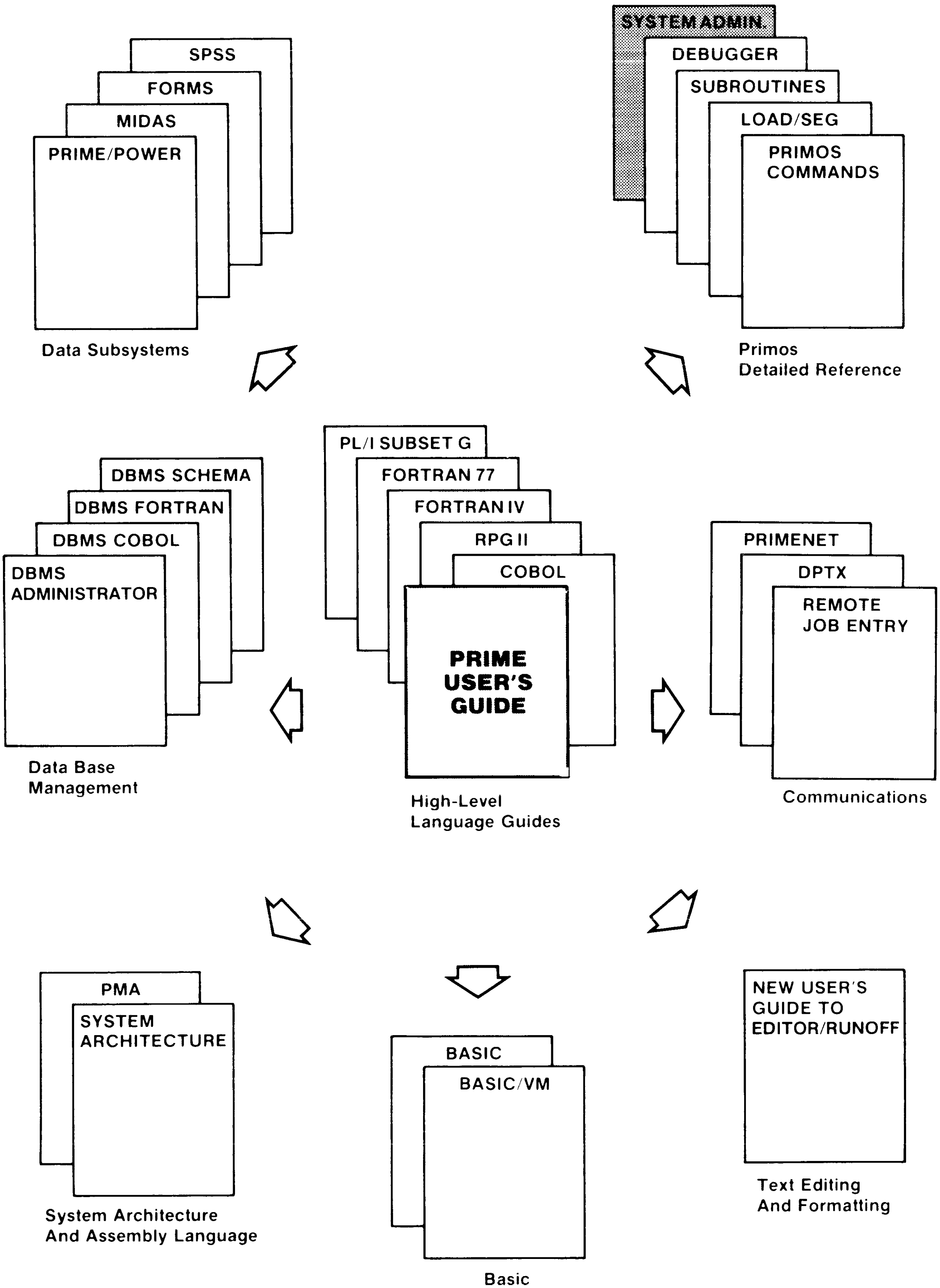
PRIME Computer

THE SYSTEM ADMINISTRATOR'S GUIDE PDR 3109

MAN 3251-001


SYSTEM ADMIN.
DEBUGGER
SUBROUTINES
LOAD/SEG
PRIMOS COMMANDS
Primos Detailed Reference
SPSS
FORMS
MIDAS
PRIME/POWER
Data Subsystems
PL/I SUBSET G
FORTRAN 77
FORTRAN IV
RPG II
COBOL
PRIME USER'S GUIDE
High-Level Language Guides
DBMS SCHEMA
DBMS FORTRAN
DBMS COBOL
DBMS ADMINISTRATOR
Data Base Management
PRIMENET
DPTX
REMOTE JOB ENTRY
Communications
PMA
SYSTEM ARCHITECTURE
System Architecture And Assembly Language
BASIC
BASIC/VM
Basic
NEW USER'S GUIDE TO EDITOR/RUNOFF
Text Editing And Formatting

THE SYSTEM ADMINISTRATOR'S GUIDE PDR3109

This guide documents the software operation of the Prime Computer and its supporting systems and utilities as implemented at Master Disk Revision Level 17 (Rev. 17).

PRIME

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

ACKNOWLEDGEMENTS

We wish to thank the members of the documentation team and also the non-team members, both customer and Prime, who contributed to and reviewed this book

First Printing February 1979
Second Printing, Revised January 1980

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

Anthony R. Lewis
Alice S. Landy
Rosemary Shields
Technical Publications Department
Prime Computer, Inc.
500 Old Connecticut Path
Framingham, Massachusetts 01701

CONTENTS

PART I - OVERVIEW

20 PRIMOS

APPENDICES

A BOOTSTRAPS

B VIRTUAL CONTROL PANEL

C CONTROL PANEL

D PHYSICAL DEVICE NUMBERS

E DISK ERRORS

F BATCH MESSAGES

G THE EVENT LOGGER

PART I - OVERVIEW

SECTION 1

OVERVIEW

INTRODUCTION

This document is a comprehensive guide for the System Administrator and Operators of a Prime computer installation. It contains everything normally necessary for operation of a Prime computer. The entire guide is written for the System Administrator; part of the guide is written specifically for the Computer Operator. At some installations, these two roles may be performed by the same person. The guide is designed to:

- Enable the System Operator to run the computer efficiently and optimally with a minimum of supervision by the Administrator.
- Provide the Operator with sufficient information to cope with certain emergency situations and to be aware of those which require the aid of the Administrator.
- Provide the Administrator with sufficient information to make choices about system building, option selection, resource allocation and system expansion.
- Serve as an archival repository for operator-oriented system information: PRIMOS, control panel, commands, bootstraps, etc.

This Version

This is an Preliminary Documentation Release, documenting the operation of Prime computer systems and supporting utilities at software revision level 17.2.

Roles of the Operator and Administrator

The roles as defined by the tasks below are suggested for guidance only. At some installations, some of the operator tasks may be assigned to the Administrator, and vice versa.

Operator: The following are the general duties of the System Operator:

- Be familiar with the operation of the system, its utilities, and features.
- Make the system available to the user community. This includes: bringing up the system, starting any special subsystems.
- Monitor the system status.
- Monitor the supervisor terminal and respond appropriately to user requests (e.g., mount tape on unit 2, etc.)
- Monitor the equipment and environment.
- Make sure that hardware preventive maintenance, filter changes and disk cleaning are performed as scheduled by the Administrator.
- Modify the system under the supervision or control of the Administrator.
- Perform normal housekeeping functions: file integrity checking (FIXRAT), backing up.
- Perform tasks for users: printing on special forms, disk formatting.
- Maintain the system logbook.

Administrator: The following are general tasks of the System Administrator.

- Build the system, possibly with the aid of a Prime field analyst.
- Configure the local operating system, optional software systems, etc.
- Configure environments for printers, Batch subsystem, etc.
- Build the network configuration file for the local system if network support has been purchased.
- Build the DPTX configuration file if the distributed processing terminal executive has been purchased.
- Establish/change system and utility defaults, if so desired.

- Allocate the resources of the systems among users: disk space, shared segments, number of users, terminal locations.
- Arrange for and schedule preventive maintenance for hardware, peripherals, disk cleaning.
- Arrange for instruction of operators.

Organization

This guide is composed of four major parts:

Part 1: An introductory section that defines the roles of the System Administrator and Operator, and refers to other sections containing information supporting these roles.

This section also covers typographic and Prime conventions used in this guide (Section 1).

Part 2: The Operator. This part covers the normal functions of the operator as we have defined the role. It starts with the basic operator function of making the system available to users and continues with operator tasks supporting this function. The order of information presentation is:

1. System Startup/Shutdown (Section 2)
2. Operator Tasks (Section 3)
3. Monitoring System Status (Section 4)
4. Checking File System Integrity (Section 5)
5. Formatting Disks and Tapes (Section 6)
6. Backing Up (Section 7)
7. Using the Spooler (Section 8)
8. Monitoring and controlling the Batch queues (Section 9)
9. Handling System Crashes (Section 10)

Prime system utilities and all PRIMOS-level commands necessary to perform these tasks are presented in this Part, either in complete detail or with references to the appropriate section in Part 4 or the Appendices. In most cases, the operator will not need to refer to any software document other than this one.

Part 3: The Administrator. In general, these sections discuss options available on the system and strategies for Administrator decision-making. Some of the generic tasks covered are:

- Building and Upgrading the System (Section 11)
- Defining the Batch queue environments (Section 12)
- Allocation of System Resources (Section 13)
- System Default and Parameter Values (Section 14)

A final section covers some more specific topics which historically have been of interest to Prime System Administrators (Section 15).

Part 4: Reference Sections 16 and 17 form a complete reference for operator-oriented PRIMOS commands. Many of these commands may be used only at the supervisor terminal.

Section 18 is a brief overview of PRIMENET and DPTX tasks for operators and administrators.

Section 19 is a brief view of PRIMOS II, Prime's single user operating system, which is used to start up the multi-user operating systems. This section also includes a discussion of single-user operations. Section 20 covers the PRIMOS operating system. This system utilizes the advanced architecture and segmentation available in the Prime 350 and higher computers.

Related Documents

The following documents contain detailed reference information useful to the Prime computer system Operator and Administrator:

- The Prime User's Guide, IDR4130
- Reference Guide, PRIMOS Commands, FDR3108
- Subroutine Reference Guide, PDR3621
- System Administrator's Programmer's Companion, FDR3622

If the appropriate software packages have been purchased:

- The PRIMENET Guide, IDR3710
- The Distributed Processing Terminal Executive Guide, IDR4035

Manuals and guides supplied with peripherals (line printers, tape drives, etc.), should be maintained in the computer room for use by the operators.

TYPOGRAPHIC CONVENTIONS

The conventions for PRIMOS command documentation are:

- WORDS-IN-UPPER-CASE

Capital letters identify command words or keywords. They are to be entered literally. If a portion of an upper-case word is underlined, the underlined letters indicate the minimum legal abbreviation.

- Words-in-lower-case

Lower case letters identify parameters. The user substitutes an appropriate numerical or text value.

- Braces { }

Braces indicate a choice of parameters and/or keywords. Unless the braces are enclosed by brackets, at least one choice must be selected.

- Brackets [ ]

Brackets indicate that the word or parameter enclosed is optional.

- Hyphen -

A hyphen identifies a command line option, as in: SPOOL -LIST

- Parentheses ( )

When parentheses appear in a command format, they must be included literally.

- Ellipsis ...

The preceding parameter may be repeated.

- Angle brackets < >

Used literally to separate the elements of a pathname. For example: <FOREST>BEECH>BRANCH537>TWIG43>LEAF4.

- option

The word option indicates one or more keywords or parameters can be given, and that a list of options for the particular command follows.

- Spaces

Command words, arguments and parameters are separated in command lines by one or more spaces. In order to contain a literal space, a parameter must be enclosed in single quotes. For example, a pathname may contain a directory having a password:

'<FOREST>BEECH SECRET>BRANCH6'.

The quotes ensure that the pathname is not interpreted as two items separated by a space.

User input usually may be either in lower case or in UPPER CASE. The rare exceptions will be specified in the commands where they occur.

