may also 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.COMI. As released, the command is:

```
RESUME 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. -DAYS takes a numeric argument. The most commonly used values are 0, 1, and 2. The argument 2 cleans out jobs which have been run 2 or more days ago. The argument 1 cleans out jobs which have been run at least 1 day ago. The argument 0 cleans out all finished jobs.

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. Attach to the BATCHQ UFD.
4. Resume FIXBAT, with the desired options (explained below).

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

The FIXBAT Command and its Options

The format for the FIXBAT command is:

    RESUME FIXBAT [options]

There are three options:

Option	Meaning
-DAYS n	Removes all cancelled, completed, or aborted jobs which are n or more days old from the Batch queues; sends a message to the terminal when a job has been removed. (n must be an integer between 0 and 60.) If n 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.

If FIXBAT aborts, the cause can generally be found by looking at the log file. Usually, deleting the offending file and restarting the Batch monitor (and therefore FIXBAT) is the fastest way to fix any problems.

If FIXBAT has been run by the BATCH -START command, then it 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.

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—i.e., that it aborted, was forcibly logged out, or was halted by a system shutdown or crash.)

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, i.e., 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 such that there are at least n full days between that date and the current date, noninclusive. 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. (n is the argument supplied to the -DAYS option.)

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 queuename may be blank. This occurs if the user did not explicitly specify a queue. Also, the queuename may not resemble the queuename as defined in BATGEN with regard to uppercase/lowercase mapping. 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. It is expected that 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 monitor 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 operator or System Administrator must first delete the queue and then redefine it, so that new jobs may be submitted to it.

There are three ways to remove jobs from queues:

- Run the INIT program as explained in the System Administrator's Guide. This is the fastest way to clean out queues, as it will empty all queues. (If you run INIT with the -RSTQ option, it will also wipe out BATDEF.)

  > Caution
  >
  > This process destroys active jobs along with inactive ones. To prevent wiping out jobs that have not yet run, block all queues with the BATGEN "BLOCK ALL" command and let all waiting jobs finish before running INIT.

- Delete one queue at a time, using the BATGEN "DELETE queue-name" command. This method assures the greatest continuity of Batch service, since it leaves some queues available at all times and destroys no information on active jobs.

  The monitor ignores the queue when handling job submissions. Users attempting to submit jobs to the queue are given a fatal "Queue does not exist" message. Active jobs in the queue, however, are not disturbed, but run as they would in any other queue. Only when the last job has been completed or aborted is the queue actually deleted. Then its database is deleted, the queue is removed from the BATDEF file, and the message "Queue queue-name deleted" is sent to the supervisor terminal.

  (If a queue has never had a job submitted to it, the deletion message is "Removed queue-name from BATDEF", indicating that no job information was wiped out during the deletion.)

  Once the queue has been deleted, a new queue—either identical to or different from the old one—may be placed in the BATDEF file. (If BATDEF originally contained less than sixteen queues, the new queue could be added before the old one is deleted or flagged for deletion; but the name of the new queue could not be identical to the name of the old one. To conserve the name: delete the old queue, wait for the monitor to remove the queue from BATDEF, and then add the new queue.)

- Forcibly remove one or more queues from BATDEF by the following method:

  1. Create an empty file with the command "BATGEN new-pathname".

  2. Add queues identical to those you wish to retain in the BATDEF file. (Names and all parameters must be identical.) Also add any new queues you wish to create.

  3. File your new BATDEF file with the BATGEN command:

     FILE BATCHQ>BATDEF

     The existing BATDEF file will then be replaced by the new one, and the new configuration will take effect immediately.

> Caution
>
> This method will abort all jobs running in the removed queues and delete all job information on those jobs and on any waiting or held jobs. It is not recommended as standard practice.

12 Working with PRIMENET

INTRODUCTION

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

- Using the ADDISK and SHUTDN commands
- Monitoring the File Transfer Service
- Monitoring half-duplex lines
- Communicating with operators on other systems
- Monitoring the Network Event Log

At some installations, system operators also perform network-related tasks that are normally described as System Administrators' duties (for example, configuring the network and maintaining 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.

USING THE ADDISK AND SHUTDN COMMANDS

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 commmand

ADDISK and SHUTDN are used for starting up and shutting down both local and remote disks. These two commands are described in general in Chapters 4 and 15. 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.
- Either you must start the disk on your system (using ADDISK), or your FAM must start it. (With FAM II, remote disks can be started even though the remote system is down or the disk on the remote system has not been started.)

A disk that has been shut down on its own system cannot be accessed by any other network node.

Adding Remote Disks (ADDISK)

When the ADDISK command is used to start up a remote disk, the command's format and action depend upon whether your system uses FAM I or FAM II to communicate with the remote disk's system. If your system uses FAM I, refer to Appendix L.

To find out which FAM is being used between your system and another, attach to the UFD CMDNC0 and run the NETCFG utility. Answer "YES" to the first prompt ("Review old network configuration?"). The resulting display includes FAM information. For example:

```
OK, A CMDNC0
OK, NETCFG
Review old network configuration? Y

Rev    19.0 network configuration file

Ring Net
        Name         Addr           Ring ID   FAM INFO   RLOG
        -------      ------------   -------   --------   ----
*ME*    SYB          123456789012         2
        SYA                               1   II/NO-VALID Yes
                         Node-Node password: ATTERCOP
        SYC                               3   II/NO-VALID Yes
                         Node-Node password: ATTERCOP
```

In this example, systems SYA and SYC are both using FAM II. (For information on the FAMs and on NETCFG, refer to the System Administrator's Guide.)

When FAM II is used between two systems, the format of the ADDISK command is:

ADDISK diskname-1 [diskname-2...diskname-9] -ON nodename

where nodename is the name of the remote system and diskname-n indicates the name of a remote disk to be added. Because ADDISK does not use physical device numbers under FAM II, an operator on any system may move disk partitions without having to inform the network of the new physical device numbers.

ADDISK simply adds the specified diskname to the local logical device list. There is no check on the up/down status of the remote system or on the existence of the disk. Thus, you need not wait until a remote system comes up in order to add one of that system's disks.

When you add a disk with ADDISK, the diskname appears in the STATUS DISKS list on your system. As a consequence of the new ADDISK functionality under FAM II, 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 disk is added under FAM II, the physical device number does not appear on the STATUS DISKS list.

In all cases, remotely added devices acquire the write-protection status assigned them on their local systems.

Shutting Down Remote Disks (SHUTDN)

Like ADDISK, the SHUTDN command differs according to whether the remote disk being shut down is accessed with FAM I or FAM II. If your system uses FAM I, refer to Appendix L.

To shut down remote disks that your system accesses via FAM II, use the command format:

SHUTDN diskname-1 [diskname-2...diskname-9] -ON nodename

where diskname-n is the diskname of a remote device to be disconnected. The command marks all file units as being on the partition that has been shut down. Under FAM II, you may shut down a disk from a remote node while that node (or the line to it) is down.

Under FAM II, a remote failure does not remove the disk from the local logical device list.

MONITORING THE FILE TRANSFER SERVICE (FTS)

This section 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.)

The Operator's Tasks

As an operator, your responsibilities with regard to FTS are:

- Monitoring user requests
- Initiating and monitoring the file transfer and file manager servers
- Monitoring and archiving FTS system log files
- Monitoring the UFD FTSQ*

Monitoring User requests: While it is the owner's responsibility 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 1.0]
82-06-03.16:49:41 SYSTEM T1$IO.FTN (100935308) Status - put on hold
 by user
82-06-08.10:58:52 SHELOB  HO.82-06-08  (65851122)  Status - put  on
 hold by user

OK, FTR -DISPLAY
[FTR rev 1.0]
Request            - T1$IO.FTN (100935308)
User               - SYSTEM
Queue              - fts$1
Queued             - 82-06-03.16:49:41  Status - put on hold by user
Last attempt       - 82-06-09.14:31:00  Attempts -      2
Current time       - 82-06-11.19:16:21
Source file        - <M1>FORMSSRC.19.1>RUN_TIME>IOS_SOURCE>T1$IO.FTN
Source file size   -       32328 bytes.
Destination file   - ABCD>FED>T1$IO.FTN
Source site        - CHAIR1
Destination site   - SYS.B
Source user        - SYSTEM
Options :-
Binary, Copy, No Delete, Source notify, No Destination notify.
More ? Y
Request            - HO.82-06-08 (65851122)
User               - SHELOB
Queue              - fts$1
Queued             - 82-06-08.10:58:52  Status - put on hold by user
Last attempt       - 82-06-09.14:31:00  Attempts -      2
Current time       - 82-06-11.19:16:32
Source file        - <COMND1>INFO>HO.82-06-08
Source file size   -         3524 bytes.
Destination file   - ECCE>HO.82-06-08
Source site        - CHAIR1
Destination site   - sys.b
Source user        - SHELOB
Destination user   - ecotop
Options :-
Binary, Copy, No Delete, Source notify, Destination notify.
```

The FTR command is described in detail below.

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 5) to make sure they are in operation.

Both the FTS server phantoms and the FTS manager phantom maintain COMO files while they are running. Server files have the pathname FTSQ*>COMO.FTS>servername. The manager COMO file 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*. 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. For example:

```
OK, ED 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
14.33.05: [3.1] RESULT: Transfer Aborted: Transfer not completed.
X
14.33.31: [2.1] RESULT: Transfer Rejected: Problem with remote
file.
X
14.38.08: [4.1] RESULT: Transfer Terminated: Satisfactory and
Complete.
```

In the above example, the operator has used EDITOR's X command to repeat the LOCATE RESULT command line.

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*

Among other 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.

The FTR command line format 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 on 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, or by the operator in distinguishing between two requests of the same name. You can find out the number of a request by means of the -STATUS option of the FTR command (see below).

The operator may use any of the following FTR Request Management options on any submitted file transfer request:

Option	Function
-ABORT	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. If the request is already aborting, an error message results. An operator may abort any request.

-CANCEL — Cancels a request from the request queue. Deletes a request from a file transfer request queue. If the transfer is currently in progress, the request will not be cancelled.

An operator may cancel any request.

-DISPLAY — Obtains and displays detailed information about a request. The display includes all the information given by the -STATUS option (see 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 an name, all requests belonging to the user with that name 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 on the queue of requests waiting to be initiated. The request is not initiated until it is released (see 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 canceled 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 Request Management option or the -HOLD Request Submittal 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 invoke the FTR command for any request. If the operator does not specify request-name, FTR displays all requests of all users.

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 servers, 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 (see the -START_MNGR option, below).

Note

When the local FTS server and YTSMAN are not running, local users may still queue requests using FTR.

The general format of the FTOP command is:

FTOP[option]

The option -START_MNGR applies to the FTS manager process YTSMAN. All the other options apply to ordinary server processes.

The FTOP options are:

Option	Function
-ABND_SRVR	Abandons an FTS server process. Causes the file transfer server server_name immediately to place all current file transfers on hold in the queue, and to log out. The format is: FTOP -ABND_SRVR server_name

If server_name is not running, an error message results. (Abbreviation: -ASV).

Note

Forced logout of a server is not recommended. The recommended way to stop a server is to use the -STOP_SRVR option (see below).

-ABRT_SRVR_LINK

Aborts an FTS server link. Causes the file transfer server server_name to place the current file transfer on link link_number (range 1 to 8) on hold in the queue. The server continues running; it does not log out. The format is:

```
FTOP -ABRT_SRVR_LINK server_name link_number
```

To find the link number of an active transfer, use the command option -LIST_SRVR_STS server_name (described below).

If server_name is not running or link_number is not active, an error message results. (Abbreviation: -ASVL).

-LIST_SRVR_STS

Lists server status. Lists the status of the server server_name. 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:

```
FTOP -LIST_SRVR_STS [server_name]
```

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 phantomed 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 Chapter 4). (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 phantomed 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 Chapter 4). (Abbreviation: -STRSV).

-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).

MONITORING HALF-DUPLEX (HDX) LINES

PRIMENET supports half-duplex (HDX) dialup connections over configured synchronous lines. At the application level, half-duplex and full-duplex lines function in the same manner. However, operator intervention is needed to assign and unassign HDX lines, and to start and stop HDX operations on a given line.

Like full-duplex synchronous lines, half-duplex lines are configured by the System Administrator through the NETCFG utility. To find out the line numbers and nodes that have been configured for HDX on your system, run NETCFG and answer "YES" to the first prompt ("Review old network configuration?").

Unlike full-duplex lines, HDX lines are initialized by the operator rather than at cold start. (Refer to the System Administrator's Guide for information on NETCFG.)

The three commands that help you control HDX lines are NET, HDXSTAT, and STATUS NET.

The NET Command

The NET command allows you to activate and deactivate a half-duplex (HDX) network connection whose lines and nodes have already been configured through NETCFG. The command may be issued only by an operator.

The command line format for NET is as follows:

NET {option argument}

Four options allow you to control the state of the HDX connection. These options are summarized below:

Option	Function
-ASSIGN line	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. line is the number of the HDX line, as configured in NETCFG. 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.
-START line [-SITE nodename]	Activates an HDX line. The line must be assigned by the -ASSIGN option before the -START option can be used. line is the number of the assigned HDX line to be activated. nodename is an HDX node (as configured in NETCFG). 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 nodename. If -SITE is omitted, the line will be used to receive calls from any HDX nodes that have been configured. 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 {line | 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. line 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 line	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.

Broken Connections: An HDX connection can be broken by other means than the NET -STOP command. The following are examples of conditions that break a connection:

- The name of the calling node is invalid or the password is unexpected.
- One of the two nodes is warm started or cleared for any reason.
- Miscellaneous line or modem problems occur.

These conditions have the same effect on the HDX line as does the NET -STOP command.

The HDXSTAT Command

The HDXSTAT command displays the current status of all HDX lines and nodes configured from your system. The display shows the condition of the phone connection for each line and node, as well as the link state. The link state is one of the following:

- Not assigned
- Assigned (assigned, but not started)
- Awaiting call (started without -SITE option; remote node has not called in)
- Trying to establish (started with -SITE option; no contact yet with remote node)
- Running
- Disconnecting (in the process of closing down)
- Offline (no connection currently exists to this node)

The following is an example of HDXSTAT's output:

```
OK, HDXSTAT
hdxstat, version 2.0
Status of HDX network on MON, JUN 7 1980 at 11:20:15

node     line     phone     state

SYS.A    SMLC01   up        running
SYS.B    —        —         offline
SYS.C    —        —         offline
SYS.D    SMLC03   up        running
```

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 5 of this book, the Prime User's Guide, and the PRIMOS Commands Reference Guide.

COMMUNICATING WITH OPERATORS ON OTHER SYSTEMS

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. The format of MESSAGE for operator-to-operator messages is:

```
MESSAGE -1 -ON nodename [-NOW] [-FORCE]
text-of-message
```

For an explanation of the MESSAGE command, refer to Chapter 15.

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 LOGPRT command with the -NET option. For example:

```
OK, LOGPRT -NET
```

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

LOGPRT is fully described in Chapter 5.

13 System Halts

INTRODUCTION

This chapter defines halt handling procedures for Prime's 50 Series machines with a Virtual Control Panel (VCP). If your equipment does not have a Virtual Control Panel, use the procedures described in Appendix E.

RECOGNIZING HALTS

Under certain unusual circumstances (hardware or software malfunctions), PRIMOS will execute the HLT instruction and halt. The three indications that a system halt has occurred are:

1. The control panel's red STOP light (the second red light on the left) turns on.

2. A halt message is printed on the supervisor terminal.

3. The VCP enters control panel mode, printing a CP> prompt.

It is also possible for the machine to "hang." In this case, the machine appears to have halted, but the red light is not on. Hung machines may be identified by the fact that no input or output is taking place at any peripheral equipment (including the supervisor terminal). Hung machines should be stopped, then treated like halted machines.

To stop a hung machine, do the following at the supervisor terminal:

1. Hit the ESCAPE key twice.

2. Type STOP.

If this procedure does not work, a MASTER CLEAR should be done.

> **Caution**
>
> Do not perform a MASTER CLEAR if a tape dump is desired.

GENERAL GUIDELINES FOR HANDLING HALTS

When a halt occurs, the operator has two duties:

- Get the system running again as quickly as possible.

- Record enough information so that an analyst can determine why the halt occurred. This will help prevent future halts.

In order to provide the necessary information for the analyst, the operator must:

- Make sure that a current set of load maps is stored in a safe place (preferably with the system logbook).

- Spool out LOGREC at regular intervals, and keep the spooled copies on file.

- Take tape dumps when halts occur. (Table 13-1 identifies those halts for which tape dumps are most likely to be useful.)

Load maps are named RING0.MAP and RING3.MAP. (Earlier systems have one map, named M_PRM0.) Load maps are stored in UFD PRIRUN. Since the load maps change whenever additions or updates of software are made to PRIMOS, the operator should be sure to spool new copies of the maps whenever the system is updated.

> **Caution**
>
> Never do a SYSCLR, VIRY, or MASTER CLEAR until you have determined the halt location. Once one of these instructions has been given, the halt location (and hence the cause of the halt) is unobtainable.

HANDLING HALTS UNDER PRIMOS

Halts that occur when the system is running under PRIMOS are handled in four steps, as follows:

1. Determine the segment number and address at which the halt occurred.
2. Look up the address on your spooled copy of the PRIMOS load map to determine the halt location.
3. Check Table 13-1 to find out which recovery procedure is appropriate for halts occurring at this location.
4. Take the appropriate steps for recovery, as detailed in Table 13-2.

These steps are described in more detail below.

Note

The P850 has a special halt handling procedure, which is described later in this chapter.

Step 1: Determine the Segment Number and Address

When a machine halts, it prints the following message at the supervisor terminal:

    Halted at xxxxxx/yyyyyy

xxxxxx is the segment number of the halt. yyyyyy is the address of the halt. Record the segment number and address in your system logbook.

Step 2: Determine the Location

Look up the address of the halt in a spooled copy of your current PRIMOS load map. Either the address you have or the address immediately preceding it in the listing should be a recognized halt location. (The names of recognized halt locations end in underscores: for example, PWRFL_.) Write the name of the halt location in the system logbook, next to the halt address.

An example of such a procedure is:

1. Halt message appears at terminal.

   HALTED AT 000004/000306: 003776

   CP>

2. Find the segment number (0004) on the load map. Your load map will be similar to this:

WARMCN	0004	000260	OTHER	
PFFLAG	0004	000262	OTHER	
PFAILS	0004	000263	OTHER	
MEMPA_	0004	000276	OTHER	
MCHK_	0004	000305	OTHER	
MMOD_	0004	000315	OTHER	
MEMH2_	0004	000317	OTHER	
CHKEND	0004	000321	OTHER	
MT2INT	0004	000321	OTHER	
MTINT	0004	000323	OTHER	
MGINT	0004	000325	OTHER	
MG2INT	0004	000327	OTHER	
INTEN2	0004	000476	OTHER	
CLKSEM	0004	000500	OTHER	
SEMCOM	0004	000500		COMMON
SLCSEM	0004	000502	OTHER	
AMLSEM	0004	000504	OTHER	
MPCSEM	0004	000506	OTHER	
MP2SEM	0004	000510	OTHER	

3. Find the word number on the map. Since 00305 is the largest number <= 306, it is the correct number for this halt.

4. The name of the halt location is MCHK_, a machine check.

How to Proceed Without a Map: If you cannot find a load map, and hence cannot look up the halt location, do a tape dump (action code D), and then try a warm start (action code W). If the warm start fails, try a cold start (action code C). As soon as the system is started, look for RINGO.MAP in the UFD PRIRUN. Look up the address of the halt (or the address preceding it) on this map and record it in the logbook. Then spool a copy of the map immediately and put it in the front of the logbook (or some other safe place) for the next time.

Step 3: Find The Appropriate Recovery Procedure

Refer to Table 13-1 for the action code that matches each recognized halt location.

If your halt occurred at an address that is not a recognized halt location, use the action codes for "all others," shown at the bottom of Table 13-1.

Table 13-1
Action Codes for Halts

Halt Location	Reason for Halt	Action Code(s)
AMLC1_	Bad AMLC interrupt	D, W
BDMEM_	Bad memory during cold start	M
BOOT0_	Halt after SHUTDN ALL or other PRIMOS error	D, C
DUMPF_	Tape error	D, C
MCHK_	Machine check	R, D, W
MEMH2_	Halt after automatic mapout of bad page	R,W
MEMPA_	Uncorrected memory parity error	X, M
MMOD_	Missing memory module	D, C
All others		D, C

Step 4: Take the Appropriate Steps for Recovery

Table 13-2 details the procedures to follow for each action code. Follow the instructions given in this table, and record the results in your system logbook.

Handling Halts with the Prime 850

Due to the fact that it has two instruction stream units (ISUs), the P850 requires a somewhat different halt handling procedure. The steps in this procedure are:

1. Determine which instruction stream has halted. This is done by typing:

   A 4/176106 (CR)

   If the number displayed is 000004/176106: 041004, the halt has occurred in stream 1.

   If the number displayed is 000004/176106: 102010, the halt has occurred in stream 2.

2. Determine the location of the halt.

   If the halt has occurred in stream 1, the halt location will be displayed in the message:

   HALTED AT xxxxxx/yyyyyy

   You may now follow the standard halt procedures outlined earlier in this section.

   If the halt has occurred in stream 2, proceed as described in the following paragraph.