PART II - THE OPERATOR

SECTION 2

SYSTEM STARTUP AND SHUTDOWN

INTRODUCTION

The major function of the operator is to make the computer system available to the user community. The Prime system is designed to allow the operator to perform this function simply, easily, and reliably.

This section takes the operator through an example of startup and actual usage on a Prime computer. This example has many more operations and subsystems than most users would have. If a subsystem is not implemented on your system, you may ignore it. We have shown operator answers to system prompts that are specific to the example; space is provided to write in the appropriate response for your own system.

VIRTUAL CONTROL PANEL (VCP)

Some Prime computer systems have a virtual control panel (VCP). This feature allows the supervisor terminal to enter a mode in which control panel functions (sense switch setting, master clearing, etc.) can be performed by commands entered at the supervisor terminal. If your system has a virtual control panel, use the commands which are equivalent to the switch operations described in this section. Virtual control panel operation is discussed in Appendix B.

START UP

The general order of start up procedure is:

- Turn on power to equipment.
- Bootstrap in PRIMOS II.
- Attach to UFD PRIRUN and resume PRIMOS.
- If the file C_PRMO is in UFD CMDNC0, it will specify system configuration and set any other parameters and conditions and startup any subsystems. (The file C_PRMO is built by the Administrator according to system needs.)
- If the file C_PRMO has not been installed, the operator must manually enter the configuration and subsystem startup commands.
- Set system time and date.

Turning on Equipment

Turn on power to equipment in this order:

- Turn on power to CPU by turning keyswitch on control panel to ON.
- Turn on all disk drives; wait for ready lights to go on.
- Turn on magnetic tape drives which are to be used.
- Turn on all dataphone sets; dataphones should be set to automatic answer.
- Other peripheral devices, such as line printers, may also be turned on at this time.

An attempt to boot in PRIMOS II 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), and repeat the boot procedure.

Booting in PRIMOS II

After the equipment has been turned on and is ready, bootstrap the PRIMOS II operating system.

- Turn the CPU front panel rotary selector switch to STOP/STEP.
- Press MASTER CLEAR switch.
- Turn the ADDRESS/DATA switch to ADDRESS.
- If booting from a storage module disk pack, set '000114 in the sense switches (lift sense switches 10,13,14 in the up position).
- If booting from a cartridge disk, set '000014 in the sense switches (lift sense switches 13 and 14 in the up position).
- Turn the selector switch to LOAD.
- Press the START switch.
- Turn rotary switch to RUN.
- Place all sense switches down.

Note

See Appendix A for details of booting PRIMOS II from devices other than disks. See Appendix C for details of control panel operation.

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

PHYSICAL DEV= 460 (your command device= ________)

Enter the physical device number of the command device (Appendix D explains the construction of physical device numbers). The command device is the one whose command UFD is searched when a PRIMOS command is given.

The device number given here is for a 2-head partition of a storage module; the partition is the first one on the module. If yours is different, write it in the space provided.

After the device number is entered, the system responds with:

```
PRIMOS II REV 16.8 07/02/79 (AT 170000)
STARTING UP DISK 000460

OK:
```

The date is that of the revision, not the current date. The number in parentheses is the starting location in memory of PRIMOS II. This number changes dynamically with Revision.

The second line of the response indicates that PRIMOS II is automatically starting up the first partition on the physical device whose number it has been given. Usually, this is the only partition needed until PRIMOS is brought up. (If another partition is needed under PRIMOS II, it can be added with the STARTUP command. See Section 19 for details.)

At this point, the operator can use PRIMOS II for backups, formatting, etc. See Section 19 for complete details.

Starting up PRIMOS

Attach to UFD PRIRUN and RESUME PRIMOS. In this example, the operating system includes many separately priced software packages including SPSS, COBOL, BASIC/VM, PRIMENET (network), etc.

```
OK: ATTACH PRIRUN
OK: R PRIMOS
```

Configurations

After the command R PRIMOS, the operating system begins running; it attaches to UFD CMDNC0 and looks for the file C_PRMO. If this file exists, configuration commands will be read from it; if it does not exist, the system will print PLEASE ENTER CONFIG on the supervisor terminal, and commands must be entered from the supervisor terminal. In either case, the first command (either in C_PRMO or entered at the supervisor terminal) must be a CONFIG command. (See Section 17 for details on the CONFIG command.)

Example of Configuration Using C_PRMO

The following example shows the dialog ensuing when the C_PRMO command file is present. (Use of this file is strongly recommended.) Explanatory comments follow their respective actual terminal printout. The contents of the startup command file C_PRMO (and the configuration data file, CONFIG, which it invokes) appear at the end of this section.

After the command R PRIMOS, the system responds with:

```
GO
CONFIG -DATA CONFIG
PRIMOS 17.2
-------------------------

2048K BYTES MEMORY IN USE


NTUSR 60
```

Sets number of terminal users (decimal 48).

```
COMDEV 460
```

Command device - a storage module on drive unit 0, partition is two heads at start of pack.

```
MAXPAG 2000
```

Number of pages of physical memory to validate at cold start.

```
NAMLC 2
```

Sets number of assignable AMLC lines.

```
NPUSR 7
```

Sets number of phantom users allowed.

NRUSR 10

Sets number of remote users allowed (decimal 8).

Note

Total number of configured users is 60 + 7 + 10 = 77 (63 decimal).

NSEG 500

Specifies the total number of segments available to the system, PRIMOS, and all users: local, phantom, and remote ('500 is 320 decimal).

PAGDEV 40462 34736

Sets paging space on paging device to '34736 (14814 decimal) records.

TYPOUT YES

Causes subsequent commands in config data file to be printed at the supervisor terminal.

NET ON

Opens the network configuration file (NETCON) for information to configure the local system in the network. This file is created by the NETCFG command (See Section 18). Network support is a separately priced software package.

LOGLOG YES

Allows users who are logged in to use the LOGIN command without having to log out first.

LOGMSG YES

Enables printing of LOGIN and LOGOUT messages.

LOGREC 0

Disables quota checking in Event Logger. See Section 4 for the operation of the Event Logger, and Appendix G for details of installation.

```
LOUTQM 74
```

Sets inactivity logout time to '74 minutes (60 decimal=1 hour).

```
GO
```

Indicates end of configuration data file. The rest of the command file C_PRMO is now processed.

```
PLEASE ENTER DATE
OK, ADDISK 460 12060 52061 2062 61463
```

Connects additional physical devices to the system. The command device is logical device 0; these disks are assigned the next available logical device numbers, 1 - 5.

```
OK, AMLC TTY 0 2213        /* 300 BAUD TELCO LINE
OK, AMLC TTY 1 2213        /* 300 BAUD TELCO LINE
OK, AMLC TTY 2 2213        /* 300 BAUD TELCO LINE
```

Configures AMLC lines 0-2. For example, the configuration word, 2213, specifies 300 Baud, parity disabled, and 8-bit character length. (The default value is 1200 baud.) See Section 16 for full details of the AMLC command.

```
OK, ELIGTS 1               /* SHORTEN ELIGIBILITY TIME SLICE
```

Changes the eligibility time slice parameter of the scheduler. See Section 16 for more details.

```
OK, OPR 1
```

Sets operator priority to allow sharing.

```
OK, SHARE SYSTEM>ED2000 2000          /* SHARE EDITOR
```

Shared Editor.

```
OK, SHARE SYSTEM>SP2004 2004          /* SHARE SPSS
OK, SHARE SYSTEM>SP2005 2005
OK, SHARE SYSTEM>SP2006 2006
OK, SHARE SYSTEM>SP2007 2007
OK, SHARE SYSTEM>SP2010 2010
OK, SHARE SYSTEM>SP2011 2011
```

Shared version of Statistical Package for the Social Sciences (SPSS). SPSS is a separately priced software package.

```
OK, SHARE SYSTEM>BA2013 2013          /* SHARE BASIC
```

This is the shared version of BASIC/VM. BASIC/VM is a separately priced software package.

```
OK, /* BRING UP SHARED LIBRARIES
```

The shared libraries are: FORMS, MIDAS, COBOL, and FORTRAN. If any of these are installed, then all must be installed. (See Section 3 on Shared Segments and Section 11 on Shared Libraries).

```
OK, SHARE SYSTEM>S2050 2050
OK, R SYSTEM>S4000
THIS IS PACKAGE #1.

OK, SHARE SYSTEM>K2014A 2014 700
OK, SHARE SYSTEM>K2014B 2014 700
OK, R SYSTEM>K4000
THIS IS PACKAGE #0.

OK, SHARE 2020 700
OK, R SYSTEM>IMIDAS
[IMIDAS rev 17.2]

Initialization complete

MIDAS is started up.

OK, SHARE SYSTEM>C2014A 2014 700
OK, SHARE SYSTEM>C2014B 2014 700
OK, R SYSTEM>C4000
THIS IS PACKAGE #3.

OK, SHARE SYSTEM>F2014A 2014 700
OK, SHARE SYSTEM>F2014B 2014 700
OK, R SYSTEM>F4000
THIS IS PACKAGE #4.

OK, SHARE 2014
```

Sets shared library access rights to read and execute.

```
OK, SHARE SYSTEM>CO2016 2016          /* SHARE COBOL
OK, SHARE SYSTEM>DB2040 2040          /* SHARE DEBUGGER
OK, SHARE SYSTEM>DB2041 2041
OK, SHARE SYSTEM>DB2042 2042
OK, SHARE SYSTEM>PO2024 2024 600      /* PRIME POWER
OK, SHARE SYSTEM>PO2025 2025 600
OK, SHARE SYSTEM>PG2044 2044          /* SHARE PL1 subset G
```

```
OK, SHARE SYSTEM>PG2045 2045
OK, SHARE SYSTEM>PG2051 2051
OK, SHARE SYSTEM>FT2046 2046           /* SHARE FORTRAN 77
OK, SHARE SYSTEM>FT2047 2047
OK, SHARE SYSTEM>FT2052 2052
```

```
OK, PH FAM>PH_FAM                     /* START FAM
```

Run File Access Manager (FAM) phantom command file.

```
PHANTOM IS USER 57
```

PRIMOS returns phantom user number.

```
OK, CHAP -57 2 /* THIS ENTRY MUST BE CHANGED IF NO OF USERS CHANGES
```

Use the CHAP command (Section 16) to change the priority of the FAM to 2. If the FAM user number, here 57, is known from previous startups, this command may be incorporated in C_PRMO. If this is the first startup for a configuration, the user number will not be known in advance. Note the FAM user number in the logbook and, after startup is completed, change the priority with the CHAP command. FAM is a separately priced software package.

```
OK, PH BATCHQ>PH_GO /* START BATCH MONITOR
```

Start the Batch subsystem.

```
PHANTOM IS USER 58
```

PRIMOS returns phantom user number.

```
OK, CO DBMSLB>C_SHARE 7      /* BRING UP DBMS
```

Open the command input file C_SHARE in UFD DBMSLB for input on file unit 7. The unit must be specified to avoid conflict as C_PRMO has been opened on the default unit number 6.

```
OK, OPR1
OK, SHARE DBMSLB>B2001 2001 700
OK, SHARE DBMSLB>B2012 2012 700
OK, SHARE DBMSLB>B2003 2003
OK, R DBMSLB>DB4000 1/5
GO
THIS IS PACKAGE #5.

OK, SHARE 2001
```

```
OK, SHARE 2012
OK, SHARE DBMSLB>DB2002 2002 700
OK, SEG DBMSLB>#IDBMS
GO
IDBMS  -  INITIALIZED DBMS <REV16.2>
SHARED MEMORY INITIALIZED (128 FILES)
SEMAPHORES SET

OK, OPR
OK, CO -CONTINUE 6
```

Return to command file C_PRMO.

```
OK, CLOSE 7


PROP E.PR0 -START        /* START SPOOLER
[PROP rev 17.2]

Wait... Acknowledged.
```

Starts phantom for first printer.

```
/*PROP E.PR1 -START     /* START SPOOLER2
```

(This command has been turned into a comment by adding the leading "/*" to the C_PRMO file. The second printer will start only if the operator starts it or if the C_PRMO file is re-edited to remove the /*.)

```
OK, A CMDNC0
```

Note

External commands will not execute from user terminals unless the supervisor terminal is attached to UFD CMDNC0.

```
OK, /* SET THE DATE AND TIME **********
OK, /* TYPE CO SYSTEM>REMOTE TO ADD DISKS ACROSS THE NETWORK ******
**
OK, OPR 0
OK, RDY -LONG
OK 00:02:47 16:542 32.300      Change to long form of prompt message
CO TTY                                Accept input from the terminal
```

Note

This completes the processing of commands from C_PRMO. Other startup functions proceeding from this point are entered by the supervisor terminal directly at the supervisor terminal.

Set System Time and Date

OK. SETIME 052478 0640

Operator sets date and time (See Section 16).

Bring Up Remote Disks

OK, CO SYSTEM>REMOTE

Run command input file REMOTE in UFD SYSTEM

OK, /*Add ENB disks over the ring net
OK, ADDISK ENB 460 3462 71063 10460 20460

Logical devices 6, 7, 10, 11, 12.

OK, /* Add ENG disks over the ring net.
OK, ADDISK ENG 10460 460 32060

Logical devices 13, 14 15.

OK, CO -TTY

Accept input from terminal

Start Batch Monitor

OK BATCH SYSTEM -START

The phantom for the Batch monitor is started within C_PRMO, but the monitor itself can't be started until the system time and date have been set.

PRIMOS Messages

During the start up 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. For clarity of presentation, these messages have been gathered together below.

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 END FAM <171F25 > OPERATIONAL**

FAM NODES ENABLED:
    ENA
    ENB
    ENC
    ENE
    ENF
    ENG
    RES.C1
    TE.CAD
    TE.PCB
    TE.PC2
    EN.D3
    EN.D5
    EN.D6
    EN.D8
```

The local FAM program is in operation.

```
**** 0'03 ENB FAM <171F25 > INITIALIZED **
```

The FAM running on ENB has been started up.

```
**** 0'03 ENG FAM <171F25 > INITIALIZED **
```

The FAM running on ENG has been started up.

```
***57 7156
*BATCH* Waiting for BATCH SYSTEM -START
```

The phantom that controls the Batch monitor has been started, and the monitor is ready to accept the BATCH SYSTEM -START command which allows it to begin processing users' jobs.

```
**** 8'38 ENB FAM <171F25 > OPERATIONAL**
```

The FAM running on ENB is operational.

```
**** 8'39 ENG FAM <171F25 > OPERATIONAL**
```

The FAM running on ENG is operational.

SYSTEM SHUTDOWN

Before shutting down the system, it is good operational procedure to inform local users and operators at other PRIMENET nodes (if any) using MESSAGE (See Section 16). The shut down procedure follows.

Send Shut Down Messages

```
OK, MESSAGE ALL NOW
SYSTEM GOING DOWN IN 5 MINS. PLEASE LOGOUT
                                         inform operators of any remote
                                         machines
OK, MESSAGE RES.Cl
END GOING DOWN IN 5 MINS.
OK, MESSAGE ENB
END GOING DOWN IN 5 MINS.
OK, MESSAGE ENE
END GOING DOWN IN 5 MINS.
OK, MAXUSR Ø                               prevent any logins
                                         4 minutes later
OK, MESSAGE ALL NOW
*** SYSTEM GOING DOWN IN 1 MINUTE *** (bells)
                                         ring bells by typing CONTROL-G
OK,
```

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

Stop Batch Subsystem

```
BATCH SYSTEM -STOP
[Batch rev 17.2]
Stop request issued.

OK, ***58 6'39
*BATCH* Operator stop.
```

The operator gives the BATCH SYSTEM -STOP command to allow the Batch monitor to log itself out gracefully.

When the message, "*BATCH* Operator Stop" is received, the monitor has logged out and the Batch subsystem is no longer running.

System Shutdown

```
LOGOUT ALL
OK, SHUTDN ALL
REALLY? YES                         any answer but YES aborts shutdown
WAIT,
LDEV Ø DETACHED, YOUR FILES CLOSED.
PRIMOS NOT IN OPERATION
```

Equipment Shutdown

Equipment power may now be turned off, in the reverse order as for start up:

- 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 by turning the keyswitch on the control panel to OFF.

CONFIGURATION DATA FILE

The configuration data file used in this example is listed below. The initial comment lines and the ASRATE command were not printed out during startup as they precede the TYPOUT YES command. The normal condition is not to type out configuration commands as they are processed. (The ASRATE command in this file sets the supervisor terminal to 300 Baud.)

```
/* CONFIG DATA FOR END, WAC, 09/25/78
/* MODIFIED FOR NEW PAGING DISK
ASRATE 1010
TYPOUT YES
NTUSR 60
COMDEV 460
MAXPAG 2000
NAMLC 2
NPUSR 7
NRUSR 10
NSEG 500
PAGDEV 40462 34736
TYPOUT YES
NET ON
LOGLOG YES
LOGMSG YES
LOGREC 0
LOUTQM 74
GO
```

START UP COMMAND FILE

Below is the command file C_PRMO used as an example in this section. The -DATA option, CONFIG, uses the configuration file above. This file contains statements for the PRIMOS static configuration. The remainder of the command file (C_PRMO) sets the dynamic configuration.

Dynamically configurable features can be changed while PRIMOS is in operation.

```
CONFIG -DATA CONFIG
ADDISK 460 12060 52061 2062 61463
AMLC TTY 0 2213        /* 300 BAUD TELCO LINE
AMLC TTY 1 2213        /* 300 BAUD TELCO LINE
AMLC TTY 2 2213        /* 300 BAUD TELCO LINE
ELIGTS 1                 /* SHORTEN ELIGIBILITY TIME SLICE
OPR 1
SHARE SYSTEM>ED2000 2000     /* SHARE EDITOR
SHARE SYSTEM>SP2004 2004     /* SHARE SPSS
SHARE SYSTEM>SP2005 2005
SHARE SYSTEM>SP2006 2006
SHARE SYSTEM>SP2007 2007
SHARE SYSTEM>SP2010 2010
SHARE SYSTEM>SP2011 2011
SHARE SYSTEM>BA2013 2013          /* SHARE BASIC
/* BRING UP SHARED LIBRARIES
SHARE SYSTEM>S2050 2050
R SYSTEM>S4000
SHARE SYSTEM>K2014A 2014 700
SHARE SYSTEM>K2014B 2014 700
R SYSTEM>K4000
SHARE 2020 700
R SYSTEM>IMIDAS
SHARE SYSTEM>C2014A 2014 700
SHARE SYSTEM>C2014B 2014 700
R SYSTEM>C4000
SHARE SYSTEM>F2014A 2014 700
SHARE SYSTEM>F2014B 2014 700
R SYSTEM>F4000
SHARE 2014
SHARE SYSTEM>CO2016 2016          /* SHARE COBOL
SHARE SYSTEM>DB2040 2040          /* SHARE DEBUGGER
SHARE SYSTEM>DB2041 2041
SHARE SYSTEM>DB2042 2042
SHARE SYSTEM>PO2024 2024 600      /* PRIME POWER
SHARE SYSTEM>PO2025 2025 600
SHARE SYSTEM>PG2044 2044
SHARE SYSTEM>PG2045 2045          /* SHARE PL1 subset G
SHARE SYSTEM>PG2051 2051
SHARE SYSTEM>FT2046 2046          /* SHARE FORTRAN 77
SHARE SYSTEM>FT2047 2047
SHARE SYSTEM>FT2052 2052
PH FAM>PH_FAM 76           /* START FAM
CHAP -57 2 /* THIS ENTRY MUST BE CHANGED IF NO OF USERS CHANGES
PH BATCHQ>PH_GO /* START BATCH MONITOR
CO DBMSLB>C_SHARE 7      /* BRING UP DBMS
CLOSE 7
PROP E.PR0 -START    /* START SPOOLER
/* PROP E.PR1 -START    /* START SPOOLER 2
CLOSE 7
A CMDNC0
```

```
* SET THE DATE AND TIME **********
* TYPE CO SYSTEM>REMOTE TO ADD DISKS ACROSS THE NETWORK **********
OPR Ø
RDY -LONG
CO -END
```

SECTION 3

OPERATOR TASKS

INTRODUCTION

In addition to starting up the system and shutting it down, there are a number of tasks routinely performed by the operator. These may be initiated by the operator, by a user or administrator request, or by the system. The task groups are:

- Allocation and Monitoring of Resources
- Subsystem Usage
- System Halts - prevention and recovery
- Direct user interactions, such as:
  - Line printer operations
  - Sending messages
  - Adding New UFDs to the System
  - Adding/Replacing Software in CMDNC0 and LIB
  - Mag Tape Assignments

ALLOCATION AND MONITORING OF RESOURCES

Changing the Assignable Disks Table

This is not often done by the operator. Before a disk or partition can be assigned, its physical device number must be added to the Assignable Disks Table by the DISKS command (section 16). Once this is done, the disk can be assigned by:

<u>AS</u>SIGN DISK physical-device-number

Under PRIMOS, devices should always be assigned to the user prior to a MAKE, FIXRAT, or COPY 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 a space for a maximum of 10 devices. Devices may be removed from the Table by the DISKS command (Section 16).

Changing Priority/Timeslice

To increase efficiency and/or system performance, priorities or timeslices can be changed. For example, the File Access Manager (FAM) is set up to run at priority 2. (Normal process priority is 1.) Priority and timeslice are changed by the CHAP command which is described in Section 16.

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:

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

pathname — The file to be restored into segment-number.

segment-number — The segment to be shared. Segments '2030 to '2037 are specifically reserved for user shared subsystems.

access-rights — User access to the segment. Default is '600 - read and execute rights.

See Section 16 for complete details. The Administrator will assign and coordinate shared segment usage.

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 Table 3-1.

Table 3-1. Contents of Shared Segments.

Segment	Product
2000	Editor (0-160000)
2001-2003	DBMS
2004-2011	SPSS
2012	DBMS
2013	BASIC/VM
2014	Shared libraries (note 1)
2015	DPTX
2016	COBOL
2017	BASIC/VM
2020	MIDAS writable shared segment
2021-2023	Reserved for Prime
2024-2025	PRIME/POWER
2026-2027	Reserved for Prime
2030-2037	Reserved for users
2040-2042	DBG
2043	SPSS
2044-2045	PL/I-G
2046-2047	FORTRAN 77
2050	V-FTNLIB
2051	PL/I-G
2052	FORTRAN 77
2053-2077	Reserved for Prime
6001	Per-user linkage segment (note 2)

Notes

1. Segment 2014

Allocated	Product
100-177	FORMS library
500-677	COBOL library
700-777	MIDAS library
1000-67777	FORMS library
70000-117777	COBOL library
120000-157777	MIDAS library

2. Segment 6001

Allocated	Product
0-30277	FORMS
30300-32777	ED
33000-40777	COBOL
41000-67777	MIDAS
120000-131777	ABBREV
132000-177777	V-FTNLIB

Monitoring System UFDs

The directories below are under the control of the operator.

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 system, PRIMOS II.
FAM	Contains the file access manager (FAM) used for remote file access (if purchased). Must be installed. (See Section 11.)
FORMS*	Contains files needed to run the Forms Management System (FORMS). Must be installed. See the FORMS Programmer's Guide.
LIB	Contains all libraries available on the system.
PRIRUN	Contains PRIMOS runfiles.
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 and PL/I-G compiler default driver programs.
SYSTEM	Contains all files for shared subsystems. Also contains DISKS file, an operator-generated file which holds list of partitions in systems. The DISKS file is used by the AVAIL * command.
PL1G>TOOLS	Contains the default driver program for the PL/I-G 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 LISTF command and may be written into a file using the COMOUTPUT command. The current contents of the system directory should be compared to the proper contents (this list should be maintained in the system logbook).

Example

```
OK, a <7>syscom
OK, listf

UFD=SYSCOM 7  OWNER

KEYS.F  KEYS.P  ERRD.F  ERRD.P  A$KEYS  PARM.K  FILD.F  FILD.P
ERRD.PL1        BDKEYS  KEYS.PL1        A$KEYS.PL1      DOCHEADER
ONCODES.P       ONCODES.PL1