Stream 2 Halts: To handle a halt in stream 2 of the P850, follow the steps described below.

1. Reset the VCP to accept the next command by typing a slash:

   /

2. Do not do a SYSCLR prior to executing the following procedure. Determine the halt address by typing:

   A 14/2516 (CR)
   (CR)

   The two numbers displayed will give the stream 2 halt address.

3. Reset the VCP to accept the next command by typing a slash:

   /

4. Determine the DSWPARITY (Diagnostic Status Word PARITY) error information by typing:

   A 14/2556 (CR)
   (CR)

   The two numbers displayed will give the stream 2 DSWPARITY error information.

5. Reset the VCP to accept the next command by typing a slash:

   /

The following example shows the correct method for handling a stream 2 halt:

```
CP> A 4/176106                  CR
 000004/176106:     102010      /
CP> A 14/2516                   CR
 000014/002516:     xxxxxx      CR        ISU-2 halt
 000014/002517:     yyyyyy      /         address
CP> A 14/2556                   CR
 000014/002556:     xxxxxx      CR        ISU-2 DSWPARITY
 000014/002557:     yyyyyy      /         address
CP>
```

HANDLING HALTS UNDER PRIMOS II

PRIMOS II may be halted by the operator to abort a long listing or to recover from a bad startup. A halt under PRIMOS II could also be caused by a hardware condition or equipment failure. If the fault is in the hardware, the operator should not attempt to restart the system until the hardware problem has been corrected.

To halt the machine:

1. Hit the ESCAPE key twice.

2. Type STOP.

To restart the machine:

3. Type RUN 170000. This restarts PRIMOS II at command level.

Table 13-2
Recovery Procedures

Code	Action
C	Cold Start Bring the system up as if it had been shut down normally as described in Chapter 4.
D	Crash Magnetic Tape Dump 1. Mount a scratch tape on drive unit 0 or 1. 2. Type: CP> SYSCLR CP> RUN 775 if tape is mounted on unit 0, or CP> SYSCLR CP> RUN 776 if tape is mounted on unit 1. 3. The tape will write and then rewind. Label the tape with the date and time of the halt (and machine identification, if needed). Record the dump in the logbook, and give the tape to the person responsible for examining unusual halts. If the tape does not rewind, repeat the procedure.
M	Map Out Bad Page Type: CP> RUN When this action is taken following a memory parity error, PRIMOS maps out the bad page; that is, it records the location as being bad and doesn't use that page of memory in the future. PRIMOS then continues the cold start.

Table 13-2 (continued)
Recovery Procedures

Code	Action
R	Record Register Settings Determine and log any hardware problems by dumping the machine's three status words (four words for the Prime 750 and 850): CP> D DSWSTAT (value) CP> D DSWRMA (value) CP> D DSWPB (value) and for the P750 and P850: CP> D DSWPARITY (value) Record the values in the logbook.
W	Warm Start Restart the machine using this procedure: CP> SYSCLR CP> RUN HALTED AT: 1001: 000010 CP> RUN *** WARM START *** Note Warm start restarts all controllers before restarting PRIMOS. Thus, the warm start may run for twenty seconds or more before the WARM START message appears at the user terminals. It will be slightly longer before a message appears at the supervisor terminal. Do not assume a warm start has failed without waiting twenty seconds and checking the user terminals for the WARM START message.

Table 13-2 (continued)
Recovery Procedures

Code	Action
X	Uncorrected Memory Parity Error 1. Determine where error occurred and record results in logbook. Type: CP> D X (Indicates what user got error) CP> D A (Indicates page number of error) CP> D B (Indicates offset without page) 2. If user number (from register X) is 1 (supervisor), then a cold start must be done. 3. If user number is not 1, have PRIMOS map out the bad page (as shown under action code M). This prevents that page of memory from being used again. Type: CP> RUN 4. When it has mapped out the bad page, the system will halt at MEMH2_. It must then be warm started (action code W).

14
Working with PRIMOS II

INTRODUCTION

PRIMOS II is Prime's single-user operating system. Its primary uses are in the bootstrapping of PRIMOS and in the running of diagnostics and maintenance utilities. At other times, in special situations, it may also be desirable to run the computer as a single-user system under PRIMOS II. Such situations, however, are extremely rare. As a general rule all operations are conducted under PRIMOS.

Operation under PRIMOS II should be avoided whenever possible.

Using PRIMOS II

PRIMOS II and its supporting software allow the user to:

- 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 disk pack maintenance with FIXRAT

- Perform operations with maximum security because of the inherent single-user configuration

Several operational aspects of PRIMOS II differ from PRIMOS. For example:

- Only certain commands may be used. The commands that run under PRIMOS II are:

ATTACH	INPUT	PHYRST
BINARY	LOGPRT	PM
CLOSE	LISTF	PRERR
CNAME	LISTING	PRIMOS
COMINPUT	MAGRST	RESTORE
COPY_DISK	MAGSAV	RESUME
CREATE	MAKE	SAVE
DELETE	MDL	SHUTDN
FIXRAT	NSED	STARTD
FUTIL	OPEN	STATUS
	PASSWD	SVCSW

  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 Appendix C and Appendix B.)

- PRIMOS II occupies the user's address space, that is, the same physical memory as user programs. This places some restrictions on program loading to ensure that PRIMOS II and the loader are 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.

- The LOGPRT command may not use the -SPOOL option. Additionally, CMDNC0 may not have a password if LOGPRT is to work under PRIMOS II.

> **Caution**
>
> If files are added, extended or deleted under PRIMOS II, quota information is rendered invalid.
>
> **Caution**
>
> ACLs are not recognized under PRIMOS II, and quotas are not enforced.

PRIMOS II Commands

This chapter documents special commands that are used when working with PRIMOS II. Included are discussions of the following:

- Disk pack maintenance (the FIXRAT command)
- File manipulation (the FUTIL command)

DISK PACK MAINTENANCE IN PRIMOS II

The external command FIXRAT resumes 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 section 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 contains 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 when the disk is formatted by MAKE, 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 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 is the root or trunk of the tree, the links between directories and files or subdirectories are branches, and the directories and files are nodes.

A directory tree consists of all files and subdirectories that have their root 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 down 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; all answers are 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 files names to be typed? (Default is NO.)

3. 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.

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 that is not backedup on the disk, 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 have not given an incorrect 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 section below). 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:

Level	Output
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 (level 3 directories)
.	.
.	.
.	.

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 the TYPE DIRECTORIES TO LEVEL = question with 1, directories are 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 later in this chapter in the section on FUTIL subcommands.)

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 CMDNC0
 END   CMDNC0       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. This example is 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 that are files 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
```

and exits to PRIMOS.

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, prints the message:

```
DSKRAT FIXED
```

and 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:

<u>filename</u> The name given to the file.

<u>filetype</u> The type of file: SAMFIL, DAMFIL, SAMSEG, DAMSEG, SAMUFD, or ILLEGAL.

<u>bra</u> The beginning record address of the file.

<u>fra</u> The record address of the father-directory.

<u>index-level</u> The current index level (0 except for DAM files).

<u>cra</u> The current record address.

<u>track</u> The cylinder number at which the error occurred. The outside rim of the disk is track 0. Track numbers increase inwards up to 822.

<u>head</u> The surface of the disk. The bottom of the disk pack is head 0. (Storage modules have up to 19 heads.)

<u>directory-path</u> 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 directory into which the file is entered. If the directory is a UFD, FIXRAT removes the entry from the directory in an action similar to that of the DELETE command. If the directory is a segment directory, FIXRAT identifies the entry as a null entry.

<u>Segment Directories</u>

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 deep). When this condition is detected, FIXRAT prints the message:

    DIRECTORY ufdname NESTED TOO DEEPLY

and then asks:

    TRUNCATE DIRECTORY?

If the answer is NO, FIXRAT aborts. Otherwise, FIXRAT truncates the UFD named ufdname, by making it an empty UFD (i.e., its entry as a UFD in the parent directory is preserved, but ufdname will have no files or directories subordinate to itself). After truncation of ufdname, 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)

Description of Bad Structure Messages

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 "daily user 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 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 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 and 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. The user 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 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 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 is too long to represent the file. FIXRAT truncates the index.

- DAM INDEX TOO SHORT

The index 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 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 there is probably not much an operator can do, other than 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 D 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. If desired, this badspot record location may be manually added to the BADSPT file using SAVE and RESTOR (see Chapter 6). 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, FIXRAT must be rerun after the BADSPT file is modified.

- 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).

FILE MANIPULATION UNDER PRIMOS II

In PRIMOS, files are moved with the COPY command. In PRIMOS II, the FUTIL utility is used.

Invoking FUTIL

FUTIL may be invoked from either PRIMOS or PRIMOS II. For operations under PRIMOS, it is recommended that the COPY command be used rather than FUTIL.

To invoke FUTIL, input the command name FUTIL. When loaded, FUTIL prints the prompt character, >, and awaits a subcommand string.

```
OK: FUTIL
[FUTIL rev 19.0]
>
```

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 a copy or delete operation may cause a pointer mismatch or bad file structure, or a directory with a partial entry. PRIMOS 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.

Subcommand	Function
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 whose filename begins with the characters specified as prefix. If level is specified greater than 1, that many levels of sub-UFDs (including the current 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 to fileb in the TO directory and, optionally, 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 the same 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 the same 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. 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.

LISTF [level] [FIRST] [LSTFIL] [PROTEC] [SIZE] [RWLOCK] [TYPE] [DATE] [PASSWDS]	Lists the FROM directory, the TO directory treename, 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, data/time modified, and (on directories) owner and nonowner passwords.
LISTSAVE filename [level] [PROTEC] [SIZE] [TYPE] [DATE] [RWLOCK] [PASSWDS] [FIRST]	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] [PASSWDS] [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.
SRWLOC filename number	Sets the per-file read/write lock for the file specified by filename. The parameter number is an octal number that is the read/write lock setting.

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 delete 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.

Restrictions

Use Under PRIMOS: When using FUTIL under PRIMOS, certain operations may interfere with the work of other users. For example, a UFDCPY command to copy all files from a UFD currently used by another logged-in user may fail. 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 it will abort. 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, i.e., 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.

15 Operator Commands

INTRODUCTION

This section describes commands that are normally used by the operator at the supervisor terminal. Some of these commands may 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, and 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 devices for an available table entry location; if one is found, ADDISK starts up each specified device. If a specified device is already started, no action is taken unless write protection is changed or -RENAME is given.

The command format is:

```
ADDISK [PROTECT] {pdev-1 [pdev-2] ... [pdev-n]}
                 {pdev -RENAME packname       }
```

pdev is the physical device number of the partition being added.

ADDISK takes the PROTECT argument and the -RENAME option:

PROTECT	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 given pdev by shutting down the disk and then respecifying the ADDISK command with or without the PROTECT option. Respecifying the ADDISK command for a running disk does not change its protection. (If an ADDISK PROTECT command is issued for a disk that does not have protection enabled, the disk should first be shut down and the 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 " " of disk pdev not unique (conflicts with LDEV nn). 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.

Networks

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 FAM II system. The node must be FAM II enabled (see NETCFG). 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.

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.

If your system uses FAM I rather than FAM II, a different format for ADDISK must be used. See Appendix L.

ADDISK Messages

ADDISK prints the messages listed below. When a warning message is 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.

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 62 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.

- For FAM I links usage is:
  "ADDISK system pdev1 [pdev2...]" (ADDISK)

The format of a (presumed) FAM I ADDISK command was incorrect. Normally this indicates use of the -ON option specifying a node connected to your system via FAM I.

- For FAM II links usage is:
  "ADDISK packname1 [packname2...] -ON system" (ADDISK)

The format of a (presumed) FAM II ADDISK command was incorrect. Usually this indicates omission of the -ON option when adding disks on a node connected to your system via FAM II.

- *** 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 " " of disk pdev not unique (conflicts with LDEV nn).

(Warning) Partition packname already exists on this system's disk list. Given when "-ON nodename" is not used.

- Node "nodename" not configured in network. (ADDISK)

The node specified as nodename is not currently FAM II enabled.

- Only one disk may be RENAMEd at a time. (ADDISK)

Only one PDEV may be specified at a time 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 overlaps with, or is assigned to, the paging partition.

- 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 used on a disk added with the PROTECT option.

▶ AMLC

Starts up an AMLC line. The command format is:

AMLC [protocol] line [configuration] [lword]

protocol is one of the following:

TTY	Default terminal protocol
TTYHS	Terminal with per-character interrupt
TRAN	Transparent (no character conversion)
TRANHS	TRAN with per-character interrupt
TTYNOP	Ignores all traffic
TTYUPC	Translates lowercase alphabetic characters to uppercase for output; uses normal terminal protocol for input
TTYHUP	Translates lowercase alphabetic characters to uppercase for high-speed output; uses normal terminal protocol for input

line is the AMLC line number, in octal. The maximum value equals the number of terminal users (specified by the configuration directive NTUSR) plus the number of remote users (specified by the configuration directive NRUSR) configuration is an octal number used to set the line configuration. Three common configuration values, and the baud rates they represent, are:

configuration	baud rate
2213	300
2313	1200 (default)
2413	9600

The AMLC command is described in detail in the System Administrator's Guide.

▶ ASSIGN

The operator may use the ASSIGN command to assign magnetic tape drives or other peripheral devices to users. ASSIGN is discussed in Chapter 10.

▶ 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.

▶ 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 rights to the BATDEF file, users can invoke the BATGEN command to monitor queue availability and parameters. BATGEN is described in Chapter 11.

▶ 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:

$$\underline{\text{CHAP}} \begin{Bmatrix} \text{-userno} \\ \text{ALL} \end{Bmatrix} \text{[priority [timeslice]]}$$

-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.

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 without waiting for a scheduler semaphore (HIPRIQ, ELIGQ, LOPRIQ). The process remains on the ready list until it waits for some other semaphore, such as BUFSEM. The process can only be interrupted by a higher priority process that is not waiting for one of the scheduler semaphores.

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 as much time as it wants, whenever it wants it.

▶ 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.

▶ 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.

▶ 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 020250 from the Assignable Disks Table and unassign the partition, type:

OK, DISKS NOT 20260
OK, UNASSIGN DISK 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 of 3271/3277 terminals or OWL 1200 terminals attached to Prime as IBM host terminals. For details, see the Distributed Processing Terminal Executive Guide.

DROPDTR

This command applies to the DTR (Data Terminal Ready) signal associated with an AMLC line. DROPDTR is useful only with a user line connected into a "port selector" or dialup modem. Issuing DROPDTR, which is allowed only while logged out, 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.

▶ FIXBAT

FIXBAT is a utility for checking the Batch queue database integrity. It is normally supplied in UFD BATCHQ as *FIXBAT. See Chapter 11 for a complete description of FIXBAT.

▶ FIX_DISK

FIX_DISK is an external command that checks the PRIMOS file integrity on any physical device or partition. FIX_DISK is described fully in Chapter 8.

▶ 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. FTR is described in Chapter 12.

▶ FTOP

FTOP is the File Transfer Service (FTS) operator command. It provides special FTS operational privileges to the operator logged in as SYSTEM. The operator can use the FTR command under FTS to monitor and control user file transfer requests. With FTOP the operator can initiate, monitor, and control the FTS servers. FTOP is described in Chapter 12.

▶ 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 decsribed in the System Administrator's Guide.

▶ 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.

▶ LIST_ACCESS

Lists the access rights for any object. The command format is:

    LIST_ACCESS [objectname]

LIST_ACCESS is discussed in Chapter 5.

▶ LIST_GROUP

Lists the ACL groups. LIST_GROUP is discussed in Chapter 5.

▶ LIST_PRIORITY_ACCESS

Reads the contents of a priority ACL on a disk partition. LIST_PRIORITY_ACCESS is discussed in Chapter 5.

▶ LOGOUT

This command logs out the specified process connected to the system. Files are closed, devices are unassigned, the UFD is detached, 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 running devices—remote, local, and phantom—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 SHUTDN 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.)

▶ LOGPRT

LOGPRT is an external command that prints the contents of the event logging file. See Chapter 5 for a detailed description of LOGPRT.

▶ LOOK

LOOK is an internal operator command that provides access to any user segment in the system. 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.

This command is intended mainly for the use of systems engineers and field analysts as a debugging tool. 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, let segno equal zero.

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 7, and in the Magnetic Tape User's Guide.

▶ 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.

▶ MAKE

MAKE creates a structure for any PRIMOS-supported disk pack or partition. MAKE is described fully in Chapter 6.

▶ MAXSCH

MAXSCH is an internal command controlling the amount of overlapped processing performed by the system. It controls 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 is 3.

▶ MAXUSR

MAXUSR is an internal command that controls the number of users that are allowed to be logged into PRIMOS at any time. Normally, this variable is set to the number of configured users. The command format is:

    MAXUSR n

n is the maximum number of allowable users.

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."

There are two situations in which this facility will be useful:

- At cold start, if there are modifications to be made to PRIMOS, setting MAXUSR to 1 or 2 will prevent users from logging in until the operator decides it is appropriate. (The MAXUSR command should be entered before setting the date and time.)
- 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 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.
- From any user terminal to the supervisor terminal.
- From the supervisor terminal to all users.
- From the supervisor terminal to a specified user.
- From the supervisor terminal to another supervisor terminal on a different node on the network.

The format of the MESSAGE command is:

```
            ( username    )
MESSAGE     { -usernumber }        [option]
            ( ALL         )
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 usernumbers 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.

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 the receive state has been set to (see the section on receive states below).

If -FORCE is not used, the operator is notified if reception of the message is blocked and asked whether to send or to abort the message.

-ON nodename — Allows the sender to communicate with other nodes.

-STATUS arg — Displays the message receiving state of system users. arg may be any of the following arguments, used to specify an individual user: username, usernumber, ME.

Operator-to-Operator Messages

The format of an operator-to-operator message is:

```
MESSAGE -1 -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. (SMSG$)" 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.

To cancel a stored broadcast message, a null line must be entered as the text of message.

Receive States

Users may set the receive state of their terminal with the ACCEPT, DEFER, and REJCT arguments of the MESSAGE command. The arguments control the flow of messages according to the following pattern:

ACCEPT — Enables reception of all messages

DEFER — Inhibits immediate messages; accepts messages sent without the -NOW option

REJECT — Inhibits all messages

Terminal receive states of all users may be determined with the -STATUS option of the MESSAGE command. For example:

```
OK, M -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
```

▶ NET

NET controls half-duplex (uni-directional) PRIMENET on previously configured SMLC/MDLC lines. The command format is:

```
    ┌ -ASSIGN lines                  ┐
NET │ -START line [-SITE sitename]   │
    │ -STOP line sitename            │
    └ -UNASSIGN line                 ┘
```

This command can be executed only from the supervisor terminal. NET is discussed in Chapter 12. For more details on NET see the PRIMENET Guide.

▶ NETCFG

NETCFG builds the network configuration file. NETCFG is described in detail in the PRIMENET Guide and in the System Administrator's Guide.

▶ 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 7, and in the Magnetic Tape User's Guide.

▶ 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.

▶ PRIMOS

PRIMOS is the command used by the operator to boot PRIMOS. See Chapter 4.

▶ PROP

PROP is an external command invoking the spool queue management utility for the system printer. PROP is described fully in Chapter 9.

▶ REPLY

The operator uses the REPLY command to reply to users' mag tape requests. See Chapter 10 for a discussion of REPLY.

▶ SETIME

The SETIME command sets date and time. The SETIME command must be given following a startup of PRIMOS before any users are allowed to log in. It may be reentered at any time during system operation. The command format is:

SETIME -mmddyy -hhmm

mmddyy are digits that represent the month, day, and year (last two digits).

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

▶ SETMOD

SETMOD sets the mode for magnetic tape assignments. It must be given by the operator from the supervisor terminal. The command format is:

$$\text{SETMOD} \begin{Bmatrix} \text{-OPERATOR} \\ \text{-USER} \\ \text{-NOASSIGN} \end{Bmatrix}$$

See Chapter 10 for a discussion of SETMOD.

▶ 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-'2277 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 '2277 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 which may contain system utilities are described in the System Administrator's Guide.

▶ SHUTDN

The SHUTDN command performs tasks necessary to shut down the PRIMOS system in an orderly manner. The command format is:

```
         ( [pdev-0 [pdev-1 ... pdev-n]] )
SHUTDN  {  ALL                           }
         ( pdev -RENAME packname         )
```

The arguments function in the following manner:

Argument	Function
ALL	Performs a complete PRIMOS system shutdown. All user files are closed, physical disks are disconnected from the system, and PRIMOS inhibits interrupts, exits page mode, stops the system clock, and halts. Under PRIMOS, the SHUTDN ALL command from the supervisor terminal causes the prompt: REALLY? The operator must answer "YES" for the shutdown to occur. **Note** If the shutdown is a normal one, it is advisable to use the LOGOUT ALL command before using SHUTDN ALL.
pdev-0 ... pdev-n	Shuts down the listed devices by closing all files opened in the listed devices and by detaching all users attached to the listed devices. The specified devices are no longer available for PRIMOS file I/O operations until they have been specified in a subsequent ADDISK command. STATUS can be used to list the devices currently started up. (The physical device numbers do not have to be given in logical drive order as they do for the ADDISK command.)

> **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.

pdev -RENAME packname Allows the operator to supply a new packname for a disk when shutting it down. Only one disk can be specified each time this option is used. For example:

OK, SH 461 -RENAME B3.BAK

Under PRIMOS II, SHUTDN with no arguments specified performs the same functions as SHUTDN ALL.

Example of Selective Shutdown

If the command at disk startup was:

OK, ADDISK 460 462 464 1260

to replace the pack on physical device 464 (logical device 2) the operator would give the command SHUTDN 464. The operator would then stop the drive, replace the pack, and restart the drive.

Networks

Remote devices connected to the system can be disconnected by the command:

SHUTDN packname-1 [packname-2...packname-9] -ON nodename

nodename	Network name of the system on which the devices are physically mounted.
packname-n	Remote devices, made available to the local system, which are to be disconnected.

Notes

If a device is shut down at its local system, it is no longer available for use on any other system in the network.

A SHUTDN ALL command disconnects all local disks and remotely available disks. It does not shut down the remote disks on the system on which they are physically mounted.

SHUTDN Messages

SHUTDN prints the messages listed below. When a warning message is printed, SHUTDN 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.

- For FAM I links usage is:
  "SHUTDN system pdev1 [pdev2...]" (SHUTDN)

The format of a (presumed) FAM I SHUTDN command was incorrect. Normally this indicates use of the -ON option specifying a node connected to your system via FAM I.

- For FAM II links usage is:
  "SHUTDN packname1 [packname2...] -ON system" (SHUTDN)

The format of a (presumed) FAM II SHUTDN command was incorrect. Usually this indicates omission of the -ON option when adding disks on a node connected to your system via FAM II.

- 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.

- 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 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.

SHUTDN is discussed in Chapter 4.

▶ 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, usernumber, line number, devices, and peripherals). See Chapter 5 for a complete description of the STATUS command.

▶ 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 10.

▶ 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, see Chapter 5.

▶ USRASR

The USRASR command allows the supervisor terminal to act as a user terminal by associating it with a different address space. After initiating USRASR, it is still possible to invoke some supervisor commands.

The command format is:

USRASR user-number

user-number is a usernumber. For example:

OK, USRASR 4

The USRASR command works correctly only if the associated communications line is not enabled on the AMLC. 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 AMLC line 2.

To return the terminal to operation as a normal supervisor terminal, type:

OK, USRASR 1

APPENDIXES

A
Bootstraps

INTRODUCTION

Chapter 4 contains instructions for performing a standard PRIMOS boot from disk. This appendix supplies additional information on booting the system. Read this appendix if you need to:

- Boot PRIMOS from magnetic tape (for example, if your system disk pack crashes or fails to boot)
- Boot PRIMOS from a nonstandard or obsolete medium (for example, paper tape, fixed-head disk, obsolete types of moving-head disks, or floppy disk)
- Boot an obsolete version of PRIMOS II
- Build, modify, or copy the BOOT program or the MTBOOT program
- Interpret an error code that appears in the control panel data lights or at the supervisor terminal during a boot

Note

If you need to perform a boot from magnetic tape, you may want to turn directly to the section on the standard magnetic tape boot. That section describes the most common procedures to follow if your master disk has crashed or will not boot. The "standard" setup is described in that section. If your situation is nonstandard in any way, you will probably need to read the other parts of this appendix.

Note

To boot a Rev. 19 system, you must have a Rev. 19 PRIMOS II (*DOS64) installed.

When you first turn on the power to a Prime CPU, the computer's semiconductor read/write memory is empty. To make the CPU useful, you must load a program (preferably an operating system) into memory from an external device such as a disk. This loading process (called "booting the system") involves two major steps:

1. By means of either the BOOT command (for a computer with a Virtual Control Panel) or the control panel rotary switch (for a computer without a VCP), you load a short program called a "bootstrap" from the computer's Programmable Read-Only Memory (PROM) into main memory. This short program in turn loads the program BOOT (or MTBOOT, for magnetic tapes) from an external device into main memory.

   Step 1 is sometimes called the "pre-boot."

2. BOOT (or MTBOOT) performs various checks of main memory, and then loads in the operating system (or another program) from an external device.

In most cases, once you initiate the pre-boot (Step 1), the rest of the boot proceeds automatically. An exception is the case of a boot from magnetic tape, which sometimes requires several additional steps.

The computer's PROM actually contains three different bootstrap programs. One is used for auto-start boots (see below), one for boots from paper tape, and one for mass storage boots (from disk, magnetic tape, or floppy diskette). When you initiate a pre-boot, you supply an octal number that determines which bootstrap is used. This octal value determines other factors as well. For example, if you are booting from disk, the value determines whether BOOT loads in PRIMOS automatically in Step 2, or whether it loads in PRIMOS II (in which case you must start up PRIMOS yourself, as described in Chapter 4).

The five remaining sections of this appendix discuss:

- How to initiate the pre-boot; the octal parameters and their meanings
- The BOOT program on disks
- The MTBOOT program on magnetic tapes
- How to perform a standard boot from magnetic tape
- The errors that can occur during a boot

Note

Certain of the devices described in this appendix are no longer supplied with Prime computers. The information is included here for reference and archival purposes.

THE PRE-BOOT

Initiating the Pre-Boot

On a computer with a VCP, after the power is turned on and the equipment is ready, the following message is printed at the supervisor terminal:

```
*** CPU VERIFIED ***
CP>
```

At this point, you initiate the pre-boot by typing the command BOOT followed by an appropriate octal value. For example, for the standard disk boot (from Controller 0) described in Chapter 4, you type:

CP> BOOT 14114

On a computer without a VCP, after the power and equipment are turned on, you initiate the pre-boot by:

1. Turning the front panel rotary selector switch to STOP/STEP
2. Pressing the MASTER CLEAR switch
3. Turning the ADDRESS/DATA switch to ADDRESS
4. Setting the control panel sense switches to the appropriate octal number
5. Turning the rotary switch to LOAD
6. Pressing START

The octal values supplied during a pre-boot have the same meanings on any Prime computer, whether they are entered through the keyboard (computers with VCPs) or through the control panel sense switches (computers without VCPs). The possible octal values and their meanings are discussed later in this section.

What Happens During the Pre-Boot

Your computer has either 256 or 512 16-bit words of PROM from which bootstrap programs can be loaded into main memory. The sequence of events just after you initiate the pre-boot is as follows:

1. PROM locations '0 to '50 are read into memory locations '6 to '56.

2. Execution begins (in 16S mode) at the address in location '7.

3. The A-register is saved in location '57.

4. Depending upon the octal value you have supplied, the appropriate bootstrap is loaded from PROM into memory.

The Pre-Boot Octal Values

If your computer has a VCP, the octal number that you type in after the BOOT command consists of up to 6 digits. (BOOT xx is interpreted as BOOT 0000xx.) If your computer does not have a VCP, you enter the octal number in bit format through the control panel sense switches. Each of the 16 sense switches represents one bit. A sense switch in the up position represents a bit that is set (1), and a sense switch in the down position represents a bit that is reset (0). To translate between the sense switches (bits) and octal digits, use the following correspondence:

Sense Switches (Bits)

$$\underbrace{1}_{1}\quad \underbrace{2\ \ 3\ \ 4}_{2}\quad \underbrace{5\ \ 6\ \ 7}_{3}\quad \underbrace{8\ \ 9\ \ 10}_{4}\quad \underbrace{11\ \ 12\ \ 13}_{5}\quad \underbrace{14\ \ 15\ \ 16}_{6}$$

Digits

The leftmost digit can be 0 or 1; the other digits can range from 0 through 7. For example, the octal number '014114 would be entered on a non-VCP machine by setting sense switches 4, 5, 10, 13, and 14.

The rightmost octal digit (sense switches 14, 15, and 16) indicates what type of physical device contains the BOOT (or MTBOOT) program. (This digit therefore also determines which bootstrap program is loaded from PROM into main memory.) The eight possible values have the following meanings:

Bit 14	Bit 15	Bit 16	Octal	Meaning
0	0	0	=0	Auto-start
0	0	1	=1	ASR paper tape (MDL format)
0	1	0	=2	High speed paper tape (MDL format)
0	1	1	=3	Fixed head disk
1	0	0	=4	Moving head disk
1	0	1	=5	Magnetic tape
1	1	0	=6	Floppy disk (diskette)
1	1	1	=7	Spare

A value of 0 reads in the auto-start bootstrap; a value of 1 or 2 reads in the paper tape bootstrap; and a value of 3 through 7 reads in the mass storage bootstrap. The eight possibilities are discussed below.

Note

The first five digits of the octal number (bits 1 through 13) have special meanings that depend on your choice of last digit. The list below tells what bits 1-13 mean in each of the eight cases. A bit without an assigned meaning should be reset (0).

Auto-Start (0): Reads in the auto-start bootstrap. This bootstrap differs from the others in that it simply starts execution of a program that is already loaded into memory. Execution begins (in 64R mode) at the location specified in bits 1 to 10 (digits 1 to 4, representing locations '1000 to '1777). If bits 1 to 10 are 0, a default of '1000 is used. For examples of auto-start boots, refer to the section on the standard magnetic tape boot, later in this appendix.

Paper Tape (1 and 2): Reads in the bootstrap for MDL-format paper tape, modifying it for either ASR paper tape (1) or high speed paper tape (2).

Mass Storage (3-7): Reads in the mass storage bootstrap, performing further selection for fixed-head disk (FHD), moving head disk (MHD), magnetic tape (MT), diskette, or spare.

Fixed Head Disk (3): Bits 12 and 13 are used to identify the disk drive from which the program BOOT is to be loaded. Bit 13 selects between controller 4001 (bit 13 = 0) and controller 4002 (bit 13 = 1). If bit 13 is 1, bit 12 selects between device addresses '21 (bit 12 = 0) and '23 (bit 12 = 1). The possible combinations appear below (x means that the bit is ignored).

Bit 12	Bit 13	
x	0	4001
0	1	4002, address '21
1	1	4002, address '23

The bootstrap reads record 0 of the disk (using the 448-word PRIMOS record format) into memory, starting at location '770. Execution begins at location '1000, via a JST '777 instruction. Record 0 is the first record of the BOOT program. (See the section on BOOT, later in this appendix.)

Bits 1 through 4 can be used to control the amount of memory that the program BOOT can use to read in PRIMOS II. (BOOT reads in PRIMOS II before starting up PRIMOS.) Before Rev. 16, these bits could be used to force BOOT to read in a smaller version of PRIMOS II than it normally would. At Rev. 16 and after, the only version of PRIMOS II that is supplied is *DOS64. However, the use of bits 1 through 4 is documented here for archival purposes.

The version of PRIMOS II that BOOT reads in (*DOS16, *DOS24, *DOS32, or *DOS64) is determined by available memory size. If bits 1 through 4 are all reset (0), the highest-memory PRIMOS II that will fit into available memory is read. If any of bits 1 through 4 are set (1), then bits 1-4 are treated as the most significant bits of the high address of memory +1. The versions of PRIMOS II corresponding to the addresses specified in bits 1 through 4 are as follows:

Address	PRIMOS II Version
0	Highest that will fit
'20000	Error
'30000	*DOS16 (obsolete)
'40000	*DOS16 (obsolete)
'50000	*DOS24 (obsolete)
'60000	*DOS24 (obsolete)
'70000	*DOS32 (obsolete)
'1x0000 (x=anything)	*DOS32 (obsolete)

*DOS64 cannot be forced by setting bits 1 through 4.

This bootstrap waits until the disk drive is ready, and retries if status errors occur.

Moving Head Disk (4): Moving head disks come in three varieties:

- Two-platter drives: 6M bytes (obsolete) or 12M bytes
- 20-surface drives (obsolete)
- Storage modules (40MB (obsolete), 80MB, or 300MB)

Bits 10, 11, 12, and 13 are used to indicate the type of drive from which the BOOT program is to be read, as follows (x means that the bit is ignored):

Bit 10	Bit 11	Bit 12	Bit 13	Type of Drive
x	0	x	0	Cartridge, upper surface, 4001 controller
x	1	x	0	Cartridge, lower surface, 4001 controller
0	0	0	1	Cartridge, upper surface, address '21 20-head disk pack, head 1, address '21
0	1	0	1	Cartridge, lower surface, address '21 20-head disk pack, head 2, address '21
0	0	1	1	Cartridge, upper surface, address '23 20-head disk pack, head 1, address '23
0	1	1	1	Cartridge, lower surface, address '23 20-head disk pack, head 2, address '23
1	x	0	1	Storage module, address '26
1	x	1	1	Storage module, address '27

In all cases except that of the storage module, record 0 of the selected surface of physical drive 0 is read into memory starting at location '770. (The 448-word PRIMOS record format is assumed.) Execution is begun at location '1000 via a JST '777 instruction. Record 0 is the first record of the BOOT program.

In the case of a storage module, record 0 is a 1040-word PRIMOS II record that contains the entire BOOT program. It is read into memory starting at location '760 (to allow for the 16-word header). Execution begins at '1000.

In the case of the storage module, bits 4 and 5 are defined. If bit 4 is set (1), the BOOT program reads in the operating system from the same device that the BOOT program itself came from. If bit 4 is reset (0), BOOT prompts you for a physical device from which to read in the operating system.

If bit 5 is set (1), BOOT starts up PRIMOS automatically. If bit 5 is reset (0), BOOT loads PRIMOS II, and you must start up PRIMOS yourself (as described in Chapter 4).

Thus, in a standard, automatic disk boot from a storage module on Controller 0, the octal value is '014114 (sense switches 4, 5, 10, 13, and 14 set). This is one of the octal values given in the discussion of standard disk boots in Chapter 4.

For all types of moving head disks, bits 1, 2, and 3 can be used to control the amount of memory that the BOOT program can use for reading in PRIMOS II. See the discussion of bits 1, 2, and 3 in the section describing the fixed head disk above.

This bootstrap waits until the disk drive is ready, and retries if status errors occur.

MT (5): The bootstrap starts up the tape drive, ensures that the tape is set at a loadpoint, and reads one tape record into memory locations '220 through '7777 (4K). Execution begins at '1000 via a JST '777 instruction.

The record read in by the bootstrap contains the MTBOOT program. When booting from magnetic tape, you may need to read in MTBOOT more than once, each time with different octal values. Use the instructions in the section on the standard magnetic tape boot, later in this section, as a guideline. Modify the octal values given there as necessary, using the information provided here and in the section on MTBOOT (later in this appendix).

Bits 1-7 specify the number (on the logical tape) of the file to be loaded, the first file being number 1. Calculating the position of a file on tape is easiest if the tape has been indexed by MAGSAV. The numbers of the files correspond to the order of files in the index. Note that if subdirectories have been saved, each subdirectory itself has a file on the tape.

If bits 1-7 are 0, MTBOOT produces the prompt

    TREENAME=

and waits for you to supply the pathname of the file to be loaded. (See the section on MTBOOT for details on valid pathnames.)

Bits 8 and 9 control the relocation of MTBOOT, if relocation is necessary and if bit 13 is 0. (MTBOOT relocates itself out of the way of the program that is to be loaded, if necessary.) The following table shows how bits 8 and 9 determine the new ending address of MTBOOT:

Bit 8	Bit 9	New Ending Address of MTBOOT
0	0	End of physical memory
0	1	16K
1	0	32K
1	1	48K

If bit 13 is 1, bits 8 and 9 must both be 0. Relocation is then controlled by the A-register of the save vector of the program being read in (see below).

If bit 10 is 0, MTBOOT automatically starts the newly loaded program. If bit 10 is 1, MTBOOT halts with a code of '000001 after loading in the program, and you must start the program yourself. (The halt code is displayed at the supervisor terminal or in the control panel data lights.)

If bit 11 is 1, MTBOOT halts at '260 to allow the SOC and OPTION A control words to be adjusted for a non-30 cps supervisor terminal. (30 cps is the default terminal speed.) The method for adjusting these control words is described in the section on MTBOOT, below.

Bit 12 selects between 7- and 9-track drives. If bit 12 is 0, a 9-track drive is assumed; if bit 12 is 1, a 7-track drive is assumed.

Bit 13 determines how relocation of MTBOOT will be controlled. If bit 13 is 0, relocation is controlloed by bits 8 and 9, as described above. If bit 13 is 1, the new ending address of MTBOOT is determined by the A-register of the save vector of the file being loaded. In addition, if bit 13 is set, the SOC and OPTION A control words are copied into locations '1004 - '1006 before the loaded program is started. (Location '725 is always 0.) The save vector keys are copied into the save vector A-register. (This process emulates the action of the disk boot.)

If your computer has sense switches rather than a VCP, switches 8-10 and 12-13 must be left in position until the selected file has been loaded. The other switches may be reset any time after the pathname prompt has been issued or the tape search has started.

<u>DISKETTE (Floppy) (6)</u>: The bootstrap reads record 0 (the first record of BOOT) into memory starting at location '770. To maintain IBM compatibility, the bootstrap alternately tries to read a 448-word PRIMOS record and a 64-word IBM record. Execution then begins at location '1000 via a JST '777 instruction.

The bootstrap waits until the drive is ready, and retries if status errors occur.

<u>SPARE (7)</u>: Intended as a user-supplied down-line loader. Currently, halts at location '57.

Keyed-In Substitutes for Control Panel Boots

If you ordered a Prime computer without the auto-load control panel PROM function, hand keyed-in programs are necessary. Because programs keyed in are likely to disappear after one use, these programs should be as short as possible.

THE BOOT PROGRAM ON DISKS

The BOOT program is placed on all disks by the MAKE utility. The source file is called BOOT, and resides in the top-level directory FILAID on the master disk. The runfile resides in the MFD, and is also called BOOT. Except in the case of storage modules, BOOT occupies two records (records 0 and 1). The PROM bootstrap reads in the first record, which is itself a disk input routine that loads in the other record. On storage modules, BOOT occupies just one record (record 0).

When the BOOT program is read in from a disk, it performs the following functions:

- Cleans up parity, nondestructively, throughout memory
- Sizes available memory
- Initializes the supervisor terminal
- Prompts you for the physical device from which to read in the operating system. (If bit 4 of the octal parameter is set, this step is omitted, and the operating system is read in from the device from which BOOT was loaded.)
- Attaches to the MFD on the specified device
- Attaches to top level directory DOS
- Depending upon memory size and/or bits 1-3 of the octal parameter (see the previous section), reads *DOS16, *DOS24, *DOS32, or *DOS64 into memory and starts it
- If bit 5 of the octal parameter is set, starts up PRIMOS

Note

> At Revision 16 and after, only *DOS64 is supplied; all other versions of DOS are obsolete. They are included here for archival purposes.

There are three possible outcomes of a boot operation: a successful boot, in which case PRIMOS II (or PRIMOS, if bit 5 is set) takes control; a detected error, in which case the boot returns to request which device to boot from; or an undetected error, such as nonexistent device, in which case the boot hangs. Error codes are discussed later in this appendix.

Once the boot has been successfully brought into memory by the pre-boot, it can be re-executed by pressing MASTER CLEAR (entering SYSCLR, on a VCP machine) and starting at '1000. If a status error is detected on the device, BOOT restarts automatically, waiting until the device is ready.

Building BOOT on Disk

On a storage module, BOOT is stored as a normal PRIMOS II SAVE file in a normal PRIMOS II-format record (record 0). Consequently, physical record 0 contains (in the order listed):

- A 16-word PRIMOS II record header
- A 9-word PRIMOS II SAVE file header
- The BOOT program itself

The record is read starting at location '760.

On a non-storage module device, BOOT occupies two records. The PRIMOS II record header is 8 words long. Record 0 is read starting at '770, but execution begins at '1000, the first word of the 9-word SAVE file header.

The SAVE file header is as follows:

Word	Meaning
0	Start address (SA = '3011)
1	End address (EA = '4401)
2	Program counter (PC = 0)
3	A-register (0)
4	B-register ('110)
5	X-register ('27)
6	Keys ('74006)
7	Unavailable
'10	Unavailable

Since execution starts at '1000, the start address must be '3011, which is also a JMP '1011 (since BOOT is guaranteed to be executed in 16S mode either by MASTER CLEAR or the control panel PROM bootstrap). BOOT actually executes in sector '1000 and so must be loaded there (at '1011), then later moved (PSD or TAP) to '3011 and saved there. Because BOOT can never be executed as a supervisor terminal command (it cannot execute in sector '3000), the PC, A-register, B-register, X-register, and Keys are available as constants to be used by BOOT. They have been defined as follows:

PC	= 0	Not used
A	= 0	Not used
B	= '110	Option A master clear default control register
X	= '27	SOC master clear default control register 1
Keys	= '74006	SOC master clear default control register 2 Left byte ('170) = SOC control word 2 Right byte (6) = number of end-of-line delay characters. (This is not used by the bootstrap but is passed by PRIMOS II to set the supervisor terminal typewriter control words.)

Once BOOT has been placed on a disk, it can be copied to another disk with the following command sequence:

```
A MFD XXXXXX 0
RESTORE BOOT
A MFD XXXXXX 1
SAVE BOOT
```

Note

BOOT parameters can be changed to accommodate other terminal characteristics, either permanently (by saving BOOT with new B, X, and Keys parameters) or temporarily (by patching locations '1004, '1005, and '1006 of the copy in memory). Be careful to change only one value at a time.

The file *BOOT is produced as a result of running the command file C_MAKE. *BOOT is a copy of the boot exactly as it appears on record 0 of a storage module. *BOOT can be restored and saved in the same way in which the file BOOT in an MFD can.

THE MTBOOT PROGRAM ON MAGNETIC TAPES

Booting from magnetic tape involves more steps than booting from a disk. Tape boots start up PRIMOS II, but not PRIMOS; after a tape boot, you must start PRIMOS yourself (as described in Chapter 4). The standard instructions for performing a tape boot are listed in a later section of this appendix.

The MTBOOT program is used to read a specified file from tape into main memory. You must know either the pathname of the file or its number on the logical tape being used. (If you do not specify a file number in bits 1-7 of the octal parameter in the pre-boot, MTBOOT prompts you for a pathname.) The file to be loaded must be in standard save-file format.

The MTBOOT program is saved by the MAGSAV utility in each logical tape header record of a magnetic tape. It is now possible to boot from any magnetic tape produced by MAGSAV Rev 14.1 or later.

When MTBOOT is read in from a tape, the following events occur:

- MTBOOT sizes available memory.

- If the terminal speed selection control words (the SOC and OPTION A control words) are to be patched, MTBOOT halts at '260 to allow you to do so. You request this halt by setting bit 11 in the octal parameter (see the previous section of this appendix). For instructions on making the patch, see below.

- If you have specified a file number in bits 1-7 of the octal parameter, MTBOOT searches to the end of the logical tape for the specified file.

  If bits 1-7 are 0, MTBOOT issues a prompt for a pathname and searches to the end of the logical tape for the file you specify. (The pathname you supply must be a series of up to 8 valid file system names — with no embedded blanks — separated by ">" signs. If you enter a null line, a question mark, or any character whose value is less than '240 — for example, a control character — the prompt is repeated.)

  In either case, if the file is not found, MTBOOT halts with code '000005.

- When the requested file is found, the first data block is read and the save vector examined. The file must have a starting address above '6207 <u>or</u> an ending address below the memory size minus '5423 (e.g., '172355 for a 64K system). If necessary, MTBOOT relocates itself out of the way of the program being loaded (thus adjusting effective memory size). If relocation is necessary but impossible, MTBOOT halts with code '000003.

  If bit 13 of the octal parameter is 0, relocation is controlled by bits 8 and 9, as described in the previous section of this appendix. If bit 13 is 1, relocation is controlled by the A-register of the save vector of the file to be loaded.

- If bit 10 of the octal parameter is set (1), MTBOOT halts with code '000001 displayed at the supervisor terminal or in the data lights. If bit 10 is reset (0), the program starts via a JMP instruction. If 13 is set (1), locations '1004 - '1006 are overwritten with the OPTION A and SOC control words.

If MTBOOT encounters an error when reading from the tape, it halts with the status word in the data lights or displayed at the supervisor terminal. The halt codes are explained in the last section of this appendix.

Loading From a Prime 300 7-Track Tape: The MTBOOT program must be manually started if it is loaded from a 7-track tape generated on a Prime 300. When the machine halts following the load from tape, start at '400. (Load '400 into location '7 and run.)

Loading From a Logical Tape Other than the First: To load from a logical tape other than the first, you must skip over the first logical tape. Do this by specifying a nonexistent file number or pathname. When MTBOOT halts with code '000005 displayed at the supervisor terminal or in the data lights, reset the switches (if necessary) and restart at '400.

Running on a Non-30 CPS Terminal: If

- The supervisor terminal is to be used to enter a pathname, and
- The supervisor terminal will not run at 30 cps, and
- MTBOOT has not yet been modified for the appropriate speed (see the section on modifying MTBOOT, below)

then you can temporarily patch MTBOOT to accommodate your terminal. If you set bit 11 of the pre-boot octal parameter before starting the boot procedure, MTBOOT will halt at '260 with '21 in the data lights (or displayed at the supervisor terminal) after sizing memory. You can then patch locations '220 - '222 as follows:

Speed	'220	'221	'222
110 baud	110	27	74006
300 baud	1010	76	34006
1200 baud	2010	373	34006
9600 baud	3410	3735	34006

After performing the patch, select RUN and press START to continue.

When booted in from tape, PRIMOS II also assumes a terminal speed of 30 cps. If bit 13 of the octal parameter is set, then MTBOOT copies its SOC and OPTION A control words into PRIMOS II, as in the case of a disk boot, so you do not have to do so yourself. If you do not wish to set bit 13, then you can reset the speed in PRIMOS II as follows:

1. When PRIMOS II (*DOS64) is being loaded into memory, set bit 10 of the pre-boot octal parameter. MTBOOT will then halt with

'000001 in the data lights (or displayed at the supervisor terminal) after loading in PRIMOS II.

2. When MTBOOT halts, load '4000 into the A-register and set locations '1004, '1005, and '1006 to the appropriate values from the table above.

Note

Under PRIMOS II, the low-order 8 bits of '1006 also give the number of delays to be used after carriage returns.

3. Press START. PRIMOS II will reset the terminal to the desired speed.

This procedure is necessary each time PRIMOS II is reloaded from tape.

Modifying MTBOOT

The source of the magnetic tape boot program is MTBOOT.PMA. You may wish to modify the code for nonstandard tape buffer sizes or terminal speeds. Once the code is modified and assembled, you can reload it by running the program MAGSR>LBOOT.SAVE. This program incorporates MTBOOT.SAVE into the MAGSAV runfile. That is, LBOOT.SAVE restores MAGSAV.SAVE, overlays MTBOOT.SAVE, and saves MAGSAV.SAVE. (This process occurs automatically when MAGSAV is built by MAGSAV.BUILD.CPL.)

The supplied version of MTBOOT handles up to 2K-word tape blocks. You can decrease the tape buffer size by decreasing the variable MCOUNT in the source code.

The supplied version of MTBOOT initiates the supervisor terminal for a speed of 30 cps. If your supervisor terminal does not run at 30 cps, you can either patch MTBOOT as described in the previous section, or modify it permanently by issuing the following commands:

```
REST MTBOOT.SAVE
SA MTBOOT.SAVE 4/<value_1> <value_2> <value_3>
R LBOOT.SAVE
```

Choose value_1, value_2, and value_3 from the table of patch values in the section on running on a non-30 cps terminal, above. Value_1 through value_3 correspond to the values for locations '220 through '222, respectively.

MAGSAV Output Buffer: LBOOT.FTN assumes that the output buffer for the logical tape header record in MAGSAV is located at '4000. If the location of this buffer (OUTBUF) changes, the parameter OUTBUF in LBOOT.FTN should be changed to LOC(OUTBUF) + '14.

THE STANDARD MAGNETIC TAPE BOOT

This section explains how to perform a standard boot from magnetic tape. If you must modify these instructions, or if you need more details on magnetic tape boots, refer to the relevant sections on the pre-boot and the MTBOOT program, earlier in this appendix.

Three sets of instructions are supplied in this section:

- Creating a tape that can be used to boot the system
- Booting from tape on a computer with a control panel
- Booting from tape on a computer with a Virtual Control Panel (VCP)

The instructions given here assume that you are booting from a 9-track tape drive and that your supervisor terminal runs at 30 cps. If either assumption is false, you will need to modify the octal values supplied in these instructions. Refer to the sections on the pre-boot octal values and the MTBOOT program, earlier in this appendix.

Creating a Tape for Booting the System

If your master disk is intact, all you need on your boot tape is PRIMOS II. However, if you must remake and restore your master disk, you will need to have MAKE, MAGRST, and possibly a few other utilities on your tape. The following is one scenario for creating a tape from which to boot the system and remake the master disk:

```
OK, /* Save the necessary files in a scratch directory
OK, CREATE WORK
OK, A *>WORK
OK, COPY DOS>*DOS64
OK, COPY CMDNC0>(MAKE MAGRST COPY_DISK PHYRST)
OK, AS MT0          /*Assign the tape deck
Device MT0 assigned.
OK, /* Save the contents of the WORK directory.  Take an index
OK, /* so that you will know which file numbers correspond
OK, /* to which files.
OK, MAGSAV
[MAGSAV Rev. 19.00]
Tape unit (9 Trk): 0
(Tape not at load point)
```

```
Enter logical tape number: 1
Tape name: BOOT
Date (MM DD YY):
Rev no: 19
Name or command: $I 7
Name or command: *
*** Start of Save ***
*DOS64 (sam)
MAKE (sam)
MAGRST (sam)
COPY_DISK (sam)
PHYRST (sam)
***End of Save ***
```

To minimize the risk of read errors, save the tape with the minimum number of required utilities. The whole of PRIMOS could be saved in a separate logical tape, or PRIMOS could be restored from another backup tape.

Booting from Tape on a Computer with a Control Panel

Use the procedure described below to boot from tape if your system has a control panel rather than a VCP.

Load PRIMOS II Into Memory:

1. Mount the magsaved tape on drive 0. The tape should be at load point or beyond.

2. Put the tape drive on line.

3. Set the sense switches to '15 (switches 13, 14, and 16 up; all others down).

4. Set the rotary switch to LOAD.

5. Press START. When the system types "TREENAME=", enter the pathname of PRIMOS II exactly as it appears on the index of the magtape. Typically:

   ```
   TREENAME= *DOS64
   ```

   The system responds with:

   ```
   PRIMOS II REV. 19.0 02/17/82   (AT 170000)
   OK:
   ```

6. If your master disk is intact, respond to the PRIMOS II "OK:" prompt by starting up the disk and attaching to its MFD:

   OK: STARTUP nnn
   OK: A MFD xxxxxx

   (nnn is the disk pack's physical device number)

   If you need to remake and restore your master disk, continue with Step 7 when PRIMOS II issues the "OK:" prompt.

Load MAKE Into Memory:

7. Set the rotary switch to STOP.

8. Press MASTER CLEAR.

9. Set the sense switches to '505 (switches 8, 10, 14, and 16 up; all others down).

10. Set the rotary switch to LOAD.

11. Press START. When the system types "TREENAME=", enter the pathname of MAKE exactly as it appears on the index of the magtape. Typically:

    TREENAME= MAKE

    The system halts with a '1 in the data lights:

    HALTED AT 072427: 000001

Run MAKE:

12. Set the rotary switch to STOP.

13. Press MASTER CLEAR.

14. Set the sense switches to '170000 (switches 1 through 4 up, all others down).

15. Set the rotary switch to LOAD.

16. Press START. PRIMOS II responds with the prompt "OK:". Enter "S 1000" to start MAKE, and enter the physical disk number when MAKE requests it:

```
OK: S 1000
GO
*** MAKE *** <Rev. 19.0>
PHYSICAL DISK: nnn
     :
     :
DISK CREATED
OK:
```

Load MAGRST Into Memory:

17. Set the rotary switch to STOP.

18. Press MASTER CLEAR.

19. Set the sense switches to '505.

20. Set the rotary switch to LOAD.

21. Press START. When the system types "TREENAME=", enter the pathname of MAGRST exactly as it appears on the index of the magtape. Typically:

    TREENAME= MAGRST

    The system halts with a '1 in the data lights:

    HALTED AT 072427: 000001

Run MAGRST:

22. Set the rotary switch to STOP.

23. Press MASTER CLEAR.

24. Set the sense switches to '170000.

25. Set the rotary switch to LOAD.

26. Press START. PRIMOS II will respond with the prompt "OK:". Enter the STARTUP command with the disk device number of the master disk, and then attach to the MFD on that disk and start at location '1000:

```
OK: STARTUP nnn
OK: A MFD XXXXXX
OK: S 1000
```

```
[MAGRST 19.00]
   .
   .
   .
*** Restore Complete ***
OK:
```

Booting from Tape on a Computer with a VCP

Use the following procedure to boot from tape if your system is equipped with a VCP:

1. Mount the magsaved tape on drive 0. Make sure the tape is at or beyond load point.

2. Boot PRIMOS II:

   ```
   CP> SYSCLR
   CP> BOOT 15
   TREENAME = *DOS64

   PRIMOS II REV. 19.0 02/17/82 (AT 170000)
   OK:
   ```

3. If your master disk is intact, respond to the "OK:" prompt by starting up the master disk and attaching to its MFD:

   ```
   OK: STARTUP nnn
   OK: A MFD xxxxxx
   ```

   (nnn is the physical device number of the disk you wish to start up.)

   If you need to remake and restore your master disk, continue with Step 4 in response to the "OK:" prompt.

4. Press the ESCAPE key twice to return to control panel mode:

   ```
   CP> <ESC><ESC>
   ```

5. Load MAKE:

   ```
   > STOP
   CP> SYSCLR
   CP> BOOT 505
   TREENAME = MAKE

   HALTED AT 072427/000001
   ```

6. Restart PRIMOS II:

   ```
   CP> BOOT 170000
   ```

7. When the PRIMOS II prompt appears, start MAKE:

```
OK: S 1000
*** MAKE *** <Rev. 19.0>
PHYSICAL DISK: nnn
    :
    :
DISK CREATED
OK: <ESC><ESC>
```

8. Load MAGRST:

```
>STOP
CP> SYSCLR
CP> BOOT 505
TREENAME = MAGRST

HALTED AT 072427/000001
```

9. Restart PRIMOS II:

```
CP> BOOT 170000
```

10. When the PRIMOS II prompt appears, start MAGRST:

```
OK: STARTUP nnn
OK: A MFD xxxxxx
OK: S 1000
[MAGRST 19.00]
      :
      :
***Restore Complete***
OK:
```

The contents of the boot tape can now be restored, or a backup tape can be mounted and restored. To run another R-mode program from tape, repeat the procedure indicated in Steps 5-7 (for MAKE) and 8-10 (for MAGRST).

ERRORS

This section explains the error codes that may appear at the supervisor terminal or in the control panel data lights during a boot.

Disk Boot Errors

While BOOT is being loaded in from disk (using the input routine in record 0), any errors detected cause a halt. The errors checked at this stage are the following:

Octal Number in Lights	Error
100	Parity
101	Machine check
102	Nonoctal physical device number
103	Bad device type
104	Bad status Option B, B', Storage Module, Diskette
105	Bad Record ID - Bad CRA (High-Low)
106	Incompatible BOOT records

Parity Error or Machine Check Error (100, 101): If a parity or machine check error occurs while the BOOT program is being loaded, a halt occurs with code 100 or 101 (respectively). Parity and machine check errors are caught by the hardware. No further information is available on the P100, P200, or P300. Additional information can be found in the diagnostic status word on the higher CPUs. After the memory test, the error message PARITY ERROR or MACHINE CHECK is displayed. If the errors persist, the messages persist.

Nonoctal Physical Device Number (102): The message OCTAL ONLY is displayed if you enter a nonoctal character for the physical device number. The PHYSICAL DEVICE= prompt is issued again at the supervisor terminal.

Bad Device Type (103): This code appears in the data lights if a device type of 7 is detected. The PHYSICAL DEVICE= prompt is issued again at the supervisor terminal.

Bad Status (104): Whenever bad status is detected, the status is stored in location '40. During the first phase, the loading of BOOT itself, a halt then occurs with code 104. If a bad status is detected while PRIMOS II is being loaded, the message BAD STATUS is printed, followed by the status word.

Bad Record Id (105): As each record is read, the requested record address is checked against the actual record address. If these addresses do not match, a halt occurs with code 105. The requested address is in locations '723 and '724 and the actual address is in locations '760 and '761.

When PRIMOS II is being searched for or loaded, the message BAD RECORD ID, rrrrrr rrrrrr ffffff ffffff is printed, where the words indicated by rrrrrr are two words of requested octal address and the words indicated by ffffff are two words of actual octal address. The PHYSICAL DEVICE= prompt is issued again at the supervisor terminal.

Incompatible BOOT Records (106): The first and second records are checked to see if they come from the same version of the BOOT program. They may come from different versions if an old (control panel) CPBOOT (which always reads from unit 1) gets the first record of a new (disk) BOOT. The new BOOT gets its second record from the unit designated by switches 8 and 9.

The second and subsequent records may therefore come from a different version of BOOT. If such an incompatibility is recognized, BOOT halts with code '106.

'FILE' Not Found: If the required version of DOS or the DOS UFD is not on the requested physical device, the message 'FILE' NOT FOUND is displayed, where 'FILE' is the name of the requested file. The PHYSICAL DEVICE= prompt is issued again at the supervisor terminal.

Memory Test Failure: While memory is being tested, if the test patterns written and read do not match, the message MEM TEST MISMATCH LOC xxxxxx is displayed, where xxxxxx is the location of the word being tested. If either a parity error or a machine check is detected, the address of the word being tested is displayed, along with the message PARITY ERROR or MACHINE CHECK. The PHYSICAL DEVICE= prompt is issued again at the supervisor terminal.

Tape Boot Errors

The following halt codes may appear in the data lights during a boot from magnetic tape:

000001 (Halt at '635 or '175175) The requested program has been loaded and auto-start has been suppressed (bit 10 of the pre-boot octal parameter set). On a machine without a VCP, pressing START and typing START <address> starts the loaded program. On a VCP machine, perform an auto-start bootstrap and type START <address>. (At the halt, A, B, X, etc. have been loaded from the program's save vector.)

072642 (Halt at '503) The pathname record for the requested file was not followed by a data record.

000003	(Halt at '641) The requested program cannot be loaded by the boot program: Its starting address is less than '6207 and its ending address is above the memory size minus '5423.
000005	(Halt at '440) The end of the logical tape was encountered before the requested program was found.
000021	(Halt at '260) Bit 11 of the pre-boot octal parameter was set, causing the boot to halt so that the SOC and OPTION A control words can be patched for a non-30 cps terminal.
xxxxxx	(Halt at '1) A tape read error or unexpected status condition occurred. The data lights display the last status word read from the controller. Status bit definitions are as follows:

100000	Parity error
040000	Runaway tape
020000	CRC error
010000	LRC error
004000	Low DMX range
002000	Permanent error
001000	Read-after-write error
000400	File mark detected
000200	Ready (OK)
000100	Online (OK)
000040	End of tape detected
000020	Rewinding
000010	Beginning of tape
000004	Protected (OK)
000002	Overrun
000001	Rewind complete

B
Virtual Control Panel

This appendix describes the Virtual Control Panel as of Software Revision level 18.1. The VCP is a dynamically evolving product. For the latest details, see your local Prime sales office or Prime system analyst.

INTRODUCTION

The Virtual Control Panel (VCP) expands the functionality of the supervisor terminal to include that of the control panel (described in Appendix C). The control panel functions are implemented by keyboard commands (as in an interactive debugging program). The VCP has two operating modes: system terminal and control panel. In system terminal mode, the VCP performs all the functions of the supervisor terminal: bringing up PRIMOS, sending messages, logging out users, etc. In control panel mode, the VCP has all the functionality of the control panel: bootstrapping PRIMOS II, interrogating and setting sense switches, performing register set operations, etc. The VCP also has two levels of remote access (described below), allowing diagnostics and system operations to be performed off-site.

The VCP replaces the System Option Controller (SOC). It provides an asynchronous interface for the supervisor terminal as well as a serial printer interface.

USE OF THE VIRTUAL CONTROL PANEL (VCP)

Controls on the VCP are a two-position key lock, four indicators, and four buttons, summarized in Table B-1.

Table B-1
Controls on the Virtual Control Panel (VCP)

Switch	Function	Indicator Light ON
Power	Turns power ON/OFF	Power ON
Key lock	Locks/unlocks next 3 switches	- - -
Master Clear	Initializes system	Stop (processor halted)
Remote enable	Permits remote access to system via VCP	Enabled
Remote privilege	Selects remote privilege level	Monitor (steady) Full (blinking)

The button with a protective cover turns system power on and off. The indicator above this button shows the power status.

The key locks or enables the three function selection buttons.

The master clear button initializes the system. Its associated indicator is lit when the processor is halted.

The remaining two buttons control remote access. The first of these two buttons is used to enable remote access. Whenever remote access is enabled its indicator will be lit.

The final button controls the extent to which a remote user may control the system. A remote user may be given full privileges (equivalent to the supervisor terminal) or monitor privileges. With monitor privileges, anything typed on the local supervisor terminal is seen by the remote user. Anything typed by the remote user will be displayed at the local supervisor terminal. However, this will not affect the machine or VCP state. Whenever a remote access is in progress, the remote active light is lit. If the remote privilege button has been pushed giving the remote user full privileges, the remote active light blinks to draw attention to this.

CONTROL PANEL MODE

This mode of operation expands the capabilities of the control panel of earlier models and adds flexibility and ease of use by displaying and accepting data from a terminal. Many control panel functions are similar to those of a software debugger. Data may be accepted and displayed in hexadecimal, ASCII, binary, decimal, and octal. Commands are entered with key words instead of by pressing switches and turning a knob.

On power-up and master clear, the VCP executes self-verify routines to insure its own integrity. The VCP sequences the lights on the cabinet indicating the progress of the internal tests. If an error occurs, the lights indicate what test failed; an error message is printed on the terminal. Upon successful completion of VCP internal verify routines, the VCP releases the signal HSYSCLR and checks to see that the central processor master cleared. If the central processor fails micro-verify, an appropriate error message is printed indicating the failure. If the central processor passes its integrity test, the VCP informs the operator by printing on the terminal:

    *** CPU VERIFIED ***

If the VCP fails to print any message (hung condition), type CONTROL-P to allow the VCP to examine and print data. The hung condition can result from a machine check.

The control panel mode prompt character, CP>, is printed to indicate the VCP's readiness to accept command input. Control panel mode can be entered from supervisor terminal mode by typing ESC ESC (ESCAPE key twice). Two characters may follow the CP> prompt character. This indicates that a message is being sent to the supervisor terminal.

Note

When in control panel mode, the VCP does not have auto restart.

Summary of Control Panel Mode Commands

Whenever an address is required in a command, either segno/wordno or wordno may be used. segno is the segment number; wordno is the word number in the segment. The VCP remembers the last segment number referenced as the current segment and will use this current segment if segno is not explicitly specified. The initial value of the current segment is segment 0.

General Commands: The following commands control the general operation of the Virtual Control Panel:

STOP

Halts the central processor unit.

SYSCLR

Performs a master clear.

VIRY

Performs a SYSCLR and then runs diagnostics to verify the VCP.

Supervisor Terminal Output: While the terminal is in control panel mode, the supervisor terminal output (login messages, etc.) may be either ignored (lost), buffered, or interleaved with control panel output. Output is regulated by the following commands:

SYSOUT BUFF

Buffers supervisor terminal output and prints this when system terminal mode is re-entered (default).

SYSOUT IGN

Ignores supervisor terminal output while in control panel mode.

SYSOUT INT

Interleaves supervisor terminal output with control panel mode output.

Sense Switches: The following commands change the sense switch settings: BOOT (with argument), DISPLAY, DISPLAYC, RCP (with argument), RUN (with argument), SS, A, C, D, and F.

Data Representation: Data are represented in five formats:

Format	Display Mode
:O	octal
:H	hexadecimal
:B	binary
:D	decimal
:A	ASCII

If two specifiers are used (for example, :B :H), the first refers to data and the second to address display. :B :H means display binary data and hexadecimal addresses. These specifiers are also valid after the Dump and Access commands. Note, however, that the address mode specification will not take effect until the next command. For example,

    D 1000 2000 :A :O

dumps, in ASCII, from the address as specified in the address mode at the time the Dump command was given. 1000 could be octal, hexadecimal, decimal, or binary. :A is not valid for address display. The default is :O :O.

Control Panel Functions: The following commands perform the basic operations of a control panel:

    BOOT

Places VCP in auto-boot condition.

    BOOT number

Boots with sense switches set to number. BOOT '114 bootstraps from a storage module. The data mode must be octal.

    DISPLAY address

Displays contents of address. Operates only when PRIMOS is running.

    DISPLAYC address

Continuously displays contents of address. The value is displayed each time it changes. Operates only when PRIMOS is running. Halt operation by a CONTROL-P.

    FETCH

Fetches data according to the previously set sense and data switches.

    LIGHTS

Displays the current lights.

LIGHTSC

Displays current lights continuously. The lights are displayed each time they change. Halt operation by a CONTROL-P.

RCP location

Identical to RUN (see below), except that the VCP stays in control panel mode. location may be a virtual address if the CPU is running in segmented mode.

RUN location

Puts location into PB and starts the CPU. If location is not supplied, the current value of PB is used. Automatically enters supervisor terminal mode (see RCP above).

SD number

Sets the data switches to the value of number for one INA '1720 only.

SS number

Sets the sense switches to the value of number.

SSTEP n

Single steps n locations. The value of n depends upon the data representation; for example, SSTEP 10 in octal specification steps 8 locations.

STEPU n

Steps until address is equal to n.

STORE number

Stores the value of number into the location specified by the previously set sense and data switches.

Memory Display Commands: The following commands allow accessing of the central processor registers and memory:

MO ABS

Sets VCP to reference absolute (physical) memory.

MO MAP

Sets VCP to reference mapped memory (default).

MO RFABS

Sets VCP to reference register file absolute.

MO RFCRS

Sets VCP to reference register file current register set.

MO RFH

Sets VCP to display/modify high side of register file.

MO RFL

Sets VCP to display/modify low side of register file.

Note

When register file mnemonics are used, both high and low sides are displayed. The high/low mode determines which side is modified by the Access command.

After a mode has been determined, the following commands may be used: (A list of legal register names is given in Table B-2.)

A n

Access address n. The address may be followed by data display specifiers.

Table B-2
Register File Mnemonics Accepted by the VCP

Mnemonic	Register Description
A	Accumulator
B	Double-precision and long accumulator extension
DSWPARITY	Diagnostic Status Word Parity (P750 only)
DSWPB	Diagnostic Status Word Procedure Base
DSWRMA	Diagnostic Status Word RMA
DSWSTAT	Diagnostic Status Word Status
DTAR0	Descriptor table address: segments 0 to '1777
DTAR1	Descriptor table address: segments '2000 to '3777
DTAR2	Descriptor table address: segments '4000 to '5777
DTAR3	Descriptor table address: segments '6000 to '7777
E	Accumulator extension for MPL, DVL
FADDR	Fault address
FAR0	Field address register 0
FAR1	Field address register 1
FCODE	Fault code
FLR0	Field length register 0
FLR1	Field length register 1
GR0	General Register 0
GR1	General Register 1
GR2	General Register 2
GR3	General Register 3
GR4	General Register 4
GR5	General Register 5
GR6	General Register 6
GR7	General Register 7
KEYS	Process status information
L	Combined A and B registers
LB	Link Base
MODALS	Process status information
OWNER	Address of PCB of process owning register contents
PB	Procedure base
PBSAVE	Saved return pointer when return pointer used elsewhere
PPA	Pointer to process A
PPB	Pointer to process B
PSWPB	Process Status Word Procedure Base
RECC1	ECC error register 1
RECC2	ECC error register 2
REOIV	Register End Of Instruction Vector
RSAVPTR	Register Save Pointer: location of Register Save after halt
RSGT1	Register Segmentation Trap: Address of page map/SDW2
RSGT2	Register Segmentation Trap: Contents of page map/DSW2
S	Stack
SB	Stack base
TIMER	1-millisecond process timer (used for time-slice)
VSC	Visible shift counter
X	Index
XB	Temporary (auxiliary) base
Y	Alternate index

A register-name

Access register-name. The current high/low mode determines which side of the register is modified. The name may be followed by data display specifiers.

The following are legal responses to the Access command:

Response	Meaning
CR (carriage return)	Access next location.
^ (uparrow)	Access previous location.
number	Modify location to value of number.
/ (slash)	Exit and return to control panel mode.

C start end to

Copy the block starting at start and ending at end to the block starting at to. Overlapping blocks where start<to<end are not allowed.

D start end

Dumps from start to end. Data display specifiers may follow (e.g., D 100 200 :H :O).

D register-name

Dumps both high and low sides of register-name. Data display specifiers may follow the name.

Long dumps may be terminated with a CONTROL-P.

F start end number

Fill the block from address start to address end with number.

Exiting Control Panel Mode: To exit control panel mode, use the following command:

MO ST

Enters supervisor terminal mode from control panel mode.

SYSTEM TERMINAL MODE

This mode of operation is the standard system (supervisor) terminal. After the system has successfully master cleared and a BOOT command is issued, the VCP awaits the standard PRIMOS commands. To exit from system terminal mode to control panel mode, type ESC ESC (ESC is the ESCAPE key).

USE OF THE VCP: EXAMPLES

The examples below illustrate VCP use in a few common situations.

Cold Start After System Shutdown

To cold start a system running under PRIMOS, issue the commands:

```
OK, SHUTDN ALL
REALLY? YES

HALTED AT:     halt-address
```

This causes the CPU to halt. To cold start PRIMOS type:

```
CP> SYSCLR
```

or, for a storage module,

```
CP> BOOT 14114
```

Proceed as described in Chapters 4 and 13.

Warm Start

If the CPU halts, the message:

```
HALTED AT:     halt-address

CP>
```

will appear (see Chapter 13 for complete details). Determine and log any hardware problems by typing:

```
CP> D DSWSTAT
                    (value)
CP> D DSWRMA
                    (value)
CP> D DSWPB
                    (value)
CP> D DSWPARITY     for P750 CPU only
                    (value)
CP> SYSCLR
CP> RUN

HALTED AT:     1001:  000010

CP> RUN
```

*** WARM START ***

Proceed as described in Chapter 13.

Tape Dump

Mount a scratch tape on unit 1, and type the following:

```
CP> SYSCLR
CP> RUN 776
```

C
Control Panel

INTRODUCTION

This appendix describes operation of Prime central processors (CPUs) from the control panel. All Prime computers are now shipped with Virtual Control Panels (see Appendix B).

The control panel procedures of most importance to the operator in a multi-user environment are:

- Using autoload to bootstrap the operating system during startup
- Examining the contents of register and memory locations following an operating system crash
- Setting up for automatic restart after a power failure

In addition, while running single-user PRIMOS II, the operator must start and stop program operation from the panel.

Other control panel capabilities such as program patching and single step operation are primarily useful to field service technicians, but are included in this Appendix for reference purposes.

CONTROL PANEL FEATURES

All Prime control panels have the same physical switch and indicator layout (see Figure C-1). In general, operation of all panels is identical, except that the Prime 350 and above have additional capabilities permitting:

- Accessing register sets
- Addressing real memory using a full 22-bit address
- Addressing mapped memory when segmentation is enabled

These advanced features are enabled by sense switch settings that have no effect in the Prime 300 and below.

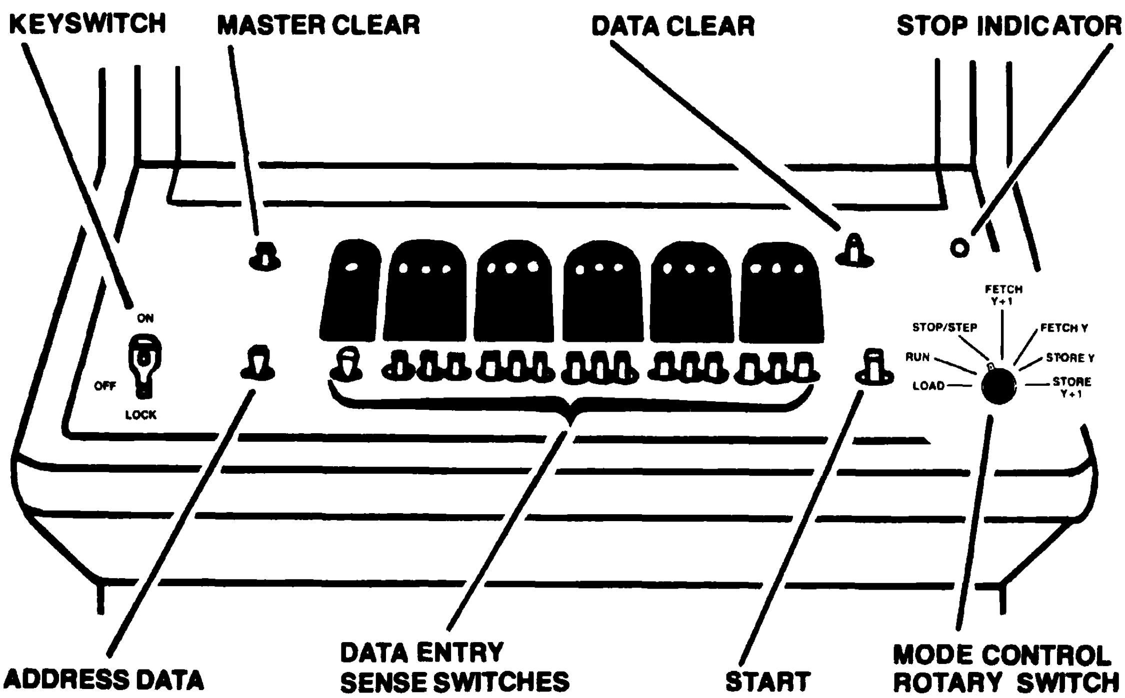


Figure C-1
Prime Control Panel Switches and Indicators

RUN FUNCTIONS

RUN mode is the normal control panel state for execution of programs. In RUN mode, the control panel has no effect on processor operation, but the 16 sense switches and indicators act as programmed input/output devices. A hardware register in the panel controls the indicators; it can be loaded with data from the A-register by an OTA instruction. The sense switches can be read into the A-register by an INA instruction, or can be tested by SKS instructions.

Data transferred to the indicator register from the CPU is displayed when the ADDRESS/DATA switch is in the DATA position. In the ADDRESS position, the indicators present a dynamic display of whatever is present in the CPU's BMA (memory address) bus.

Single Step Functions

Immediately after power-up or whenever the rotary switch is turned to STOP/STEP, the CPU is halted (STOP light is on), and the control switches on the panel enable the operator to access, modify, or execute single 16-bit cells of high-speed memory (including the register files).

Access (FETCH): The memory cell specified by the indicator display when the ADDRESS/DATA switch is at ADDRESS is accessed by turning to FETCH and pressing START. The content of the cell is then displayed by the indicators when the ADDRESS/DATA switch is at DATA.

Modify (STORE): The contents of the addressed memory cell can be modified by setting up a new data display, turning to STORE, and pressing START.

Consecutive FETCH or STORE: The FETCH Y+1 and STORE Y+1 functions permit accessing or modifying of consecutive memory locations. The address is incremented by 1 automatically when the START switch is pressed before the operation is carried out.

Single-Step Execution: While the rotary swtich is in STOP/STEP, each operation of the START switch executes the instruction designated by the content of the P-register, memory location 7. All levels of indirect addressing and indexing are carried out, and the instruction is executed. The address of the next instruction to be executed is loaded into the P-register (and displayed on the indicators if ADDRESS/DATA is at ADDRESS). After executing each instruction, the operator can FETCH or STORE other registers or memory locations. When the ADDRESS/DATA switch is at DATA, the indicators display the data or instruction in the last accessed location.

LOAD Functions

The LOAD position of the rotary switch enables Automatic Program Load (APL), which simplifies program loading and permits automatic restart (or reload) after a power failure.

Other Features

Master Clear: The MASTER CLEAR switch (operative in all rotary switch positions except LOAD and RUN) places the CPU and all peripheral controllers in a known starting condition.

STOP Lamp: The STOP lamp lights when the CPU is in a halt condition (either from a panel halt or execution of the HLT instruction).

Battery Backup Indication: The STOP lamp flashes on and off if there has been a power failure and the memory is being refreshed by the backup battery pack (optional).

Keyswitch: A rotary keyswitch provides line power on/off control and a lockout position that disables the control panel except for sense switch settings and indicator displays controlled by the running program.

CONTROL AND INDICATOR FUNCTIONS

Functions of the Prime control panel switches and indicators are described in Table C-1. Sense switch functions peculiar to the Prime 350, 400, and 500 are shown in Table C-2.

SUMMARY OF OPERATING PROCEDURES

Note

> The following procedures apply to all CPUs unless otherwise specified.

Table C-1
Control Panel Functions

Item	Function	
Keyswitch	Power control and panel locking rotary keyswitch.	
Position		
OFF	Line power is removed from Prime CPU; the power supply and battery backup are inactive.	
ON	Line power is applied to the Prime CPU power supply and control panel rotary switch determines CPU mode of operation.	
LOCK	Same as ON, but control panel rotary switch has no effect on CPU run condition. (Data switches and indicators can be accessed by programmed I/O, however, and the ADDRESS/DATA switch functions normally.)	
MASTER CLEAR Switch	Momentary-down switch sets CPU registers and status to known starting condition (see section above on Master Clear).	
Address/Data Indicators	What these indicators display depends on the position of the rotary mode switch, the ADDRESS/DATA switch, and the last operation that took place.	
Rotary Switch		
Position	ADDRESS/DATA Switch Position:	
	Address	Data
RUN or LOAD	Contents of CPU's BMA (address) bus	Content of panel's indicator register (loaded by OTA '1720)

Table C-1 (continued)
Control Panel Functions

STOP/STEP	After LOAD, RUN, or single step execution	
	Address of next instruction to be executed same as contents of P-register (location 7)	Contents of next instruction location
	After FETCH or STORE operation	
	Same as FETCH or STORE	Same as FETCH or STORE
FETCH Y or STORE Y	Address of next cell to be accessed	Data last set up by data switches, or FETCH or STORE
FETCH Y+1 or STORE Y+1	One less than address of next cell to be accessed (address is incremented before FETCH or STORE takes place)	Same as FETCH or STORE
DATA CLEAR Switch	Momentary-down switch clears address or data display (selected by ADDRESS/DATA switch).	
STOP Indicator	Lights when CPU is halted either from programmed HLT instruction or panel halt (rotary switch in any position except RUN or LOAD).	
Mode control rotary switch	Seven-position rotary switch controls Prime CPU mode of operation. Active only when keyswitch is at ON.	
	Position	Description
	LOAD	Enables automatic program load sequence, a CPU option that takes the place of a key-in loader and permits automatic hardware read-in of programs from various media. Also implements automatic reload or restart after a power failure, and CPU restart at address in sense switches.

Table C-1 (continued)
Control Panel Functions

Position	Description
RUN	Normal operating state of CPU; after START switch is operated, clock runs and CPU executes sequential instructions.
STOP/STEP	Stops CPU and permits address/data changes, Master Clear, and single-step execution. Interrupt and DMX operations are disabled.
STORE Y	When START switch is depressed, content of data display is stored in register or memory location specified by address display. The address and data displays are unaltered.
FETCH Y	When START switch is depressed, content of register or memory location (specified by address display) is displayed on indicators.
FETCH Y+1	Same as FETCH, but address display is incremented by 1 before location is read. Each operation of START switch accesses the next sequential memory location for display.
STORE Y+1	Same as STORE, but address display is incremented by 1 before location is loaded. To load sequential locations, leave ADDRESS/DATA switch at DATA, enter each new data value in data switches, and press START. Next memory address is set up automatically.

Table C-1 (continued)
Control Panel Functions

START Switch	Initiates operation set up by rotary switch (RUN, LOAD, FETCH, STORE).
Sense/Data Switches	Bank of 16 toggle switches. (UP = latching; DOWN = momentary.) Momentary DOWN position of any switch sets a 1 into corresponding position of indicator display (address or data, depending on position of ADDRESS/DATA switch). Note In RUN mode, momentaries can be program tested by INA '1720. Center and latching UP position provide sense switch data to INA '1620 and sense switch skip instructions (SNS, etc.). These positions provide device identification for bootstrap LOAD operation. UP is 1 and center is 0 for each switch. Switches 1 to 16 correspond to A-register bits 1 to 16. For P350, the sense switches also make it possible to examine locations in the general register sets and to specify 22-bit absolute addresses or segment addresses (see Table C-2).

Table C-2
Register and Memory Access in the Prime 350, 400, and 500

Sense Switch		Register Access		Memory Access
SS1		UP		DOWN
SS2	UP:	SS 10-16 address the four 32-bit register sets: '0-'37 microcode scratch (RF0) '40-'77 DMA (RF1) '100-'137 User Set (RF2) '140-'177 User Set (RF3)		Not used
	DOWN:	SS12-16 address a location in the current register set (CRS)		
SS4	UP:	Address high-order half of 32-bit register	UP:	Absolute addressing of physical memory.
	DOWN:	Address low-order half of 32-bit register	DOWN:	Mapped addressing of segmented memory (if segmentation is enabled). Set segment number in SS5-16

Turning Power On

1. If automatic program load is desired, turn rotary switch to LOAD. If not, select STOP/STEP.

2. Turn CPU keyswitch ON. The blowers must begin operating. A MASTER CLEAR sequence occurs, and an automatic program load takes place (if the option is present and the device is ready).

3. On the paper tape unit (if present), press the POWER switch; the internal indicator should light and the fans should begin operating.

4. Turn the supervisor terminal LINE/OFF/LOCAL switch to LINE.

5. Refer to the appropriate vendor manuals for turn-on procedures of other peripheral devices.

Master Clear

The MASTER CLEAR switch places the Prime CPU in a known starting condition:

- Except for the P-register, all registers in the register files (memory locations 0-'37 for the Prime 100-200-300) are cleared (set to all 0s).

- The P-register is set to '001000 (the starting address of most self-loading programs, and the restart address of most system programs).

- The CPU is set up in 16K sectored addressing mode and single precision arithmetic mode, with interrupt and machine check inhibited and standard interrupt mode in effect.

- All peripheral controllers are initialized.

- Micro-verification takes place if the CPU contains the microverify feature.

MASTER CLEAR operates in any position of the mode control rotary switch except LOAD or RUN.

Setting Address Display

Before a series of register or memory FETCH or STORE operations, the memory address must be set to a starting value using the panel data entry switches. The rotary switch can be in any position except RUN or LOAD.

1. Set the ADDRESS/DATA switch to ADDRESS.

2. Clear the address display by pressing the DATA CLEAR switch (the indicators go out). Set the desired address by pressing the momentary-action data entry keyswitches. (If you make a mistake, clear the display by presssing DATA CLEAR and try again.)

Setting Data Display

Data to be stored in a memory location is first set into the DATA display from the panel data entry switches. The rotary switch can be in any position except RUN or LOAD.

1. Set the ADDRESS/DATA switch to DATA.

2. Clear the display by pressing the DATA CLEAR switch. Set the desired value by pressing the momentary-action data entry switches. (If you make a mistake, clear the display and try again.)

Examining Registers and Memory Locations (P100-200-300)

1. Set the address display to the first location to be examined. Return ADDRESS/DATA switch to DATA position. Turn to FETCH and press START. The contents of the register (or location) will appear in the indicators. (See Table C-3 for a map of registers and reserved memory locations.)

2. To examine successive locations, turn to FETCH Y+1. Every time START is pressed, the address register is incremented before the location is read. The indicators display the contents of successive locations after each depression of the START switch.

Table C-3
Reserved Memory Locations (Prime 100, 200, and 300)

Address	Assignment
'000000	X-Register - Index Register
01	A-Register - Arithmetic, Shift, I/O
02	B-Register - Ext. Arithmetic, Shift
03	Stack Pointer
04	FLPH - Floating Point High (Optional)
05	FLPL - Floating Point Low (Optional)
06	VSC - Visible Shift Counter
07	P-Register - Program Counter
10	PMAR - Page Map Address Register (Optional)
11	Microcode Scratch Location
12	EAS - Effective Address Save (ILL, UII, Interrupts, etc.)
13	Microcode Scratch Location
14	Y-Register Save for Control Panel DMA
15	M-Register Save for Control Panel DMA
16	Microcode Scratch Location
17	Microcode Scratch Location
20-37	DMA Range/Start Address Pairs
40-57	Reserved for DMC Channel Pairs
60	PFI - Power Failure Interrupt
61	RTCI - Real Time Clock Increment (Optional)
62	REVI - Restricted Execution Violation (Optional)
63	INT - Standard Interrupt (Compatible Mode)
64	Page Fault - Addressed Page Not in Mem. (Optional)
65	SVC - Supervisor Call Trap
66	UII - Unimplemented Instruction Interrupt
67	PE - Memory Data Parity Error
70	Machine Check - Processor Detected Error
71	Missing Module - No memory at Accessed Location
72	ILL - Illegal Instruction Interrupt
73	PWV - Page Write Violation (Optional)
74	FLEX - Floating Point Exception (Optional)
75	PSU - Procedure Stack Underflow (PRIME 300 only)
76-100	Debugging Scratch Area
101-177	Interrupt Vectors
200-777	General Cross Sector Links

Loading Registers or Memory Locations (P100-200-300)

1. Set the address display to the first location to be altered. Then, set the data display to the desired content and turn to STORE. Press START. The content of the data register will be loaded into the location. The indicators can display either data or the address, depending on the position of the DATA/ADDRESS switch.

2. To load sequential locations, turn to STORE Y+1. Every time START is pressed, the address display is incremented before the location is loaded. Therefore, by repeatedly loading a new value into the data register and pressing START, sequential locations are loaded.

Accessing Physical Memory (P350 and up)

When segmentation is not enabled, the control panel accesses real physical memory. Sense switches 11-16 provide the high-order bits of a full 22-bit physical address. Locations 0-'37 are the Prime 100-200-300 registers listed in Table C-3.

1. To do a STORE operation, set the 16-bit data value in the DATA display.

2. Set low-order 16 bits of address in ADDRESS display.

3. Set high-order 6 bits of address in SS11-16.

4. Set SS1 down and SS4 up.

5. Set rotary switch to desired function and press START for access.

Accessing Mapped Memory (P350 and up)

If segmentation is enabled, memory accesses are mapped. The segment number is set in SS5-16. Locations 0-'17 are specific Prime 350-400-500 register values as shown in the second column of Table C-4.

1. If this is to be a STORE operation, set the 16-bit data value in the DATA display.

2. Set address of word within segment in ADDRESS display.

3. Set segment number in SS5-16.

4. Set SS1 and SS4 down.

5. Set rotary switch to desired function and press START for access.

Note

If the addressed page is not currently in physical memory, the DATA indicator display is unpredictable.

Accessing Prime 350-400-500 Registers (Absolute Addressing)

With SS2 up, any location in any register set can be accessed by an absolute address:

'0 - '37	Microcode scratch
'40 - '77	DMA
'100 - '137	User Set 2
'140 - '177	User Set 3

The Y+1 functions increment the register address before the access. The address overflows from '177 to 0 as higher-order bits are ignored.

1. If this is to be a STORE operation, set the 16-bit data value in the DATA display.

2. Place SS2 up.

3. Set absolute address of desired register location in SS10-16.

4. Set SS4 to select the register half to be accessed: UP=high, DOWN=low.

5. Set rotary switch to desired function and press START for access.

Table C-4
Prime CPU Register Correspondence

P350-400-500 Register Location	P100-200-300 Register Location	Register Contents
2H	7	P (program counter)
2L	1	A (accumulator)
	2	B (double-precision and long accumulator extension)
3 H,L	-	EH,EL (accumulator extension for MPL,DVL)
5 H	3	S (stack), Y (alternate index)
7 H	0	X (index)
10 H	13	-
10,11	-	FAR0,FLR0 (field address and length register 0)
12,13	-	FAR1,FLR1 (field address and length register 1)
12 H	4	FAC (floating accumulator, mantissa high)
12 L	5	FAC (mantissa middle)
13 H	6	FAC (exponent)
13 L	-	FAC (mantissa low, double-pecision)
14 H,L	-	PB (procedure base)
15 H,L	14,15	SB (stack base)
16 H,L	16,17	LB (linkage base)
17 H,L	-	XB (temporary base)
20 H	10	(High half of DTAR3)
20 H,L	-	DTAR3 (descriptor table address segments 3072-4095)
21 H,L	-	DTAR2 (segments 2048-3071)
22 H,L	-	DTAR1 (segments 1024-2047)
23 H,L	-	DTAR0 (segments 0-1023)
24 H,L	-	Keys, modals (see Table 2.9)
25 H,L	-	OWNER (address of process control block of process owning register contents)
26 H	11	FCODE (fault code)
27 H,L	-	FADDR (fault address)
27 L	12	(Fault address word number)
30 H	-	Process 1024-microsecond CPU timer

Note

1. In mapped access for P350-400-500, the P100-200-300 Register Contents are accessed as shown in the second column.

2. H means high-order half of 32-bit register. L means low-order half.

Accessing Prime 350-400-500 Current Register Set (CRS)

With SS2 down, the current register set is accessed by addresses '0 through '37. The first column of Table C-4 shows the contents of the CRS. For Y+1 functions, the address is incremented before the address overflows from '37 to 0.

1. If this is to be a STORE operation, set the 16-bit data value in the DATA display.

2. Place SS2 down.

3. Set address of desired location in CRS in SS12-16.

4. Set SS4 to select the register half to be accessed: UP=high, DOWN=low.

Single Step Operation

1. Turn to STOP/STEP. The address display shows the address of the next instruction to be executed. (The data display is the instruction itself.) To begin single stepping at another location, STORE a new address in the P-register.

2. Press START. The current instruction is executed and the address of the next instruction is formed. (The next address is normally P+1, but may be P+2 after a skip or a two-word instruction, or the jump destination after a jump instruction.)

3. Registers or memory locations can be examined or altered using the FETCH and STORE functions. When the rotary switch is returned to STOP/STEP, the CPU is ready to execute at the current program count in the P-register, even though the address display may contain the address used for the last FETCH or STORE cycle. After the next operation of the START switch, however, the address display is updated to show the program count again.

4. Interrupt and DMX operations are disabled in STOP/STEP mode.

Starting a Program

Note

Also see AUTOMATIC PROGRAM LOAD for alternate ways to start execution.

1. Turn the rotary switch to any position except RUN or LOAD. If the machine needs to be initialized, press MASTER CLEAR.

2. Load the P-register with the desired starting address, obtained from the listing, the paper tape label, or other documentation. (A MASTER CLEAR leaves the P-register set to '1000, the starting location of most Prime system software.)

3. Determine whether other registers or memory locations must also be preset to starting values (see Table C-4).

4. Turn to RUN and press START. The program should begin running (STOP indicator goes out). Panel switches and indicators are then accessible to programmed I/O instructions only.

Recovering from Errors

If an equipment failure or program error causes the CPU to leave PRIMOS II control, it is usually possible to restart the CPU from the control panel. The procedure is:

1. Turn the rotary switch to STOP/STEP; press MASTER CLEAR.

2. Set the sense switches to the restart address.

3. Turn rotary switch to LOAD; press START.

4. Reset sense switches to 0.

Panel Lockout

Turn the keyswitch to the LOCK position. The CPU is then insensitive to the position of the rotary switch, and the running program cannot be stopped from the panel. (The sense switches and indicators are still accessible to programmed I/O instructions, however.)

Stopping Execution

1. To stop program execution, turn panel lockout keyswitch to ON and turn rotary switch to STOP/STEP. (STOP indicator should light.) Memory and registers can be examined or altered, and a new starting address can be set.

2. To continue execution, turn rotary switch to RUN and press START. Lock panel if desired.

AUTOMATIC PROGRAM LOAD

The Automatic Program Load (APL) feature of Prime CPUs simplifies loading of programs and operating systems from external storage devices. Operating procedures are described briefly below. See Chapter 4 for standard disk boot instructions, and Appendix A for standard tape boot instructions. Appendix A also explains the various sense switch settings used during boots.

1. Install on the appropriate device the medium from which a program is to be loaded. Make sure power is on and both the device and the supervisor terminal are ready to operate.

2. Set the sense switches as explained in Appendix A.

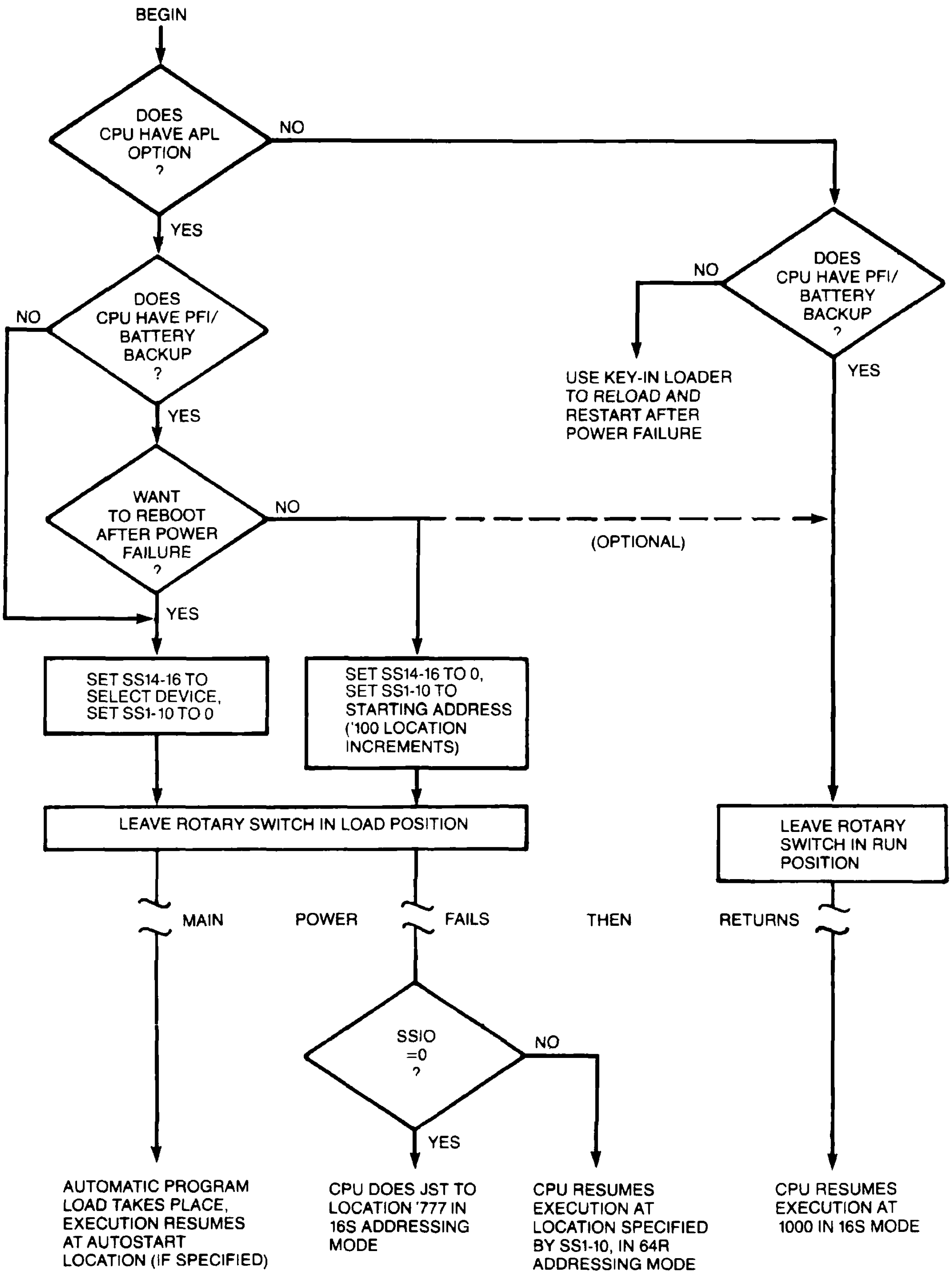
3. Press MASTER CLEAR.

4. Turn rotary switch to LOAD and press START. If bootstrapping is successful, a message will appear on the supervisor terminal.

AUTOMATIC RESTART OR REBOOT AFTER POWER FAILURE

Once a program has been loaded and started, the CPU is capable of an automatic restart (or reload) after a power failure. There are four possible methods, depending on whether the CPU contains the APL or battery backup/PFI features. Figure C-2 summarizes the panel setup required to prepare the CPU for a restart/reboot sequence.

When a reboot is desired, the autoload device must be ready to operate. If paper tape is used, it must be repositioned to the beginning of tape. (The tape must have an autostart address.)



Panel Setup for Automatic Restart or
Reload after Power Failure

Figure C-2

D
Physical Device Numbers

INTRODUCTION

Each physical disk or disk partition has a physical device number identifying the type of storage device, the drive unit on which it is mounted, and, for partitions, the size of the partition and its location on the disk pack. These physical device numbers are used in the commands: ADDISK, ASSIGN DISK, CONFIG, COPY, DISKS, FIX_DISK, FIXRAT, MAKE, SHUTDN, and UNASSIGN DISKS. Some of the devices listed in this section are no longer supplied; the information is included here for archival and reference purposes.

DRIVE UNIT NUMBERS

The drive unit number identifies the physical drive on which the disk is mounted. 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.

Diskette and cartridge disk drive unit numbers are set by a thumbwheel switch on the drive unit. Thumbwheel numbers must be different for all drives connected to the same system. Drive unit numbers for storage modules are set by the removable buttons. The system installer should have labeled these units; in many cases the drive unit number will be on one of the push button switches on the front of the drive unit.

DISKS

Fixed Head Disks

Physical device numbers or fixed head disks are:

Sectors/Track	Physical Device Number
8	10
64	40

Diskettes (Floppy Disk)

All diskettes have four sectors per track and a total of 304 records (448 words per record). The physical device number is 2u, where u is the number of the diskette drive minus one.

Diskette Drive Number	Physical Device Number
1	20
2	21
3	22
4	23
5	24
6	25
7	26
8	27

Moving Head Disks - Cartridges

Cartridges have 8 or 32 sectors per track, and a total of 448 words per record.

The physical device number is ts, where t is the type of disk and s defines the surface of the disk and the drive unit on which it is mounted.

Sectors/Track	Type (t)	Records	Comments
8	0	3248	6MB cartridge (early model controller)
8	3	6496	6MB (early model disk pack)
32	5	6496	6MB cartridge
32	5	12992	12MB cartridge

s = 2*(drive unit number) for upper (removable) surface
s = 2*(drive unit number)+1 for lower (fixed) surface
Drive unit numbers are 0, 1, 2, 3.

Following is an example of device numbers for a 12 MByte cartridge:

Drive Unit	Surface	Device Number
0	upper	50
0	lower	51
1	upper	52
1	lower	53
2	upper	54
2	lower	55
3	upper	56
3	lower	57

Moving Head Disks - 60 MByte (30 million words)

Moving head disks (60 MB) have 32 sectors per track and 64960 records (448 words per record).

The physical device number is 5c5u. c is 0 if the drive is connected to controller address '21 (default) and 2 if connected to controller address '23 (explicit). u is twice the value of the drive unit (0-3) on which the disk is mounted.

For example, a 60 MByte disk mounted on drive unit 2 connected to address '23 of the controller has the physical device number 5254.

Partitioning: A disk may be subdivided into partitions, each of which will be treated as if it were an actual physical device. Partitions must be an integral number of heads in size and must be offset an even number of heads from head 0 of the disk pack. Physical device numbers for partitions are given in the table below.

Partition Size		Physical Device Number	
Heads	Records	Controller Address='21	Controller Address='23
2 (default)	6496	xx005y	xx025y
2(explicit)	6496	xx045y	xx065y
4	12992	xx105y	xx125y
6	19488	xx145y	xx165y
8	25984	xx205y	xx225y
10	32480	xx245y	xx265y
12	38976	xx305y	xx325y
14	45472	xx345y	xx365y
16	51968	xx405y	xx425y
18	58464	xx445y	xx465y
20	64960	xx505y	xx525y

The head offset is the number of heads (disk surfaces) which lie between the start of the partition and the start of the disk pack (bottom).

xx is one-half the head offset of the partition (octal value)

y is twice the value of the drive unit on which the disk is mounted

Following are two examples of device numbers for partitioned disks.

- A 6-head partition with an offset of 4 heads is mounted on drive unit number 0 connected to controller address '21.

  xx=02

  y=0

  The physical device number is 021450.

- A 20-head partition (entire disk) is mounted on drive unit number 2 connected to controller address '23.

  xx=00 (a 20-head partition takes up the entire disk and cannot have a nonzero offset)

  y=4

  The physical device number is 005254, the same as constructed above for the nonpartitioned disk.

Moving Head Disks - Storage Modules

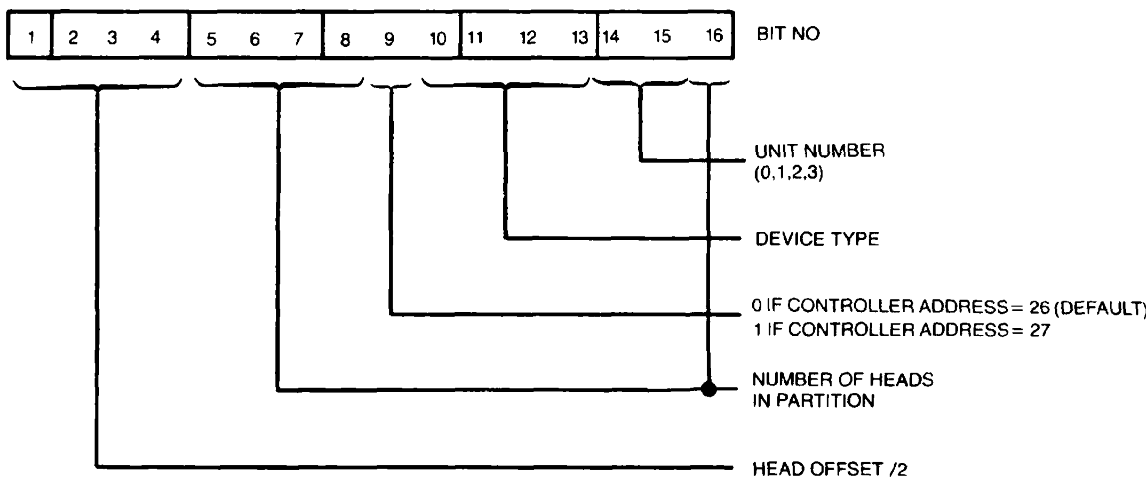
Storage modules exist in the three sizes indicated below. (They have 1040 words per record.)

Size of Module	Number of Heads	Type
40 MBytes	5	6
80 MBytes	5	6
300 MBytes	19	6

Storage modules are usually partitioned (subdivided), with each partition being treated as if it were an actual physical device. Partitions must be an integral number of heads in size and must be offset an even number of heads from the start of the disk pack. However, the last partition on the disk may contain an odd number of heads.

The physical device number is constructed as a 16-bit number, in octal (see Figure D-1).

A complete list of valid physical device numbers for storage modules is given in Table D-1.



Construction of Physical Disk or Partition Number
Figure D-1

Table D-1

Physical Device Numbers for Storage Modules
and Fixed Media Devices

Number of Heads in Partition	Starting Head Number: 2	4	6	8	10	12	14	16
1	———	———	02006z	———	———	———	———	———
2	00046y	01046y	02046y	03046y	04046y	05046y	06046y	07046y
3	———	01046z	———	———	———	———	———	———
4	00106y	01106y	02106y	03106y	04106y	05106y	06106y	07106y
5	00106z	———	———	———	———	———	———	07106z
6	00146y	01146y	02146y	03146y	04146y	05146y	06146y	07146y
7	———	———	———	———	———	———	06146z	———
8	00206y	01206y	02206y	03206y	04206y	05206y	06206y	07206y
9	———	———	———	———	———	05206z	———	———
10	00246y	01246y	02246y	03246y	04246y	05246y	06246y	07246y
11	———	———	———	———	———	———	———	———
12	00306y	01306y	02306y	03306y	04306y	05306y	06306y	07306y
13	———	———	———	———	———	———	———	———
14	00346y	01346y	02346y	03346y	04346y	05346y	06346y	07346y
15	———	———	———	———	———	———	———	———
16	00406y	01406y	02406y	03406y	04406y	05406y	06406y	07406x
17	———	———	———	———	———	———	———	———
18	00446y	01446y	02446y	03446y	04446y	05446y	06446x	———
19	———	———	———	———	———	———	———	———
20	00506y	01506y	02506y	03506y	04506y	05506x	———	———
21	———	———	———	———	———	———	———	———
22	00546y	01546y	02546y	03546y	04546x	———	———	———
23	———	———	———	———	———	———	———	———
24	00606y	01606y	02606y	03606x	———	———	———	———
25	———	———	———	———	———	———	———	———
26	00646y	01646y	02646x	———	———	———	———	———
27	———	———	———	———	———	———	———	07646z
28	00706y	01746x	———	———	———	———	———	———
29	———	———	———	———	———	———	06706z	———
30	00746x	———	———	———	———	———	———	———

All partitions marked with _x_ must be combined with an 11-head partition starting at head 30 for full utilization of the disk capacity. Combining these partitions with a partition of less than 11 heads will reduce the storage capacity by approximately 16K bytes per unused head.

_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 D-1 (Continued)

Physical Device Numbers for Storage Modules
and Fixed Media Devices

Starting Head Number

18	20	22	24	26	28	30	32	Number of Heads in Partition
—	11006z	—	—	—	—	—	—	1
10046y	11046y	12046y	13046y	14046y	15046y	06046x	—	2
10046z	—	—	—	—	—	—	—	3
10046z	11106y	12106y	13106y	14106y	15106x	—	—	4
—	—	—	—	—	—	—	—	5
10146y	11146y	12146y	13146y	14146x	—	—	—	6
—	—	—	—	—	—	—	—	7
10206y	11206y	12206y	13206x	—	—	—	—	8
—	—	—	—	—	—	—	—	9
10246y	11246y	12246x	—	—	—	—	—	10
—	—	—	—	—	—	—	17246z	11
10306y	11306x	—	—	—	—	—	—	12
—	—	—	—	—	—	16306z	—	13
10346x	—	—	—	—	—	—	—	14
—	—	—	—	—	15346z	—	—	15
—	—	—	—	—	—	—	—	16
—	—	—	—	14406z	—	—	—	17
—	—	—	—	—	—	—	—	18
—	—	—	13446z	—	—	—	—	19
—	—	—	—	—	—	—	—	20
—	—	12506z	—	—	—	—	—	21
—	—	—	—	—	—	—	—	22
—	11546z	—	—	—	—	—	—	23
—	—	—	—	—	—	—	—	24
10606z	—	—	—	—	—	—	—	25
—	—	—	—	—	—	—	—	26
—	—	—	—	—	—	—	—	27
—	—	—	—	—	—	—	—	28
—	—	—	—	—	—	—	—	29
—	—	—	—	—	—	—	—	30

Table D-1 shows all the valid physical device numbers for the 40, 80, 300, and 600 MB disks. To use Table D-1:

1. Decide upon the number of surfaces in the partition.

2. Decide upon the head number of the first head in the partition.

3. Look up the physical device number in the table.

Notes

If the partition defined is not in Table D-1, then it is not a legal partition.

All partitions must begin on an even head number.

To make the most efficient use of the storage device, only the last partition on the disk pack should have an odd number of surfaces.

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 D-2). 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:

Drive 0	Drive 1	Drive 2
000460	003462	000464
010460	071063	010465
021460		
050460		
060460		
070460		
100461		

This example is illustrated in Figure D-2.

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.

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.

CMD Type	Platter(s)	First Controller	Second Controller
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.

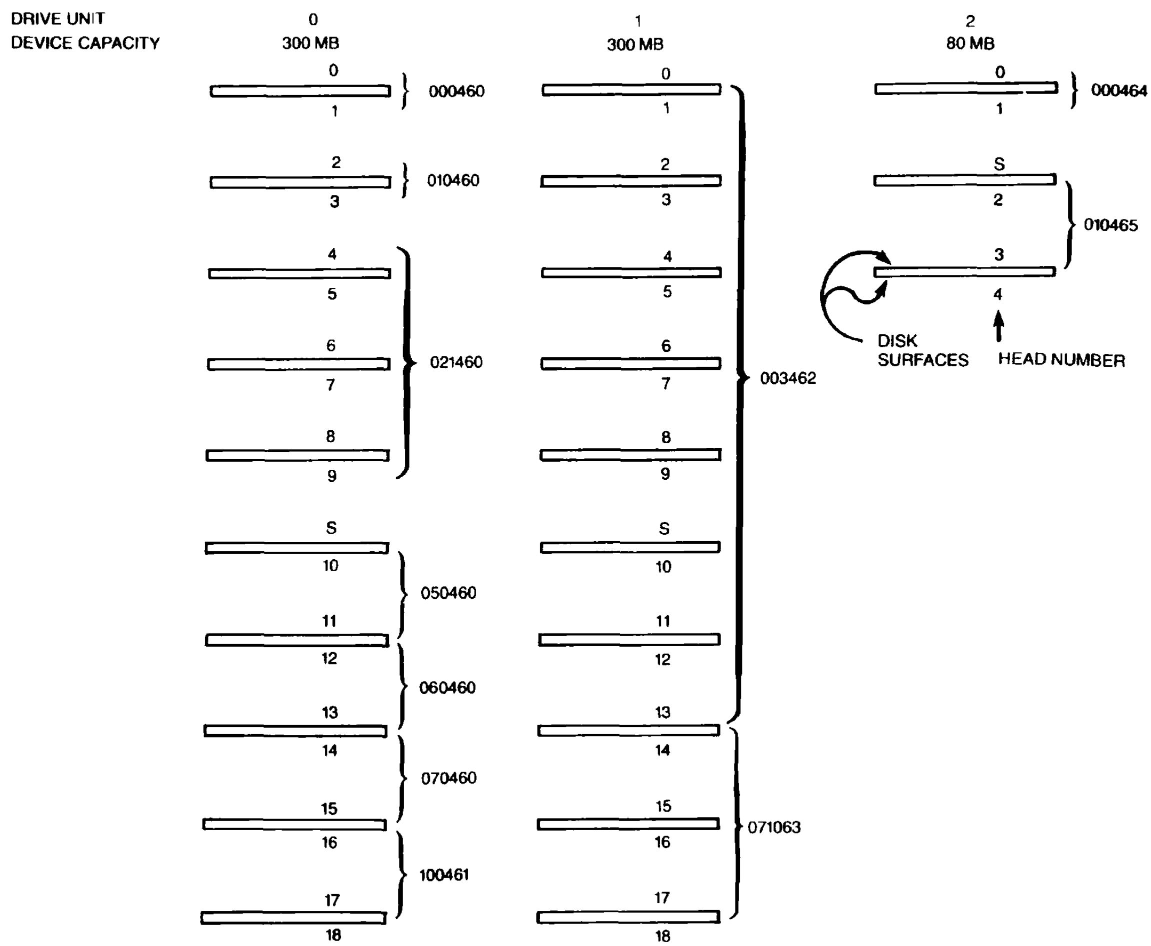
The nonremovable surfaces of the 64MB CMD can be organized as 1 or 2 partitions.

The nonremovable surfaces of the 96MB CMD can be organized as 1, 2, or 3 partitions.

Fixed Media Devices (Winchester Drive)

Fixed media devices exist in the four sizes indicated below:

Size of Device	Number of Heads
34	3
68	3
160	10
600	40



Example of Storage Module Partitions
Figure D-2

E

System Halts on Non-VCP Machines

RECOGNIZING HALTS

Under certain unusual circumstances (hardware or software malfunctions), PRIMOS will execute the HLT instruction and halt. A system halt is indicated by a red light directly above the rotary switch on the control panel.

It is also possible for the machine to "hang." In this case, the machine appears to have halted, but the red light is not on. Hung machines may be identified by the fact that no input or output is taking place at any peripheral equipment (including the supervisor terminal).

When a machine hangs, turn the rotary switch to STOP/STEP to stop the machine. Then treat it like a halted machine.

GENERAL GUIDELINES FOR HANDLING HALTS

When a halt occurs, the operator has two duties:

- Get the system running again as quickly as possible.
- Record enough information so that an analyst can determine why the halt occurred, which will help prevent future halts.

In order to provide this information, the operator must:

- Make sure that a current set of load maps is stored in a safe place (preferably with the system logbook).
- Spool out LOGREC at regular intervals, and keep the spooled copies on file.
- Take tape dumps when halts occur, if requested to do so by the analyst. (Table E-1 identifies those halts for which tape dumps are most likely to be useful.)

Load maps are named RING0.MAP and RING3.MAP. (Earlier systems had one map, named M_PRMO.) Load maps are stored in UFD PRIRUN. Since the load maps change whenever additions or updates of software are made to PRIMOS, the operator should be sure to spool new copies of the maps whenever the system is updated.

HANDLING HALTS UNDER PRIMOS

Halts that occur when the system is running under PRIMOS are handled in four steps, as follows:

1. Determine the segment number and address at which the halt occurred.
2. Look up the address on your spooled copy of the PRIMOS load map to determine the name of the halt location.
3. Check Table E-1 to find out which recovery procedure is appropriate for halts occurring at this location.
4. Take the appropriate steps for recovery, as detailed in Table E-2.

These steps are described in more detail below.

> **Caution**
>
> Never do a SYSCLR or a MASTER CLEAR until you have determined the halt location. Once either of these instructions has been given, the halt location (and hence the cause of the halt) is unobtainable.

Step 1: Determine the Segment Number and Address

Machines without Virtual Control Panels distinguish between hardware-related faults (called checks) and software-related faults.

Checks: When PRIMOS halts due to a check, an octal address is displayed in the control panel lights. Checks currently defined are:

Lights	Check Header Address	Location Name
206	200	PWRFL_
277	270	MEMPA_
306	300	MCHK_
316	310	MMOD_

Write the address and location name of the halt, together with the date and time of the halt, into your system logbook. (See the System Architecture Reference Guide for a complete discussion of checks.)

Software-related Halts: For software-related halts, take the following steps:

First find the halt address:

1. Turn the rotary switch to STOP/STEP.

2. Set the ADDRESS/DATA switch to ADDRESS. Enter the address displayed in the lights in the logbook.

Then find the segment number of the halt:

1. Turn the rotary switch to FETCH Y.

2. Set the ADDRESS/DATA switch to ADDRESS.

3. Push the DATA CLEAR switch.

4. Enter '14 in the lights.

5. Set sense switches 1 and 4 to the up position.

6. Set the ADDRESS/DATA switch to DATA.

7. Push START. Enter the segment number displayed in the lights into the logbook.

Step 2: Determine the Location

If the halt was a software halt, look up the address of the halt in a spooled copy of your current PRIMOS load map. Either the address you have or the address immediately preceding it in the listing should be a recognized halt location. (The names of recognized halt locations end in underscores: for example, PWRFL_.) Write the name of the halt location in the system logbook, next to the halt address.

An example of such a procedure is:

1. Halt message appears at terminal.

   HALTED AT 000004/000306: 003776

   CP>

2. Find the segment number (0004) on the load map. Your load map will be similar to this:

WARMCN	0004	000260	OTHER	
PFFLAG	0004	000262	OTHER	
PFAILS	0004	000263	OTHER	
MEMPA_	0004	000276	OTHER	
MCHK_	0004	000305	OTHER	
MMOD_	0004	000315	OTHER	
MEMH2_	0004	000317	OTHER	
CHKEND	0004	000321	OTHER	
MT2INT	0004	000321	OTHER	
MTINT	0004	000323	OTHER	
MGINT	0004	000325	OTHER	
MG2INT	0004	000327	OTHER	
INTEN2	0004	000476	OTHER	
CLKSEM	0004	000500	OTHER	
SEMCOM	0004	000500		COMMON
SLCSEM	0004	000502	OTHER	
AMLSEM	0004	000504	OTHER	
MPCSEM	0004	000506	OTHER	
MP2SEM	0004	000510	OTHER	

3. Find the word number on the map. Since 00305 is the largest number <= 306, it is the correct number for this halt.

4. The name of the halt location is MCHK_, a machine check.

How to Proceed Without a Map: If you cannot find a load map, and hence cannot look up the halt location, do a tape dump (action code D), and then try a warm start (action code W). If the warm start fails, try a cold start (action code C). As soon as the system is started, look for RINGO.MAP in the UFD PRIRUN. Look up the address of the halt (or the address preceding it) on this map and record it in the logbook. Then spool a copy of the map immediately and put it in the front of the logbook (or some other safe place) for the next time.

Step 3: Find the Appropriate Recovery Procedure

Refer to Table E-1 for the action code that matches each recognized halt location.

If your halt occurred at an address that is not a recognized halt location, use the action codes for "All others," shown at the bottom of Table E-1.

Table E-1
Action Codes for Halts

Halt Location	Reason for Halt	Action Code(s)
AMLC1_	Bad AMLC interrupt	D, W
BDMEM_	Bad memory during cold start	M
BOOT0_	Halt after SHUTDN ALL	D, C
MCHK_	Machine check	R, D, W
MEMH2_	Halt after automatic mapout of bad page	R,W
MEMPA_	Uncorrected memory parity error	X, M
MMOD_	Missing memory module	D, C
All others		D, C

Step 4: Take the Appropriate Steps for Recovery

Table E-2 details the procedures to follow for each action code. Follow the instructions given in this table, and record the results in your system logbook.

Table E-2
Recovery Procedures

Code	Action
C	<u>Cold Start</u> Bring the system up as if it had been shut down normally (as described in Chapter 4).
D	<u>Crash Magnetic Tape Dump</u> 1. Mount a scratch tape on drive unit 0 or 1. 2. Push the MASTER CLEAR switch. Restart at address '775 if tape is on unit 0, or at address '776 if tape is on unit 1 (see Appendix C). If switch 15 or 16 is up and the restart address is '775, then the tape unit indicated by the octal setting of the switches (0-3) is used . 3. The tape will write and then rewind. Label the tape with the date and time of the halt (and machine identification, if needed). Record the dump in the logbook, and give the tape to the person responsible for examining unusual halts.
M	<u>Map Out Bad Page</u> 1. Turn rotary switch to RUN. 2. Push START button. When this action is taken following a memory parity error, PRIMOS maps out the bad page; that is, it records the location as being bad and doesn't use that page of memory in the future. PRIMOS then continues the cold start.

Table E-2 (continued)
Recovery Procedures

Code	Action
R	<u>Record Register Settings</u> Record the contents of registers '27, '34, '35, and '36 in the logbook. (See Appendix C.) For each register: 1. Turn the rotary switch to FETCH Y. 2. Push the DATA CLEAR switch. 3. Enter the register ('27, '34, '35, or '36) in the lights. 4. Place sense switches 1, 2, and 4 in up position. 5. Set ADDRESS/DATA switch to DATA. 6. Push START. The number in the lights is the high half of the 32-bit register. To display the low half, put sense switch 4 down. The contents of the registers should be recorded in the logbook. <u>Note</u> Register '27 is for the Prime 750 and 850 only.
W	<u>Warm Start</u> 1. Turn the rotary switch to STOP/STEP. 2. Press the MASTER CLEAR switch. 3. Turn the rotary switch to RUN. 4. Press the START switch twice. The system should restart and type WARM START at all terminals except the supervisor terminal. <u>Note</u> A warm start restarts all input/output controllers before restarting PRIMOS. Thus, a warm start may run for 20 seconds or more before the WARM START message appears at the user terminals. It will be slightly longer before a message appears at the supervisor terminal. Do not assume a warm start has failed without waiting 20 seconds and checking the user terminals for the WARM START message.

Table E-2 (continued)
Recovery Procedures

Code	Action
X	<u>Uncorrected Memory Parity Error</u> 1. Determine where error occurred and record results in logbook. Access locations 0, 1, and 2 (with no sense switches set) to obtain the contents of the X-, A-, and B-registers. These are 16-bit registers containing: <u>Location</u> <u>Register</u> <u>Contents</u> 0 X user getting parity error 1 A page number 2 B offset within page 2. If user number (from register X) is 1 (supervisor), then a cold start must be done. 3. If user number is not 1, have PRIMOS map out the bad page by turning the rotary switch to RUN and pushing the START button (as shown under action code M). This prevents that page of memory from being used again. 4. When it has mapped out the bad page, the system will halt at MEMH2_. It must then be warm started (action code W). Turn the rotary switch to LOAD.

HANDLING HALTS UNDER PRIMOS II

PRIMOS II may be halted by the operator to abort a long listing or to recover from a bad startup. Such a halt could also be caused by a hardware condition or equipment failure. If the fault is in the hardware, the operator should not attempt to restart the system until the hardware problem has been corrected.

RESTARTING AFTER A HALT

To restart from a halt:

1. Turn the rotary switch to STOP/STEP.
2. Press the MASTER CLEAR switch.
3. Set '170000 in the sense switches (sense switches 1, 2, 3, and 4 in up position).
4. Turn the rotary switch to LOAD.
5. Press the START switch.
6. Reset the sense switches to 0.

PRIMOS II will restart and respond with:

```
OK:
```

F
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 MESSAGES

- BAD BADSPT FILE ON PARTITION pdev - IGNORED

The badspot file contains an error, and 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 occured, 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.

- NEED 32K MIN FOR BIG DISK

Not enough memory to copy large disk. COPY_DISK will abort.

- 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 AVAILABLE ON PARTITION pdev
  OK TO WRITE TO IT WITHOUT BADSPOT HANDLING (YES/NO)?

No free records are available on the target disk for mapping around the badspot. 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.

- UNRECOVERED ERROR

10 successive unsuccessful write operations have occurred.

- 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
- YOU WILL NEED n SURFACES TO COPY FROM

(CMD information only.)

- VERIFY ERROR record-number word-number

A discrepancy at the indicated location has been detected during the verification procedure. Verification continues.

G

FIX_DISK Error Messages

INTRODUCTION

This appendix contains error messages generated by the FIX_DISK command. For a discussion of FIX_DISK, see Chapter 8.

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.

- 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.

- 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 physical device number

The physical device number that is specified in the command line is bad.

- The BADSPT file is bad, ignored

The BADSPT file that is found by FIX_DISK is bad. Badspots will not be handled.

- The Beginning Record Address (BRA) pointer is bad. It should be YY instead of XX

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.

- The current record address (CRA) is bad. It should be YY is XX

The current record address word of this record does not match the current record. If the -FIX option is specified, the CRA is corrected.

- 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 read/write error. Record = XX TRACK = YY HEAD = ZZ

An error occurred while reading/writing record XX. If the -FIX option is specified, the file is truncated on read errors and this badspot record is added to the BADSPT 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 -INT 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.

- The father pointer is bad. It should be YY is XX

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 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, the file is deleted.

- 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 record of a DAM file is not zero

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 points to a record that belongs to another file

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.

- 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 to represent the data records

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; otherwise the file is truncated.

- 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 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.

- 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 {-DISK} is missing

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 DOS. Since DOS doesn't support quotas, there may be directories on this partition with incorrect quota information.

- 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 wrong length. It should be YY is XX

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 -INT and -FIX options are omitted, FIX_DISK aborts. 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:

    80 or 300 MB storage module?

FIX_DISK then asks:

    Split partition?

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:

    Partition XX File-records XX Paging-records XX

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 appropriately.

- The second 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.

- System file is bad, ignored

An error which 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.

- The third file entry of the MFD file is not BOOT

FIX_DISK checks that the third entry in the MFD is BOOT. If this entry is missing, FIX_DISK will not abort.

- 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.

- 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 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 -AT option is specified, in which case directories that are nested too deeply 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 option are omitted, the file entry is left untouched, and no compression is performed for the UFD in which this file entry resides.

- 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.

H
BATCH Messages

INTRODUCTION

Messages in this appendix include 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 database 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 which is being reported was the result of an earlier, unreported error.

BATCH MESSAGES

Following is a list of Batch messages. The nature of each message (e.g., 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 with legal -FUNIT and -PRIORITY values. Note that the system may be configured to have fewer than 126 units per user at cold start, and the -FUNIT argument will 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 (end of line). This error message is a warning, but it may indicate differing degrees of fatality 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 in entering the BATGEN command, the user is returned to PRIMOS, although the "ER!" condition is not raised.

- Bad $$ command.

(Fatal) A command file was submitted using the JOB command that 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 database files is inaccessible or has a bad format. The Batch subsystem is therefore inoperative until it is fixed.

- Bad queue definition file.

(Fatal) A file referenced by BATGEN did not comply to format requirements, i.e. was not a legal queue definition file. If this error occurs in 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 database, 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 spawned from the supervisor terminal, and therefore 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 respawn the Batch monitor 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 (i.e. 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. 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, then 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 database, or a Disk Write-Protected error), FIXBAT should be run. If that fails, the "BATCHQ>INIT" using the "-RSTQ" option should be resumed in BATCHQ to reinitialize the entire database. If invoking C_BDIF doesn't work, there are probably disk errors. If it does, 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 have been set. These parts of the Batch system cannot be run until the system data and time are set using the SETIME command from the supervisor terminal.

- Elapsed time limit must be specified.

(Fatal) See the explanation for the "Cpu limit must be specified" message. References to CPU limits and the -CPTIME option should be read as references to elapsed time limits and the -ETIME option.

- End of line.

(Fatal) One of the Batch programs was expecting to find more information on the command line, but found end-of-line instead. 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 <u>option</u> 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 which required at least one numeric parameter, but there was none. 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 character string "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 <u>no</u> 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. <u>filename</u> (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 home 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 cannot fit, use a shorter description of it when you resubmit the command file, after editing the file to include an ATTACH command with a relative pathname to 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), so 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. The job did successfully submit, however.

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 abort, 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) Output when the answer to a question is not "YES", "NO", or "OK". Will ask the question 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, i.e. they were less than or equal to zero, or were not legal decimal numbers and not the string "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 character string "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, i.e., 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 (first character must not be a digit, character set limited to alphabetics, digits, and selected special characters). Reenter the command with a legal queue name. Note that a queue name of "ALL" is illegal, so that "DELETE ALL" will not produce undesirable results.

- 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 monitor cannot get 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 being given from the supervisor terminal. After determining that the BATCHQ UFD exists, respawn 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." (note that the period is included in the treename) for perusal by the System Administrator. 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 database. It will flag the database as invalid. The System Administrator should be notified, as he has the responsibility for reinitializing the database (or running FIXBAT or FIXRAT as the case may be). The BATCH and JOB commands will be inoperative until the situation is resolved.

- ?Job <extnam><intnam> <status>.

(Warning) An attempt was made to perform an operation on a job using the JOB command that could not be performed because of its status. Examples are trying to restart a completed job or attempting to release a job that is not held.

- Job <extnam> for <username><intnam> <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"). extnam is the external name of the job, username is the submitting user, intnam 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 (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 (i.e. a -RESTART will abort the job but 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 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 that have that name are not active jobs (i.e. have completed, aborted, or been canceled); or that a job exists with that external name but the user making the request is not the same user that 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 queues initialized.

(Response) This is output by the INIT program. It means that the queue and execution data have been zeroed.

- (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 is done on it. 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 in this case.

- Job will be restarted.

(Message) This is sent to the supervisor terminal after a "Job <extnam> for <username>(<intnam>) aborted/completed" message is sent, only 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. It generally indicates that the job will be recoverable from 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 database has not been initialized, or that something has happened to it (e.g., 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 SYSTEM -STOP", and will verify the validity of the database when it is started up. Either way, the user will be immediately returned to command mode, i.e., the operation the user attempted will <u>not</u> 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 already started. This message is just a reminder, not a fatal error.

- Monitor in operation

(Message) The Batch monitor has finished fixing the database (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) A reference was made to a job using an external name in the JOB command, and there were at least two such jobs that were active belonging to the user making the reference. 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 in the database.

- Multiple occurrence.

(Fatal) An option was specified twice during job submission or job changing on either the JOB or $$ JOB line (example: JOB C_TEST -HOME HERE -HOME THERE). If an option is specified once on the JOB line and once on the $$ JOB line, no error will result and 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 options -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. It will retry every five minutes, sending this message at each unsuccessful try. (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 run FIXBAT with the -DAYS option to remove old files from the database. (See Chapter 11 for details on FIXBAT.)

- My quota is exceeded. Please help me.

(Message) The Batch monitor has encountered a "Quota Full" condition. It will retry every five minutes, sending this message at each unsuccessful try. (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 run FIXBAT with the -DAYS option to remove old files from the database. (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 jobs belonging to that user that are waiting, held, or executing. If user is SYSTEM, then there are no jobs that are waiting, held, or executing in the entire system.

The text in brackets is output 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 after the -CHANGE option.

- No jobs named jobname in system.

(Response) The operator has requested information on a job (specifying 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 usernumber of the process, it had disappeared. If the message "(Job restarted)" had been typed out, then the job would be restarted. Although the operation itself was unsuccessful, the desired results were achieved.

- No queue available for job.

(Fatal) A job was submitted using the JOB command that did not specify which queue it was to be submitted to (no -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; etc.) Use of the BATGEN -STATUS or -DISPLAY command may yield a list of legal queues and their status if the file BATCHQ>BATDEF is read-permitted.

- No queues have waiting or held jobs.

(Response) A BATCH -DISPLAY command was issued, and there were no queues that had any waiting or held jobs in them. A queue may have one executing job in it, but an executing job is not considered a waiting or held job.

- 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) A BATCH -DISPLAY command was issued, and there were no jobs that were currently running. It is possible for there to be no running jobs and to have jobs waiting, however, even when the monitor is running and there are free phantoms; there is always a small amount of turnaround time between the submittal of a job and its execution. This time is about 20 seconds maximum.

- Not an absolute treename.

(Fatal) The home UFD specified with the -HOME option during submission using the JOB command (or changing of job parameters) was a relative treename, i.e. it began with "*>". Resubmit the job, giving an absolute pathname after the -HOME option.

- Not your job.

(Fatal) A reference was made to a job using an internal name in the JOB command, and the referenced job did not belong to the user making the reference. Use "JOB -STATUS" to obtain a list of all active jobs belonging to the user making the request.

- 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 which 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, while normally limited to 126, may have a smaller upper limit, depending on the coldstart configuration of the number of 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 (e.g., "File in use, please stand by") will be output 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 will result in a fatal error, as it could indicate that system security is broken.

- Please wait.

(Response) This message asks that the user be patient because the program which is running has been locking up the Batch database too long and is now allowing other processes to have access to it. It is not a fatal error. It generally is output 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) An attempt was made to add a queue which already existed while in BATGEN command mode. The status referred to is either "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 returns 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, because there are no longer any waiting, held, or executing jobs in that queue.

- Queue <name> flagged for deletion.

(Warning) An attempt was made to delete a queue which had already been deleted but was still flagged for deletion while in BATGEN command mode. 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 return also causes the QUIT to be taken (i.e., "YES").

- Queue deleted.

(Fatal) The queue that the job was being submitted to was present when it was first checked out, 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 should provide a list of currently available queues and their status, if the file that defines queues is accessible to users (i.e., 'R' access rights to BATCHQ>BATDEF for regular users of the BATCH subsystem).

- Queue full.

(Fatal) There are already 10,000 jobs (whether active or inactive) in the queue that the job is being submitted to by the JOB command. The queue must be deleted and recreated before more jobs can be submitted to it. The System Administrator should be asked to do this. Meanwhile, if any other queues are available, they can be used instead by the user.

- 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 a queue in the BATDEF file that is flagged for deletion but has never had a job submitted to it. It indicates that it has deleted the queue from BATDEF, but that no job data was lost as a result.

- Searching for free command file, please stand by.

(Response) This and other messages like "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 database. (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 database.

- 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 and the job successfully run, then try a queue with higher limits.

- Stop request issued.

(Response) The BATCH -STOP command has resulted in the Batch monitor being requested to 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 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 has had a JOB -CANCEL done to it while it was executing, or was submitted with the -RESTART NO option. Any restarts done to the job will abort the job (if they succeed), but the job will not be restarted.

- (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 being done on that job, or a system cold start after being brought down while the job was executing.

- This job will be restarted.

(Response) This message is output by a JOB -DISPLAY command if the job being displayed has had a JOB -RESTART done to it but it has not yet aborted or completed and is still executing. When the monitor sees that the job has aborted/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 options were entered that conflicted with each other, 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 to add a queue when there were already 16 queues (blocked, unblocked, <u>or</u> flagged for deletion) defined, using the ADD command in BATGEN.

- Unknown command.

(Warning) A command was entered while in BATGEN command mode that was unrecognized. The user will be left in BATGEN command mode and the erroneous line will be thrown away.

- Unknown option.

(Fatal) An option was entered to the BATCH or JOB command that was not recognized.

- Unknown queue name.

(Warning) A command that was 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, a subcommand was given which was not recognized. The user will be left in subcommand mode.

- Unrecognized option.

(Fatal) BATGEN was invoked with an option on the command line that was not recognized. 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. The operation that the user requested will still be performed. Note that if the monitor is force-logged out, or the system is shut down without the monitor logging itself out, there may be a database problem as a result.

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)

The operator or System Administrator has either tried to run FIXBAT with the -STARTUP option interactively from the supervisor terminal or tried to run FIXBAT as a phantom logged in as SYSTEM; for FIXBAT to run with the -STARTUP option, it must be run as a phantom, and the phantom must be started 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) that held entries that were all past the -DAYS argument, and that FIXBAT deleted the file.

- Execute data not found (reinitialize). jobid

A job that had execution status in the queue file had no corresponding entry in EXECUT. This can occur if the Batch monitor is logged out focibly in between updating EXECUT and updating the queue file. Resume "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. Resume "INIT" to reinitialize.

- FIXBAT finished.

The process of fixing the Batch database has been successfully completed. FIXBAT will now exit to PRIMOS.

- Fixing database.

This message is output when FIXBAT decides to actually fix the entire BATCHQ database.

- <filename> leftover words=n

The indicated queue file had _n_ words at the end of it, which was not enough for a full queue entry. This is not a fatal error, and the queue file will merely be truncated. It could indicate that a process submitting a job was force-logged out in the middle of creating the 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 that had more than one entry in the EXECUT file, which is impossible. Resume "INIT" to reinitialize the database.

- Unknown -STARTUP argument.

The argument supplied to the -STARTUP option is not SAVE, DELETE, SPOOL, or NOLOG.

I
Disk Errors

INTRODUCTION

All record, device, and status numbers are octal.

STORAGE MODULES

Following is a discussion of error detection and correction on storage modules.

Storage Module Error Detection

Under PRIMOS, disk read/write errors on storage modules 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 module or partition on which the error occurred

rec-num — The desired record number within the partition (2 words)

act-rec — The actual record number read (words). On reads it should agree with the rec-num. On writes it has no validity.

status — The reason for the error. It may be one of the words in the table below, or the sum of two or more words.

Status Word	Meaning
177777	Bad record identifier
177776	Device not ready
100000	Always set
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 storage module controller writes a two-word correcting code checksum on each record. This code is generated by the following:

```
G(X) = X** (1+X**2+1)*(X**21+1)
```

An error detection and correction scheme (ECC) is implemented for the storage module. The code, together with the correction logic in the storage module 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; and (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 storage module error correction is attempted and fails is:

UNCORRECTABLE.

The message printed when error correction succeeds is:

wordno error error

wordno — The offset relative to the beginning of the record at the beginning of the correction.

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 storage module error message:

```
DISK  RD ER  020063 000000 016357 000000 015477 100014 000012
UNCORRECTABLE.
DISK  RD ER  020063 000000 016360 000000 016355 100014 000012
UNCORRECTABLE.
```

OTHER DISKS

Disk error messages are printed in the same format as error messages for storage modules. For 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 (2 words) is the actual record number read and should match the requested record record-address (2 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

The meaning of the status word, typed as the right-most octal number of a disk error message, depends on the type of controller.

The meaning of the status word with the 4000 Controller early model is as follows:

Status Word	Meaning
177777	Bad record identifier
177776	Device not ready
100000	Data transfer complete (good if present)
040000	Read/write past end of record
040000	Seek complete (good if present)
002000	Write protect violation
000400	Command error
000200	Checksum error
000100	DMX overrun
000040	Stack overflow

The meaning of the status word with the 4001 Controller (cartridge disk) is as follows:

Status Word	Meaning
177777	Bad record identifier
177776	Device not ready
100000	Bit 1 always set
040000	DMX overrun
020000	Disk is write protected
010000	Checksum error
000100	Disk drive seeking
000040	Disk drive seeking
000020	Disk drive seeking
000010	Disk drive seeking
000004	Illegal seek
000002	Malfunction detected

The meaning of the status word with the Diskette Controller is as follows:

Status Word	Meaning
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

J
LOGLST Messages

INTRODUCTION

This appendix contains all messages that may appear in the output system event logging file (LOGLST). For a discussion of the LOGLST file and the LOGPRT command, see Chapter 5.

LOGLST 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 is given for read operations only.

STATUS(ACCUM) is the 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 — xxxxx 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, LOGPRT positions 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 — xxxxx 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 DSWSTATL), yyyyy is (INV); otherwise yyyyy is absent.

DSWPARITY is not present on all CPU models. At Rev. 19, DSWPARITY is displayed only for the P750 and the P850. If DSWPARITY is not present, xxx is an encoding of the machine check code and not RCM Parity in DSWSTAT as follows:

xxx	Meaning
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

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 RCMPE is printed, it means that the signal RCMPE- 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 - FIPBAPE+  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 - FIPBIPE+  D  Prefetch Buffer instruction

08 - RCMPE-       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 - GCBDPE-      C   C board detected error
02 - FBMDEVPE+    C   BMD input even word
03 - FBMDODPE+    C   BMD input odd word
04 - LMMOD+       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 - LRCDODPE+    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
```

- 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 (wwww) DSWSTAT= xxxxxx xxxxxx DSWRMA= xxxxxx xxxxxx
  DSWPB= xxxxxx xxxxxx
  DSWPARITY= xxxxxx xxxxxx PNN,WN= xxxxxx xxxxxx BIT= y OP=z

A memory parity error occurred. wwww 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.)

xx	Meaning
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= dddddd ...

A system logging file entry of type 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

Indicates a warm start of PRIMOS was performed.

K
NETLST Messages

INTRODUCTION

This appendix contains all messages that may appear in the output network event logging file (NETLST). For a discussion of the NETLST file and the LOGPRT command, see Chapter 5.

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.

- INCOMING CALL REQUEST TO PORT xxxx
  FROM ADDRESS yyyy

An incoming call for FAM debugging from address yyyy was sent to port xxxx.

- 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]

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.

- OUTGOING CALL REQUEST FROM PORT xxxx
  TO ADDRESS yyyy

An outgoing call for FAM debugging to address yyyy was received from port xxxx.

- 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 Prime 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 XMITS. 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.

L
Obsolete Commands

This appendix documents commands and utilities that have been replaced at Rev. 19.

▶ FAM I

Note

> FAM I is Prime's older File Access Manager. At Rev. 19, it has been replaced by FAM II.

FAM I must be started/restarted by a phantom. 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 FAM II format. An additional command, REMOTE, is required to set access to local disks.

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 already have been closed down.)

Setting Access to Local Disks (REMOTE): The REMOTE command is used to permit or deny access to local disks by specific network nodes or by the entire network. The format of the command is:

REMOTE {PERMIT | DENY} [option]

where PERMIT and DENY permit or deny access to local disks by remote network nodes according to the options. The options are:

```
nodename pdev-1 [pdev-2 ... pdev-9]
nodename -ALL
-NET     pdev-1 [pdev-2 ... pdev-9]
-NET     -ALL
```

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]

Permits the operator at system nodename to 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 startedup 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 executed.

REMOTE PERMIT -NET pdev-1 [pdev-2 ... pdev-8]

Permits all configured network nodes to access the specified local disk partitions.

REMOTE PERMIT -NET -ALL

Permits all network nodes to 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 starting up the disk the next time it tries to do so.

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
        SYSC
        SYSE
        SYSF
        SYSG
        SYSH
        SYSJ

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.

- **** 0'03 SYSG FAM <171F25 > INITIALIZED **

The FAM running on system SYSG has been started up.

- **** 8'38 SYSB FAM <171F25 > OPERATIONAL**

The FAM running on system SYSB is in working order and is in communication with your system.

- **** 8'39 SYSG FAM <171F25 > OPERATIONAL**

The FAM running on system SYSG is in working order and is in communication with your system.

▶ FIXRAT

The FIXRAT command has been replaced by FIX_DISK at Rev. 19. FIXRAT may only be used on disks made prior to Rev. 19. (See Chapter 14 for a complete description of FIXRAT.)

▶ REMOTE

The REMOTE command has been eliminated at Rev. 19.

The REMOTE command is used by FAM I to permit or deny access to local disks by specific network nodes or by the entire network. For a discussion of REMOTE, see the section on setting access to local disks (REMOTE), above.

▶ STARTUP [PROTECT] comdev [pdev-1...pdev-8]

The STARTUP command has been replaced by ADDISK at Rev. 19.

The STARTUP command defines a list of physical devices to be used by PRIMOS. 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, and if no user has assigned a disk specified in the list.

comdev and pdev-n... are items in a list of physical disk (device) numbers. The argument comdev must be specified in the initial STARTUP command; the remaining device numbers are specified optionally. The order of the list defines the logical number sequence of the devices (e.g., comdev is logical 0, pdev-1 is logical 1, etc.)

comdev must match the comdev specified in the CONFIG command. Example:

```
STARTUP 52 53 54
```

defines that physical devices 52, 53, and 54 are to be used with PRIMOS and associates the following logical device numbers with the physical device numbers specified: 52 is logical 0; 53 is logical 1; and 54 is logical 2. In PRIMOS 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.

Examples:

    STARTUP 0/52 1/53 2/54

    STARTUP 4/100250

If a nonexistent or not ready device is specified, the message

    DISK pdev NOT READY

will be printed at the supervisor terminal or, if the device is assigned, at the terminal of the user to whom the device is assigned.

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 eight logical devices can be connected by STARTUP; additional devices must be connected with the ADDISK command. ADDISK is normally preferred to STARTUP.

Write protection: The option PROTECT is specified to assign write protection for pdev.

PROTECT may be specified only for disks that are added locally via the ADDISK or STARTUP command and for disks on which the write-protection status can be set.

The status of the write protection assignments may be changed for a given pdev by shutting down the disk and then respecifying the ADDISK or STARTUP command with or without the PROTECT option. (A STARTUP or STARTUP PROTECT to a running disk does not change the protection.)

INDEX

Index

USER SURVEY

Tell us how we're doing, and we'll send you a free Programmer's Companion.

Your name ______________________________

Company or School ______________________________

Address ______________________________

City, State, Zip ______________________________

---

1. What is your job title or function? ______________________________

2. What specific task describes what you do? ______________________________

3. Does your company or school own a Prime computer? ☐ YES ☐ NO

a. If YES, which model? ☐ 450 ☐ 550 ☐ 650 ☐ 750 ☐ OTHER

b. Is it networked with other Prime computers? ☐ YES ☐ NO

c. Is it networked with any of these? ☐ IBM ☐ CDC ☐ UNIVAC ☐ HONEYWELL

d. Which of these software packages do you use?

☐ FORTRAN	☐ COBOL	☐ BASIC/VM
☐ FORTRAN 77	☐ PL/I-G	☐ POWER
☐ MIDAS	☐ DBMS	☐ SPSS
☐ RPGII	☐ FORMS	☐ PRIMENET
☐ RJE	☐ PASCAL	☐ OAS
☐ DBG	☐ DPTX	

e. Have you read any other Prime documents? ☐ YES ☐ NO

f. If YES, which ones? ______________________________

4. Are you presently evaluating Prime? ☐ YES ☐ NO

Is the documentation playing a part? ☐ YES ☐ NO

5. What book are you reviewing? ______________________________

6. My initial reaction to this book was: ☐ EXCELLENT ☐ GOOD ☐ FAIR ☐ VERY GOOD ☐ FAIR

7. After reading it my reaction was: ☐ BETTER ☐ THE SAME WORSE

If BETTER or WORSE why? ______________________________

8. How often have you used this book? ☐ EVERY DAY ☐ FAIRLY OFTEN ☐ VERY OFTEN ☐ JUST GOT IT

9. Did the book have the content you expected? ☐ YES ☐ NO

If NO, why? ______________________________

10. Did you find the organization useful? ☐ YES ☐ NO

If NO, why? ______________________________

11. How did you find the examples? ☐ TOO MANY ☐ ABOUT RIGHT ☐ TOO FEW

12. How did you find the illustrations? ☐ TOO MANY ☐ ABOUT RIGHT ☐ TOO FEW

13. Could you locate the information you needed? ☐ YES ☐ NO

14. Was the index adequate? ☐ YES ☐ NO

15. Please evaluate our writing style and edit quality?

a. Clarity ☐ VERY CLEAR ☐ AVERAGE ☐ UNCLEAR

b. Tone ☐ CORRECT ☐ NEUTRAL ☐ STILTED

c. Technical level ☐ TOO HIGH ☐ ABOUT RIGHT ☐ TOO LOW

d. General writing quality ☐ EXCELLENT ☐ AVERAGE ☐ FAIR ☐ POOR

e. Editorial quality ☐ EXCELLENT ☐ AVERAGE ☐ FAIR ☐ POOR

16. Which of these manufacturers documentation have you used the most?

☐ IBM ☐ DIGITAL ☐ DG
☐ HP ☐ CDC ☐ UNIVAC
☐ HONEYWELL ☐ BURROUGHS ☐ WANG
☐ GE ☐ XEROX ☐ ________

How is Prime compared to theirs?

☐ MUCH BETTER ☐ A LITTLE WORSE
☐ A LITTLE BETTER ☐ MUCH WORSE
☐ SAME

Any comment on your rating? ________________________

17. Please evaluate the graphic quality:

a. Rate the general presentation ☐ EXCELLENT ☐ GOOD ☐ POOR ☐ VERY GOOD ☐ FAIR

b. Do you like the paper color? ☐ YES ☐ NO

c. Did you find any quality defects? ☐ YES ☐ NO

d. Do you like the way we've used color for abbreviations and user input? ☐ YES ☐ NO

e. Did the shading over the tables and charts make them EASIER or HARDER to read? ☐ EASIER ☐ HARDER

f. Do you like the Programmer's Companion concept? ☐ YES ☐ NO ☐ HAVEN'T SEEN ONE

g. Which form of bindery do you find most useful? ☐ BOUND ☐ LOOSE-LEAF

18. Do you know about the AIDUS program? ☐ YES ☐ NO

19. Any other comments:

____
____
____
____
____
____
____
____
____
____
____

20. What book don't we offer that you'd like to see? ____________________________________

Thank you for filling out the survey. Check off which Programmer's Companion you would like to receive.

☐ PRIMOS
☐ FORTRAN 77
☐ ASSEMBLY
☐ ADMINISTRATOR
☐ FORTRAN
☐ BASIC/VM
☐ POWER
☐ WORD PROCESSING

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