OK,
```

Monitoring Disk Space

UFDs have no specific quotas limiting their size. When a partition does fill up, users often cannot perform tasks until some files are deleted.

FIXRAT and FUTIL may be used to check disk usage. When disk partitions become very full (98% and up), a list of UFDs, and the amount of space used by each, may be generated using FIXRAT (Section 5). This list should be distributed and posted. Ask the people using the most disk space to reduce the number of records they are using. Use MESSAGE to tell all users to delete unnecessary files.

SUBSYSTEM USAGE

Disk Formatting

Before a disk pack can be used on the system, it must be formatted by the MAKE utility. This process is described in detail in Section 6. The new disk or partition number must first be added to the table of assignable devices with the DISKS command. The new device can then be assigned during the MAKE procedure.

Note

> The disk pack or partition to be formatted must not be connected to the system via a STARTUP or ADDISK command.

Magnetic Tape Formatting

Magnetic tapes can be initialized and formatted with the LABEL command. This command writes volume 1 labels in either ANSI standard X3.27-1978 or IBM specifications. Complete details are given in Section 6.

FAM Operation

The FAM is a phantom that assists the operating system in performing file access across the network. FAM runs under login name FAM and is set up to run at priority 2. If the FAM has logged itself out for some reason, it must be logged in again. (FAM should not log itself out. If this occurs, note it in the system logbook and report it to the System Administrator). To log FAM in again, type PH FAM>PH_FAM. Then set the FAM priority to 2 from the supervisor terminal with the CHAP command. The sequence of operations is:

Assuming the operator is logged in at a user terminal (not the supervisor terminal):

```
OK, PH FAM>PH_FAM
PHANTOM IS USER nn
OK,
```

nn is the user number of the FAM phantom.

The operator now types, from the supervisor terminal:

```
OK, CHAP -nn 2
OK,
```

nn is the user number of the FAM phantom, the value printed by the phantom user message.

Controlling the Batch Subsystem

Generally, the System Administrator is responsible for configuring the Batch subsystem and maintaining its database. These operations are explained in Section 12. The operator is responsible for starting and stopping the Batch monitor when the system (or the Batch subsystem) is brought up and down, and for helping with user's jobs, when necessary. These tasks are explained in Section 9.

There are two main reasons for operator intervention in user jobs. If some 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 some resource, he may ask the operator to hold that job in the queue until the resource is available. When the operator knows the resource is available, he can release the job.

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 (less than the configured number of users) specifying 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 shut down. (also see Forced Logouts in this section).

Forcing Logouts

From the supervisor terminal, the operator may forcibly logout any user by the command:

```
        { ALL         }
LOGOUT  { -usernumber }
```

usernumber   The decimal number of the user being logged out.

ALL   logs out all currently logged-in users (except FAM).

The connection of a process to a terminal over the network is also forcibly broken by the command:

LOGOUT -usernumber

user-number 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 users - remote, local, and phantom, can be forcibly logged out by the command: LOGOUT ALL. In addition to logging out all users, this automatically sets MAXUSR to 1, 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.

Note

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

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, LOGREC, in UFD CMDNC0. The contents of this file are examined with the LOGPRT command. Details of the event logging system usage are in Section 4; details of its structure are in Appendix G.

It is recommended that the LOGREC file should be printed out and then purged once per day, either just after startup or just prior to shutdown. The printouts of the Event Logging file should be stored for such a length of time as the System Administrator has determined to be useful. The System Administrator can define the size of the LOGREC file using the LOGREC CONFIG directive. Default size is 4096 words. (See Section 17.) When this size is reached, a warning message is sent to the supervisor terminal. However, the file itself will continue to grow until the operator purges it.

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, NETREC, in UFD CMDNC0. The contents of this file are examined with the NETPRT command. Details of the network event logging system usage are in Section 4; details of its structure are in Appendix H.

There is no limit set on the size of the NETREC file so it should be printed out periodically to prevent excessive growth. The printouts of the Network Event Logging file should be stored for such a length of time as the System Administrator has determined to be useful.

SYSTEM HALTS: PREVENTION AND RECOVERY

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 from user or operator error or by system halt (from power failure, etc.); user error; or faulty peripherals (disk). The backup details are in Section 7.

Backing Up on Magnetic Tape

The following backup procedures are performed from the supervisor terminal.

1. Send MESSAGE to users that they will be logged out (Section 16).

2. Logout users with the LOGOUT ALL command (Section 3).

3. Mount Tape and ASSIGN tape drive unit.

4. Backup with MAGSAV (Section 7).

5. Attach to CMDNC0 (Section).

6. Reset maximum number of users with MAXUSR (Section 16).

Backing Up on Disk

1. Send MESSAGE to users that system is being shut down (Section 16).

2. Use SHUTDN ALL to logout any user (Section 16).

3. Boot in PRIMOS II (Section 2).

4. STARTUP any necessary devices (Section 16).

5. Perform backup with COPY (Section 7).

6. Bring up PRIMOS (Sections 2,20).

7. Startup additional system devices with STARTUP or ADDISK (Section 16). This is normally done with the C_PRMO command file.

Note

Back up procedures may be performed by command files with the COMINPUT command.

Recovering Backed Up Files

This is normally performed under PRIMOS after bringing up the system but before allowing users to log in. Usually this this entails simply starting up a disk and using FUTIL. The recovery procedures from disk (FUTIL) and magnetic tape (MAGRST) are described in Section 7.

Checking File System Integrity

File system and directory integrity are checked and repaired, if necessary, by the FIXRAT utility. A complete description of FIXRAT usage is in Section 5. In addition to checking the integrity, FIXRAT performs general file housekeeping by compressing directories. FIXRAT should be run on a regular basis by the operator. It has been found to be convenient to perform the FIXRAT operations in conjunction with system backups. (see Section 7 for a sample schedule).

Handling System Halts

The procedures for handling a system halt vary with the reason for the halt. This reason can usually be determined by obtaining the halt location from the control panel. The operator should become familiar with the control panel operations described in Appendix C. The handling of halts, halt location meanings, restart procedures, and information to be entered in the logbook is detailed in Section 10.

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 Section 4. In logging events:

- All entries should have time and date (including year).
- All entries should be signed. The administrator or system analyst will then know whom to ask for clarification or additional information.

DIRECT USER INTERACTIONS

Line Printer Operations

Details of the operator's Spooler commands are given in Section 8. In addition to generally overseeing the line printer(s), it is suggested that the operator is responsible for:

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

If there are problems with the line printer spooling output, 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 seeing if user SYSTEM has the line printer assigned (use STATUS USER command). If the Spooler is not running, login to UFD SYSTEM (use the owner password). Restart the spooler phantom by typing: PROP printername -START.

4. Make sure the spooler is not set up to print on special forms paper.

5. 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 (Section 16).

6. See the Systems Administrator if the printer still does not function.

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 the local node of the network
- The supervisor terminal of a different network node

This facility is useful for giving users general information (system to be shut down, new utility available), communicating with a single user (answering questions, requesting action), or for passing information between nodes (remote disk availability, etc). The command format is given in the MESSAGE command (Section 16).

Adding New UFDs to the System

New UFDs in an MFD can only be added by the operator as passwords to MFDs are not usually distributed to the users. When a request is made to add a new UFD to the system, the operator should first determine from the user: its name, and the partition on which it is to reside. The new UFD name should not duplicate any existing UFD names.

The operator should log in as a user, attach to the MFD on the appropriate partition, and generate the new UFD with the CREATE command (Section).

Since the MFD is password-protected, the operator must also perform all requests for UFD name changes (use the PRIMOS CNAME command).

ADDING/REPLACING SOFTWARE IN CMDNC0 AND LIB

These UFDs are password-protected directories under operator control. New software is copied into these directories with the COPY command of FUTIL. All new or changed software should be debugged before installation, insofar as is feasible.

All changes to these directories should be noted in the system logbook. No new or changed software should be installed without complete details of operation being obtained from the software implementer. For commands, this should include command line options and keywords and answers to any queries asked by the program. Libraries should have indicated their proper position in loading sequences. 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)

Run-time 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.

Even more space is saved during execution by multiple users if the program uses shared code.

Program Memory Images Saved by the Loader: Installation in the command UFD is extremely simple. The runtime version of the program is copied into UFD CMDNC0 using PRIMOS' FUTIL file handling utility. For example, assume you have written a utility program called FARLEY. This utility acts as a "tickler" for dates. Using FARLEY, each user builds a file with important dates. The FARLEY utility program, upon request, prints out upcoming events or occasions of interest to the user.

Note

> This utility does not exist; it is used as a plausible example.

First, compile the program

```
OK, FTN FARLEY -64R                    Compile in 64R mode
0000 ERRORS [<.MAIN.>FTN-REV17.3]      Compiler message
OK, LOAD                               Invoke the Loader
$LO B_FARLEY                           Load the object file; the
```

	default name is used
$	Load other required modules
.	
.	
.	
$LI	Load the FORTRAN library
LOAD COMPLETE	Load is complete
$SA *FARLEY	Save the memory image
$QU	Return to PRIMOS
OK, FUTIL	Invoke the file utility
>TO CMDNC0 owner-password	Defines the TO UFD as CMDNC0
>COPY *FARLEY FARLEY	Copies the runtime program *FARLEY into UFD CMDNC0 under the name of FARLEY
>QUIT	Return to PRIMOS Command level
OK,	

It was not necessary to define a FROM UFD; the default (home) was used.

Any user can now invoke this program:

OK, FARLEY	Invoke program
TILL WHAT DATE:	Asks for future time period
etc.	

Segmented Runfiles Saved by SEG's Loader: A segmented program cannot be run directly from UFD CMDNC0 because PRIMOS' command processor cannot directly handle the SEG runfiles. The segmented program may be invoked by means of a non-segmented 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 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 FUTIL's COPY command. The file name should be that by which the program will be invoked.

Examples

1. Extensions to the FARLEY utility described above make it desirable to compile and load it as a segmented program:

```
OK, FTN FARLEY -64V        Compile in 64V mode
0000 ERRORS [<.MAIN.>FTN-REV17.3]

OK, SEG                    Invoke SEG utility
[SEG rev 17.1]
# LOAD #FARLEY             Establish runfile name
$ LO B_FARLEY              Load object file
$ .                        Load any user subroutines
  .
  .                        Load any libraries needed
$ LI                       Load  FORTRAN library
LOAD COMPLETE
$ SA                       Save the file
$ QU                       Return to PRIMOS
OK,
```

2. The command file SEG>CMDSEG creates the interlude program:

```
OK, * CMDSEG, SEG, CEH-LSS-KJC 08/02/79
OK, * COMMAND FILE TO CREATE 'CMDNC0' SEG RUNFILES
OK, R SEG>*CMDMA
RUN FILE NAME: FARLEY

OK, FTN S$$SEG 1/5707
0000 ERRORS [<.MAIN.>FTN-REV17.3E]

OK, FILMEM

OK, LOAD
$ SZ
$ ER 2
$ MO D64R
$ SY BUFTBL 150000
$ CO 172400
$ LO B_S$$SEG 172400
$ AU 2
$ LO SEG>CMDLIB * 15 0 21 20 0 2 5 0 177777
MORE :3 0 0 0 0 2
$ AU 0
$ SE * 6
$ LI APPLIB
$ LO LIB>FTNLIB * 6 11 2 0 0 0 177777
MORE :0 0 0 0 0 0 177777
MORE :0 0 0 0 0 1
LOAD COMPLETE
$ MA 3
```

```
$ MA 2
*START  172400   *LOW   172364   *HIGH  177746   *PBRK  177747
*CMLOW  166627  *CMHGH  172377   *SYM   000112   *UII   000043

*BASE   000200  000200  000777  000777
*BASE   172543  172560  172557  172557
*BASE   173445  173465  173464  173465
*BASE   174323  174343  174342  174342
*BASE   174540  174542  174541  174541
*BASE   174544  174546  174545  174545
*BASE   175032  175037  175036  175036
*BASE   175110  175112  175112  175112
*BASE   175152  175152  175153  175153
*BASE   175344  175346  175345  175345
*BASE   175417  175421  175424  175424
*BASE   176004  176012  176011  176011
*BASE   176644  176655  176654  176654
*BASE   177022  177024  177023  177023
*BASE   177603  177603  177603  177603

$ SAVE *TEST
$ QU

OK, DELETE S$$SEG
OK, DELETE B_S$$SEG
OK, CO TTY
```

3. UFD SEG contains the SEG runfiles which are actually executed by the interlude programs. The SEG runfile is copied here from the UFD in which it was SAVEd.

```
OK, FUTIL                     Invoke FUTIL
>TO SEG
>TRECPY #FARLEY FARLEY        Make a copy under the invocation
>                             name
```

There is no FROM UFD defined, as the default (home) is being used.

4. The interlude program *TEST is copied into the Command UFD under the name by which it will be invoked.

```
OK, FUTIL
[FUTIL rev 17.2]
>TO CMDNC0 owner-password     TO UFD is CMDNC0
>COPY *TEST FARLEY            Copy the interlude
>QUIT                         Return to PRIMOS command level

OK,
```

When FARLEY is entered at the user terminal, the FARLEY interlude program in CMDNC0 is executed. This program attaches to the SEG UFD, restores the segmented runfile FARLEY, re-attaches 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.

MAG TAPE ASSIGNMENT FOR THE SYSTEM OPERATOR

The extended ASSIGN command for magnetic tapes provides for better utilization of magnetic tape resources. It allows the user to request operator assistance in assigning a drive or mounting a tape. The operator may assign drives based upon user-supplied attributes.

ASSIGN Process

Each mag tape operation requires the use of at least one tape drive. Tape drives are reserved with the PRIMOS 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 user-number and device number correlation exists internally, the user has exclusive access to the drive. Access privileges are relinquished with the UNASSIGN command. (Physical device numbers are assigned at system start-up.)

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. (See below.)

ASSIGN Command Format

The format of the ASSIGN command is:

```
ASSIGN { MTpdn  [-ALIAS MTldn] } [-options]
       { MTX    -ALIAS MTldn   }
```

The arguments and options are:

Argument	Description
MTpdn	Mag tape (MT) unit number from Ø to 7, inclusive. pdn is the physical device number assigned to each drive at system startup. Numbers can be obtained from the system operator.
MTX	Tells the operator to assign "any available drive"; MUST be accompanied by -ALIAS MTldn which assigns a number (alias) to the drive for reference purposes. See below. The actual drive assigned depends on any other options which appear on the command line.
-ALIAS MTldn	The logical drive number, from Ø to 7, inclusive. ldn is a user-specified number. This feature is particularly handy for use in programs that run from command files, as 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). Logical and physical device numbers can be used interchangeably in MAGNET, MAGSAV and MAGRST dialogs; however, to avoid confusion, give MAGRST/MAGSAV the logical device number, if you're using aliases.

Option	Description
-WAIT	Indicates user is willing to wait until requested drive is available.

-TPID id	Requests the operator to mount a particular reel of tape, identified by a tape id; requires operator intervention. id is a list of tape identifiers (arguments) 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	Protection rights may be specified by:
-RINGOFF	

RINGON or	Read and write permitted.
RINGOFF	Read only; write-protection in effect.

Requires operator intervention for removal or replacement of write-ring.

	Particular tape density settings are requested
-800BPI	with these options. Most drives can handle 800
-1600BPI	and 1600 bpi settings. Requires operator
-6250BPI	intervention.
-7TRK	Indicates 7- or 9-track tape drive;
-9TRK	default is 9-track. Requires operation intervention if -7TRK is specified.

Setting the Mode of Assignment

The System Operator can choose from three modes of tape drive assignment:

- Default mode: Users can assign tape drives without operator assistance unless requested for special favors.
- 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 establishes the assignment mode and can only be issued from the supervisor terminal. If SETMOD is not issued, the default mode prevails. (See USER option below.)

SETMOD has three arguments, each of which corresponds to one of the modes described earlier:

```
         ( -USER     )
SETMOD  {  -OPERATOR  }
         ( -NOASSIGN )
```

-USER mode is the default assignment mode. It permits user-assignment of tape drives by physical device number (pdn) alone, or by pdn and the "-ALIAS ldn" option. All other options (e.g., MTX, -RINGON, etc.) require operator intervention. Only commands which require operator assistance are displayed on the supervisor terminal.

-OPERATOR mode requires operator intervention in all tape drive assignment operations. All user-issued tape ASSIGN commands are displayed, with user-numbers, at the supervisor terminal. The operator answers each ASSIGN request with the REPLY command, discussed below. REPLY sends an informative message to the user terminal from which the command originated.

-NOASSIGN 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.  (As)
ER!
```

In environments which restrict user access to tape drives, this informs users that the operator is not available for request-handling.

Messages to the Operator

Requests for operator assistance are displayed in the following form at the supervisor terminal:

```
***** MAGTAPE REQUEST *****
From user-name (userno):  command-line
```

user-name and userno identify the originator of the request. The command-line is the actual ASSIGN order issued by the user.

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

Replying to Requests

The operator must perform certain requests before the user can proceed with a mag tape operation. When the requested task has been accomplished, the user should be informed. The REPLY command is the operator's method of communicating with each user terminal. It allows the operator to:

- Inform a user that a special request has been fulfilled.
- Deny a request.
- Approve a simple request (in OPERATOR mode).
- Display the actual pdn assigned when MTX option is specified.
- Request repetition of an ASSIGN message.

The REPLY command format is:

$$\text{REPLY -userno -TAPE}\left\{\begin{array}{l}\text{ABORT}\\ \text{GO}\\ \text{pdn}\\ \text{RESEND}\end{array}\right\}$$

The command must include the user number (-userno) and the -TAPE argument, or an error message will result. Only one of these listed options can be specified at a time:

Options	Description
ABORT	The operator is unable to assign the requested drive, for whatever reason: no drive available, tape not found, etc.
GO	The operator okays 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 which did NOT specify the MTX -ALIAS ldn option.
pdn	The pdn option is used in all cases where a user specifies the MTX -ALIAS option. The operator selects a suitable drive, performs any special requests, then uses this option to send the drive's physical device number (pdn) to the user's terminal. Then, the message: "Device MTpdn Assigned" is displayed.
RESEND	The most recently sent assignment request is repeated at the supervisor terminal.

Improper REPLY Format: If the operator issues an improper form of REPLY, an informative error message is displayed. For example, an incorrect command and typical response are:

```
REPLY -7 -TAPE MT1
"MT1" not implemented or improper use of argument. (REPLY)
Usage: REPLY -usrnum -TAPE [RESEND | ABORT | GO | pdn]
       REPLY -TAPE RESEND
       REPLY [ -usrnum | -ALL ] [ -RESEND ]
```

Repetition of Requests

All outstanding (unanswered) requests may be repeated by using the -ALL option of REPLY. The operator can display just the most recent, or all, of the outstanding requests made by a specific user:

```
REPLY  [-userno] [-RESEND]
       [-ALL   ]
```

-userno is the number of the user whose requests the supervisor wants to see again. -ALL repeats all the outstanding requests. -RESEND repeats only the last assignment request issued.

Altering Repeat Frequency

The default message repeat-frequency is 180 seconds, or 3 minutes. This number can be altered with the -REPEAT option of REPLY. The format is:

```
REPLY -REPEAT seconds
```

The seconds parameter is a decimal number.

Device-User Correspondence

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  USRNAM  USRNUM  LDEVICE
MT1     DOUROS    7     MT0
```

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 mag tape drives, card readers, punches, etc.:

```
OK, STAT US
```

```
USER    NO LIN PDEVS
SYSTEM   1 76  660
DOUROS   7  5  10660 MT1
SYSTEM  23 77  660 (2)
SYSTEM  24 77  10660 PR0      (0)
```

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 UNASSIGN (release) any tape drive "owned" by any user. The operator must use the pdn option to unassign a user-held device; the "-ALIAS ldn" option can only be used by the operator when the drive released was actually assigned with an ALIAS from the supervisor terminal.

This restriction stems from the internal correlation of pdn-to-ldn-to-user-number. STAT DEV clearly shows this correlation, as displayed above. 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:

Userno	Physical Device Number	Logical Device Number
17	MT1	MT2
17	MT2(formerly MTX)	MT0

Note that this represention is similar to the table displayed by STAT DEV. It should be noted that every mag tape drive has a default logical device number. It is the same as the drive's pdn, unless changed with the ALIAS option.

User 17 can unassign physical drive MT1 with these commands:

```
UN MT1

   or

UN -ALIAS MT2
```

Similarly, user 17 can release physical drive MT2 with:

UN MT2

or

UN -ALIAS MT0

The operator could release these drives with these commands:

UN MT1

and

UN MT2

In other words, only the user number associated with the <u>pdn</u> - <u>ldn</u> correlation can successfully issue the -ALIAS option when <u>unassigning</u> the drive.

No message other than "OK," is printed at the terminal when a successful UNASSIGN operation is completed.

<u>Examples of Assignment</u>

The messages displayed at the supervisor terminal depend on the current assignment mode established by SETMOD. Below are a few examples of user requests and corresponding operator replies in each of the three modes:

<u>Default Mode</u>: In default mode, (if SETMOD-USER has been issued or if the SETMOD command has not been used), only special ASSIGN requests will appear at the supervisor terminal. For example, if a certain user types:

AS MT1 -TPID POWER -6250

then the operator sees:

```
***** MAGTAPE REQUEST *****
From DOUROS (7) :  MT1 -TPID POWER -6250
```

To acknowledge the request, the operator should mount the POWER tape on MT1 and set the density switch to 6250 bpi. Then the operator should issue the message:

REPLY -7 -TAPE GO

To deny the request, the operator would respond:

REPLY -7 -TAPE ABORT

<u>MTX Request</u>: If a user specifies the MTX -ALIAS option as in:

AS MTX -ALIAS MT0 -TPID POWER -1600

then the operator receives the message:

***** MAGTAPE REQUEST *****
From DOUROS (7) : MTX -ALIAS MT0 -TPID POWER -1600

The operator mounts the POWER tape on an available drive and sets the density to 1600 bpi. If the operator chooses MT1 as the available drive, the following response would be issued:

REPLY -7 -TAPE 1

User number 7 then sees: Device MT1 assigned.

Request for RESEND: To repeat the latest mag tape request, the operator types:

OK, REPLY -TAPE RESEND

OK,

***** MAGTAPE REQUEST *****
From DOUROS (7) : MTX -ALIAS MT0 -RINGOFF

To display all unanswered requests, the operator types:

REPLY -ALL

OK,

***** MAGTAPE REQUEST *****
From DOUROS (7) : MTX -ALIAS MT0 -RINGOFF

In this case, only one request was waiting to be answered.

OPERATOR Mode: All mag tape requests are seen by the operator in this mode. The user receives no acknowledgement of the ASSIGN request until the operator responds with REPLY. For example, if user number 7 types:

AS MT1 -ALIAS MT0

then the operator would receive the request:

***** MAGTAPE REQUEST *****
From DOUROS (7) : MT1 -ALIAS MT0

If physical drive MT1 is available, the operator would reply:

REPLY -7 -TAPE GO

In this mode, the operator is informed whenever the user unsassigns a device:

Device MTpdn Unassigned.

The pdn value depends on the actual device being released. However, if the operator UNASSIGNs a tape drive, the user is NOT informed.

NOASSIGN Mode: User assignment requests are NOT permitted in NOASSIGN mode. No message appears at the supervisor terminal when a user issues a mag tape ASSIGN command. The user, however, is informed that no mag tape assignment is permitted at this time.

SECTION 4

MONITORING THE SYSTEM

OVERVIEW

This section contains the following:

- A discussion of commands that give the operator information about system status.
- Messages which are printed at the supervisor terminal due to actions of users or of the system. These actions are not initiated by the operator.
- Use of the Event Logging mechanism in PRIMOS.
- Helpful hints on how to keep a system logbook: contents, format, and uses.

LOOKING AT THE SYSTEM

Introduction

The major tool available to monitor the condition of the system is the STATUS command. It prints information on the status of users, devices, and the network (if any) to which the local system is attached. A complete description of STATUS, its options, and details of the information it prints at the supervisor terminal is in Section 16. Certain other commands are available to monitor specific aspects of the system (spooler, Batch, etc.). They are described fully in Reference Guide, PRIMOS Commands. Brief descriptions as they pertain to system monitoring are given below.

Disk Space Utilization

The AVAIL command prints, for a specified disk or partition, the number of records used, number of records still available and percent of disk or partition full. (Records are given as "normalized" - i.e., 880-byte-records.) The format useful to the operator is:

$$\text{AVAIL} \begin{Bmatrix} \text{volume-name} \\ \text{logical-device-number} \\ * \end{Bmatrix}$$

<u>volume-name</u> The name of the disk or partition.

<u>logical-device-number</u> The logical-device-number spelled out in English (e.g., FOUR, SIX, etc.).

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

```
OK, AVAIL SYSENC
   VOLUME SYSENC
    34476 TOTAL RECORDS (NORMALIZED)
     4126 RECORDS AVAILABLE (NORMALIZED)
    88.0% FULL


OK, AVAIL ZERO
   VOLUME SYSENC
    34476 TOTAL RECORDS (NORMALIZED)
     4126 RECORDS AVAILABLE (NORMALIZED)
    88.0% FULL


OK,
```

If the command AVAIL * is given, a table of record utilization on all started up devices will be printed. Example:

OK, avail *

VOLUME ID	TOTAL RECS	FREE RECS	% FULL	COMMENTS	
SOFTWR	241333	8676	96.4	3462	ENB
SPOOLB	34476	24587	28.7	460	ENB
MISCEL	86190	9630	88.8	71063	ENB
DBTEST	86190	18376	78.7	71061	ENB
SYSENC	34476	6255	81.9	460	ENC
SYSEND	34476	4952	85.6	460	END
TRANS	137904	6635	95.2	12060	END
TRANS2	155142	4212	97.3	52061	END
DTEST	137904	33705	75.6	2062	END
TEST	120666	11627	90.4	61463	END
SYSENE	34476	5359	84.5	460	ENE
CPUGR1	172381	4703	97.3	12460	ENE
CPUGR2	120666	32940	72.7	61461	ENE
MFGTFR	34476	3446	90.0	462	ENE
SYSENF	34476	13463	60.9	460	ENF
PERIPH	155142	46012	70.3	52061	ENF
MISCE2	137904	114108	17.3	12060	ENF
SYSENG	34476	10866	68.5	460	ENG
SUBSYS	68952	5876	91.5	11060	ENG
DBGRP	120666	15641	87.0	61461	ENG
SPOOLA	34476	15918	53.8	460	ENA
OSA	103428	3516	96.6	21460	ENA
TPUBS	86190	10579	87.7	71061	ENA

```
M168A1      34476       7542   78.1     660  ENA
M168B1      34476       8987   73.9   10660  ENA
M168S1      34476      12446   63.9   20660  ENA
M172A1      34476       7370   78.6   60660  ENA
M172B1      34476       9127   73.5   70660  ENA
M172B2      34476      24531   28.8   50660  ENA
M172S1      34476      23461   31.9  100660  ENA
SPOOL       17238      16595    3.7  110061  TE.CAD
RESRCH      68952       7002   89.8    1060  RES.C1


OK,
```

Note

AVAIL requires non-owner MFD passwords to be XXXXXX and the packname protection set to 7 1 (i.e., owner all rights; nonowner read rights only).

Building the DISCS File: AVAIL * will not work unless the file DISCS has been built in the UFD SYSTEM. The DISCS file is created with the editor. It contains three columns:

```
VOL-ID    PHYSICAL-DISC    COMMENTS
```

The VOL-ID column contains the name of each disc. The PHYSICAL-DISC column contains the disc's number. The COMMENTS column identifies the node on which the disc is mounted. 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, a system
OK, slist discs
SOFTWR     3462 ENB
SPOOLB      460 ENB
MISCEL    71063 ENB
DBTEST    71061 ENB
SYSENC      460 ENC
SYSEND      460 END
TRANS     12060 END
TRANS2    52061 END
DTEST      2062 END
TEST      61463 END
SYSENE      460 ENE
CPUGR1    12460 ENE
CPUGR2    61461 ENE
MFGTFR      462 ENE
SYSENF      460 ENF
PERIPH    52061 ENF
MISCE2    12060 ENF
SYSENG      460 ENG
SUBSYS    11060 ENG
```

```
DBGRP      61461 ENG
SPOOLA       460 ENA
OSA        21460 ENA
TPUBS      71061 ENA
M168A1       660 ENA
M168B1     10660 ENA
M168S1     20660 ENA
M172A1     60660 ENA
M172B1     70660 ENA
M172B2     50660 ENA
M172S1    100660 ENA
SPOOL     110061 TE.CAD
RESRCH      1060 RES.C1

OK,
```

Spool Queue

The contents of the spool queue may be printed by the command:

SPOOL -LIST option

option	action
omitted	Lists all files.
ALL	Lists all files.
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 login name.
PLOT	Lists files in plot queue.
PRINT	Lists files in print queue.

Each listing contains the following information:

user name, spool file name, date/time of spool request, options, file size, file name, form (if any), defer time (if any). See section 8 for more details on the spooler.

Number of Users

The total number of system users (terminal, phantom, remote) can be obtained with the internal command USERS. The dialog is:

```
OK, USERS

USERS= 23
```

System Parameters

Below 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
Active Batch jobs	JOB -STATUS or JOB -DISPLAY
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 -DISPLAY
Batch jobs, executing	BATCH -DISPLAY
Batch jobs, for one user	JOB username -STATUS or -DISPLAY
Batch jobs, specific	JOB job-id -DISPLAY
Batch queue names	BATGEN -STATUS
Batch queue parameters	BATGEN -DISPLAY
Batch subsystem usage	BATCH -DISPLAY
Deferred spool files	SPOOL -LIST DEFER
Device, command	STATUS or STATUS DISKS
Device, paging	STATUS
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
Disks mounted	STATUS DISKS
Disks, remote	STATUS DISKS
Executing Batch jobs	BATCH -DISPLAY
File units in use	STATUS UNITS
Free records	AVAIL
Line, user	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 or STATUS USERS
Number, user	STATUS USERS
Packnames	STATUS DISKS
Paging device	STATUS
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
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, users 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	STATUS USERS
User logins to other nodes	STATUS USERS
User number	STATUS USERS
User physical devices	STATUS USERS
User priority	STATUS USERS
Users Batch jobs	JOB username -STATUS or -DISPLAY
Users own spool files	SPOOL -LIST OWN
Users, number of	STATUS USERS or USERS
Users, phantom	STATUS USERS
Users, remote	STATUS USERS
Volume names	STATUS DISKS

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.

Terminal (Local and Remote) Users

▶ UFD-name (user-number) LOGGED IN AT hh'mm mmddyy

A terminal user has logged in:

```
TEKMAN (35) LOGGED IN AT 12'17 040378
```

▶ ufd-name (user-number) LOGGED OUT AT hh'mm mmddyy
TIME USED = hh'mm mm'ss mm'ss
TIMED OUT (prints if PRIMOS logs out inactive user)

A terminal user has logged out. Time used is printed.

Times used are: connect time, CPU time, disk I/O time:

```
TEKMAN (35) LOGGED OUT AT 13'43 040378
TIME USED = 1'26 67'20 0'42
```

Phantom Users

▶ USER user-number: PHANTOM TTY REQUEST

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

```
USER 35: PHANTOM TTY REQUEST
```

▶ PHANTOM phantom-user-number: error-text

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

```
PHANTOM 52: BAD PARAMETER
```

Batch

The Batch messages that appear at the supervisor terminal are explained in Section 9. They are easily identified, since they all begin with *BATCH*.

▶ ***56 12'45
BATCH Executing job o_test for user JONES (#20056)

A full list of Batch messages is given in Appendix F.

Disk Errors

Under PRIMOS, disk error messages are of the form:

    DISK xx ER - other information.

xx is RD for a read error and WT for a write error

Under PRIMOS II, disk error messages are of the form:

    DISK xx ERROR - other information.

xx is RD for a read error and WT for a write error.

See Appendix E for a full explanation of the other information.

Networks

    **** hh'mm node-name   FAM<171F25 > NOT OPERATIONAL**

The FAM on system node-name on the network has shut down:

    **** 15'17 SYSD   FAM<171F25 > NOT OPERATIONAL**

    **** hh'mm node-name   FAM<171F25 > INITIALIZED**
    **** hh'mm node-name   FAM<171F25 > OPERATIONAL**

The FAM on system node-name on the network has been brought up and is functional:

    **** 15'30 SYSD   FAM<171F25 > INITIALIZED**
    **** 15'31 SYSD   FAM<171F25 > OPERATIONAL**

    **** hh'mm node-name   DISK device-number SHUTDN**

A remote device at system node-name on the network has been shutdown at the system where it is physically mounted. This message prints only if device-number has been declared a remote device on the local node by a STARTUP node-name or ADDISK node-names command:

    **** 15'46 SYSD   DISK 2062 SHUTDN**

EVENT LOGGING IN PRIMOS

Overview

An event logging mechanism is provided as part of the operating system to record information about significant 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.

The following paragraphs describe how the logging mechanism works, necessary operator actions, and steps for the programmer or administrator to take to modify the mechanism.

Interrogating the Event Logging File

LOGPRT is a program that writes the contents of the event logging file (LOGREC) to a disk file or a user terminal. The command line to invoke LOGPRT under PRIMOS is as follows.

```
R 'SYSTEM owner-password>*LOGPRT' [destination] [options]
```

destination — The pathname of the destination for LOGPRT's output. If TTY is specified, the output will be to the user's terminal. If destination is omitted, output will be to the file LOGLST in the current UFD. Any other name will be considered a pathname to which the output is directed.

options — The values and meaning of the options are as follows:

Option	Meaning
-HELP	Prints a list of LOGPRT options. The LOGPRT command must be reissued after the list of options is printed.
-SPOOL	Automatically spools the resulting output file. LOGPRT prints the name of spool file and indicates whether the spool file is long or short. This option is ignored if TTY is specified for destination.
-DELETE	deletes the output file (after spooling). This option is ignored if TTY is specified for destination.
-PURGE	Empties LOGREC after LOGPRT has finished processing. The default is to leave LOGREC unmodified. Requires owner rights on LOGREC.

-INPUT [pathname]	Specify the pathname of the LOGREC file to be processed as pathname. Default is <0>CMDNC0>LOGREC.
-FROM [mmddyy / TODAY] [hhmm]	Only LOGREC entries from the specified date mmddyy or today's date to the latest entry are printed.
-TYPE t1 t2...tn	Process entries only of the indicated type. The types, t1, t2...tn can be any of the following:

No.	Type	Meaning
0	COLD	Cold starts
1	WARM	Warm starts
2	TIMDAT	Time/date entries (see Note)
3	CHECKS	Machine checks (including memory parity).
4	DISK	Disk errors
5	OVERFL	Record LOGREC overflow entries
6	SHUTDN	Operator shutdowns
7	CHK300	Prime 300 machine checks
8	PAR300	Prime 300 memory parity checks
9	MOD300	Prime 300 missing memory module checks
10-15	TYPE10...TYPE15	Entries for types 10 to 15
16	DSKNAM	Either ADDISK or STARTUP entries
17	POWERF	Power fail checks
20	REMARK	Operator message

Note

The time/date stamps associated with the selected entries will not be processed unless TIMDAT is explicitly selected. For example: -TYPE DISK TIMDAT will process all disk errors and their associated time/date stamps. If TIMDAT alone is specified 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.

-CONTIN Continue LOGPRT after encountering an invalid entry. LOGPRT normally halts if an invalid entry is encountered in LOGREC. When this option is specified, LOGPRT continue processing and attempts to find the next valid entry.

-DBUG Causes LOGPRT to read entries from the terminal. These entries can be used for testing LOGPRT formatting for entry types.

Each entry must be entered as a series of tokens (using the rules described for the RDTK$$ subroutine; see the Subroutine Reference Guide). Octal tokens are converted to binary. All others are interpreted as ASCII strings and truncated to the leftmost two characters.

LOGPRT leaves the -DBUG mode of operation when a token starting with a hyphen (-) is entered.

The -DBUG option also enables the TTY output and -CONTIN option.

-CENSUS Prints number of each event type processed. Only selected types are counted and only non-zero 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.

-REMARK Enter comment directly into the LOGREC 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 LOGREC. The text may be up to 160 characters and need not be surrounded by apostrophes. Write access is required.

-DUMP Dumps each entry in octal.

Under PRIMOS II, the operation of LOGPRT is identical, with the exception of the -SPOOL option which is not supported by PRIMOS II.

LOGPRT Messages

If LOGPRT finds that the output file already exists, it prints:

OK TO DELETE OLD destination? ANSWER: 'Y' OR 'N'!

The reply should be Y to delete the file or N to enter a new destination pathname. 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 LOGREC unmodified.

EVENT MESSAGE FORMAT

Overview

LOGPRT writes a header line followed by formatted entries, one or more lines per entry. (All numbers are octal except where noted.)

The specific format for currently defined event types, with a brief explanation of the format, is given below. The messages are listed by approximate frequency of occurrence.

Information Printed by LOGPRT

The first line of an entry is a header line containing the pathname of the input file and the system time and date in the format:

hh:mm:ss day dd xxx yyyy

This gives the date/time record entered by LOGEV2 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 shown happened during the minute immediately before the time shown. An example of a date/time record is:

***** <Ø>CMDNCØ>LOGREC, Ø9:23:44 TUE 12 DEC 1978 *****

The header is followed by formatted entries, one or more lines per entry. The following entries define specific type of events which may be listed after a date/time entry. See Appendix ELGR for a list of LOGPRT messages.

NETWORK EVENT LOGGING

Overview

A network event logging mechanism is provided to record information about significant network events in an internal buffer, periodically dump this buffer to a disk file, and format and print the disk recording file.

Interrogating the Network Event Logging File

NETPRT is a program that writes the contents of the network event logging file (NETREC) to a disk file or a user terminal.

NETPRT is invoked by:

R 'SYSTEM owner-password>*NETPRT' [destination] [-options]

or its equivalent sequence:

A SYSTEM owner-password
R *NETPRT [destination] [options]

destination — The destination for NETPRT's output. If TTY is specified, the output will be to the terminal. If destination is omitted, output will be to the file NETLST in the current UFD. Any other specification will be taken as a filename to which the output will be directed.

option — An option keyword, possibly followed by subfields. All option keywords begin with a hyphen.

-HELP — A list of NETPRT options is printed. The NETPRT command must be retyped after the options are printed.

-INPUT name — Specify pathname of NETREC file to process. If this option is omitted, a prompt is issued for the pathname.

-FROM mmddyy — Only NETREC entries from the specified date to the latest entry are processed.

-TYPE t1 t2 ... — Process entries only of the indicated types. The types (t1, t2, etc.) can be any of the following:

COLD	Cold starts
WARM	Warm starts
TIMDAT	Time/date entries
RESET	Circuit resets
BADSEQ	Packets out of sequence
OVERFL	NETREC 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 protocols down
PWFAIL	Power fail checks
INCREQ	Incoming call requests (for FAM debugging)
OUCREQ	Outgoing call requests (for FAM debugging)
REMARK	Operator remark

Note

The time/date stamps associated with the selected entries will not be processed unless TIMDAT is explicitly selected. If TIMDAT alone is specified, all time/date stamps in NETREC will be processed. If TIMDAT is specified in conjunction with one or more other types, only the time/dates of the selected types will be processed. If the -TYPE option is not specified, all entries will be processed.

-SPOOL	Spool the output file when done. NETPRT will print the name of the output spool file and a long/short indication.
-DELETE	Delete the output file when done (meaningful only when using the -SPOOL option).
-PURGE	Empty NETREC when done. Owner rights are required on NETREC.

NETPRT Messages

If NETPRT finds that the output file already exists, it will print the message:

OK TO DELETE OLD destination? ANSWER: 'Y' OR 'N'!

The reply should be Y to delete the file or N to enter a new destination. If N is entered, the message:

NEW OUTPUT TREENAME:

is printed.

Finally, if no -INPUT option was specified, NETPRT prints the message:

INPUT TREENAME:

The pathname of the NETREC file to be printed should be entered. If a null line is entered, <0>CMDNC0>NETREC will be assumed.

NETWORK EVENT MESSAGE FORMAT

NETPRT first writes a header line containing the name of the input file. For example:

    ***** NETREC LISTING *****

The header is followed by formatted entries, one or more lines per entry. The following entries are currently defined. All numbers are in octal except where noted:

    09:01:20 WED 16 FEB 1979

This is a date/time record entered by NETEV2 when NETBUF was dumped to NETREC. All events following this entry and before the next date/time entry occurred during the thirty seconds just prior to the time shown.

Purging of NETREC

The size of NETREC is not limited; the file should be purged periodically.

THE SYSTEM LOGBOOK

As a matter of good operating procedure, the System Administrator should establish a logbook for recording system status and operation. The Administrator should see, that as part of their training, operators are instructed in the details of maintaining a logbook.

Purpose

A logbook should contain sufficient information about system operation and history to allow unusual or undesirable occurrences to be analyzed. Logbook records should be adequately detailed so that the Administrator or operator can restore all or part of the system to normal status. Unusual operational events include: system crashes, hardware changes, operator errors, or external happenings such as power failure.

Format

How logbooks should be kept and what information should be recorded is at the discretion of the Administrator. However, we would like to pass on some of the experience accumulated by Prime's system operators. Certain standards will facilitate logbook usage; the following procedures are recommended:

- Logbooks should be bound. Looseleaf pages are easily detached and lost. This is especially true of pages frequently referred to. Log books should also be numbered.
- The logbook should stay open and flat for ease of writing.
- Exact page size is not important. However, select a convenient size that will allow printouts and listings to be pasted in.
- All logbook entries should be labelled with time and date. This gives a historical record, useful in reconstructing how a crash or other unusual event occurred. It also enables these events to be correlated with external events (power failure, etc.).
- Logbook entries should be signed (or initialed) by the person making the entry. The Administrator (or an analyst) then knows whom to ask for more information about a specific event.
- All logbook entries should be made in ink, not pencil. Incorrect entries should be crossed out neatly.

Contents

The following types of information and events should be recorded in the system logbook.

Hardware information:

- Equipment configuration - disks, peripherals, etc. (including model number and serial number)
- Changes to equipment configuration: additions, removals, replacements, alterations

- Changes in operating status of equipment (component failures, etc.)

Software information:

- Listing of the startup command file C_PRMO (listing of other command files, if alternative configurations are used)

- Copy of the CONFIG directive or CONFIG data file

- Listing of the loadmap, M_PRMO, for each version of PRIMOS used on the system

- Allocation of shared segments, both Prime-supplied and user-installed

- Contents of the command directory, CMDNC0, and the library directory, LIB

Any changes - additions, deletions, replacements - in the above, should be entered in the logbook.

Operations information:

- System startup. Include any special conditions: subsystems (SPOOL, BATCH, FAM, etc.) not brought up

- Any FIXRATs - disk partition, result

- Disk formatting - new partition names created

- Backups performed - partitions copied: to tape or to disk

- Files restored to the system

- Information on archived files; i.e., filenames, where and how archived

- Time, date, and place of storage, of Event Logger Dumps

- New UFDs created or names changed

- Replacement or other changes of commands in CMDNC0, or libraries in LIB

- Shutdowns - total or partial

- Reason for shutdowns; i.e., environmental factors, plant shutdown, configuration change, system update, etc.

In all the above, any anomalous conditions or responses should be noted.

Halts: See Section 10 for complete details on information to be recorded.

- System status - register values
- WARM or COLD start - did system continue to function properly immediately after start? after a half-hour?, etc.

Environment information:

- Abnormal temperature/humidity conditions
- Other adverse conditions - dust, chemicals, etc.
- Unauthorized access to computer area
- Any theft or damage to computer or components
- Unauthorized computer usage
- Any other unusual events
- Steps taken to correct any unusual environmental conditions

SECTION 5

FIXRAT

INTRODUCTION

The external command FIXRAT resumes a maintenance program that checks the PRIMOS file integrity on any disk pack. FIXRAT fully supports nested UFDs and nested segment directories. FIXRAT may be run from a command file. This section assumes the reader is familiar with the file structure described in the Subroutine Reference Guide. This guide should also be consulted for a description of segment directories and nested directories.

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 availablity table (RAT) from the existing file structure and compares it to the DSKRAT file for agreement. If discrepancies are found, FIXRAT prints an error message.

Note

> The packname is the name of the file containing the disk record availablity 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

Figure 5-1 shows a sample PRIMOS 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.

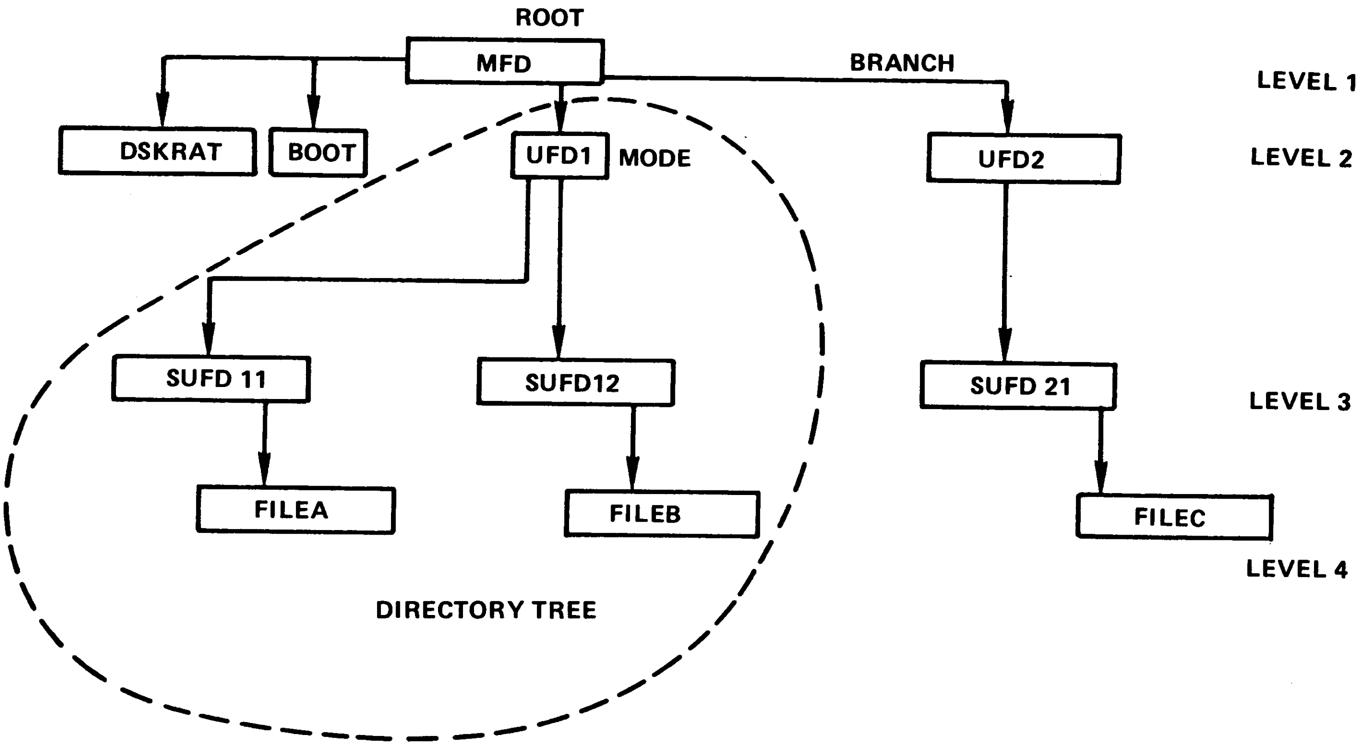
A directory tree consists of all files and sub-directories that have their root in that directory. In Figure 5-1, the directory tree for UFD1 is circled. The level of a file is the depth of that file in the tree. For example, as shown in Figure 5-1, the MFD is at level 1 in the tree, UFD1 is at level 2 in the tree, and FILEC is at level 4.

FIXRAT traverses the file structure, as shown in Figure 5-2, 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. Unless OPTIONS is specified, FIXRAT processing of the tree shown in Figure 5-1 generates the following default output:

```
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,
```



FIXRAT terminal output for the above file structure would be:

```
DISK PACK ID IS DSKRAT

BEGIN MFD
 BEGIN UFD1
  BEGIN SUFD11
  END   SUFD11      10
  BEGIN SUFD12
  END   SUFD12      10
 END   UFD1        21
 BEGIN UFD2
  BEGIN  SUFD21
  END   SUFD21      10
 END   UFD2        11
END   MFD         35
RECORDS USED(DECIMAL)=       35
RECORDS LEFT=              6223
DSKRAT OK
OK,
```

Figure 5-1. Sample File Structure

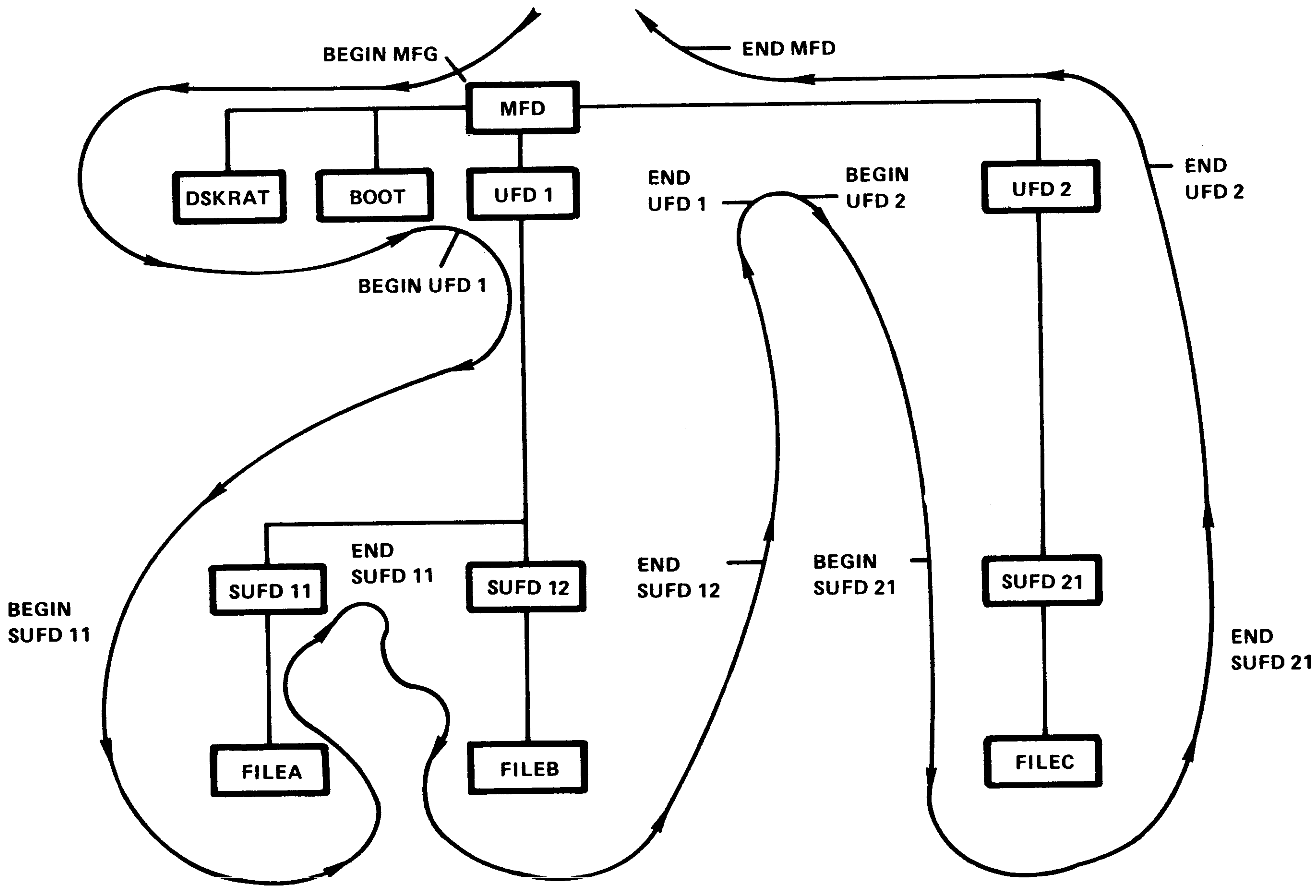


Figure 5-2. Typical FIXRAT Traverse of File Structure

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 printout option 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

We recommend the answer be NO for the first time. If any problems occur, first make sure these problems are not caused by hardware or the operating system. (This also shows the operator what information, i.e., 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 backed-up on the disk, then do not run FIXRAT and answer YES. Consult a senior programmer or Prime field analyst.

If the answer is NO, FIXRAT will ask:

UFD COMPRESSION ?

YES causes FIXRAT to compress UFDs, eliminating entries flagged as being 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.

In all cases, 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.

If the disk is partitioned, disk numbers include head-offset and number-of-heads information. 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

If the command was invoked as FIXRAT OPTIONS, three print option questions will be asked. If invoked as FIXRAT, the questions will not be asked and FIXRAT will use the default answers.

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 gives the option either truncating the UFD that is nested too deeply or leaving it as 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.

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 scattered on a disk. FIXRAT begins processing the disk after this point, as it would if the OPTIONS parameter was not specified.

If the utility was invoked as FIXRAT rather than FIXRAT OPTIONS, the three print option questions will not be asked and operations will proceed with the default answers of 2, NO, NO. That is, information will be printed about the MFD and directories in the MFD and no filenames and no file chains will be printed.

Following the file structure analysis printout, 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.

FIXRAT Output Example

The following is sample 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 is the disk packname, the name of the DSKRAT file.

The next section of output concerns FIXRAT examining the file structure on the disk for consistency. This sample output 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 name 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 and types:

    DSKRAT FIXED

then exits to PRIMOS. Otherwise, FIXRAT asks the user: FIX DISK? If the user answers YES, the DSKRAT is repaired. 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 in the following format (all numeric values are octal except index level which is decimal):

```
reason for error

FILE = filename  TYPE= filetype

BRA= bra      FATHER= fra      INDEX LEVEL= index-level

BAD RECORD=  cra       TRACK=track  HEAD=head

DIRECTORY PATH= directory-path (or MFD if file is MFD)

FILE DELETED, FILE TRUNCATED or blank
```

filetype is the type of file:SAMFIL, DAMFIL, SAMSEG, DAMSEG, SAMUFD, or ILLEGAL.

bra is the beginning record address of the file.

fra is the record address of the father-directory.

index-level is current index level (0 except for DAM files).

cra is the current record address.

track is the cylinder number at which the error occurred. The outside rim of the disk is track 0. Track numbers increase inwards up to 822.

head is the surface of the disk. The bottom of the disk pack is head 0. (Storage modules have up to 19 heads).

The directory-path is the list of nested file directories needed to get from the MFD to the bad file. For example, if FILEC in Figure 5-2 was broken, the directory path would be MFD, UFD2, SUFD21. 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 file, 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 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 sets the entry to a null entry.

Segment Directories

A segment directory may contain references to files and other segment directories. The distinction between a UFD and a segment directory is that entries in a UFD are referenced by name and those in a segment directory by position in the directory. Each entry in a UFD consists of a disk address that is the beginning record address of the file, followed by a name (Refer to the Subroutine 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, Section 6). In checking integrity of the DSKRAT, the file BADSPT in the MFD is also examined by FIXRAT. The file BADSPT contains information about badspots on the disk, i.e., the record addresses of badspots on the disk. Disk records that have badspots are not available for file system use, and FIXRAT must take this factor into account when fixing a disk or checking file integrity.

The file BADSPT may not be present on the disk, or it may be empty. In both cases, FIXRAT performs no badspot handling.

The format of the file BADSPT in the MFD is described in Section 7. If the format of the file 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 complain if any file except the BOOT in the MFD contains record 0. The message given is:

BAD DISK ADDRESS BAD RECORD = Ø

If this occurs, RESTore the BOOT from a good MFD on another disk and 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= treename (or MFD)

Description of Bad Structure Messages

▶ FILE = MFD BAD RECORD = 1
DIRECTORY PATH = MFD
FIXRAT ABORTED

A 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 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 (identification). The error messages that may arise as a result of one of these entries being missing are:

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 each of these questions, the specified action asked in the message 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 first running FIXRAT. Records already 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 the description of an Unrecovered Read error in this section.

▶ 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 one of the abovenamed files. 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.

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 any bad record pointers, or contains inconsistent information. If the DSKRAT is BAD, FIXRAT reconstructs it, using parameters typed by the user in response to the following questions. If the user types CARRIAGE RETURN to any of the questions, default values are used. The questions are:

```
INPUT  DECIMAL  RECORD SIZE =
INPUT  DECIMAL  FILE SPACE  RECORD  COUNT  =
INPUT  DECIMAL  CYLINDERS  =
INPUT  DECIMAL  HEADS  =
```

The default values for the particular disk model are used by FIXRAT if the user does not specify them and type (CR) instead. FIXRAT prints the default or the specified values back to the user for verification then asks "OK"?. If the answer is YES, FIXRAT repairs the DSKRAT and continues, or else 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-consistant and cannot be reconciled. The entry for this file 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 section 6). Normally, a succession of many unrecovered read errors indicate 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. When FIXRAT is rerun as a result of the unrecovered read error recovery procedure described here, multiple errors of the form:

    2 FILES POINT TO SAME RECORD

may occur. These errors result because a badspot may affect up to eight records belonging to multiple files.

EXAMPLE OF FIXRAT

```
OK, FIXRAT OPTIONS
FIXRAT 16.4
FIX DISK?  YES
PHYSICAL DISK = 70460
TYPE DIRECTORIES TO LEVEL = 3
TYPE FILE NAMES?  YES
TYPE FILE CHAINS?  NO

DISK PACK ID IS DTNEW
BEGIN MFD
  DTNEW
  MFD
  BOOT
  CMDNC0
  BEGIN CMDNC0
    AVAIL
    SIZE
    FILVER
    PSD
    MAGSAV
    MAGRST
    SLIST
    FILMEM
    SAVER
    FUTIL
    FIXRAT
    HILOAD
    EDB
    MAKE
    TAP
    UPCASE
    TRAMLC
    COPY
    ED
    FTN
    SFRWLK
    LOAD
    PMA
    TA
    VPSD
    LS
    DL
    PSD120
    CMPF
    VLOAD
    PRUFD
  END   CMDNC0          229
  DOS
  BEGIN DOS
  END   DOS             1
  LIB
  BEGIN LIB
```

Partition of a storage module (type 6). The partition has 2 heads and a head-offset of 14 heads. The device is mounted on unit 0 and connected to controller address 27. See Appendix D for details.

```
    FTNLIB
    APPLIB
    UII
    B_PSD
    B_LOAD
  END   LIB              84
  SYSCOM
  BEGIN SYSCOM
    FILERR
    KEYCOM
    FILD.F
    OFFCOM
    FILD.P
    PARM.K
    KEYS.F
    KEYS.P
    A$KEYS
    ERRD.F.OLD
    ERRD.F
    ERRD.P.OLD
    ERRD.P
    DEFI
    CBLK.I
  END   SYSCOM           26
  MORRIS
  BEGIN MORRIS
    RDENDIR
    BEGIN RDENDIR
      QDOS64
      C_RDEN$$
      C_LOAD
      B_RDEN$$
      L_RDEN$$
      L_LOAD
      QMAP64
      TDOS64
      PRUFD
      *PRUFD
      L_PRUFD
      B_PRUFD
      M_PRUFD
      RDEN$$.IV
      RDEN$$.III
      *LDUT
      RDEN$$.CMP
      RDEN$$.TEST
      RDEN$$
    END   RDENDIR          88
  END   MORRIS           89
  MAMNT
  BEGIN MAMNT
    *TEST
    *PRUFD
    *TDUMP
```

```
      LSDIR
      BEGIN LSDIR
        CRCV$A.FTN
        LS.FTN
        SPAN$A.FTN
        IGCH$A.FTN
        INDX$A.FTN
        SBST$A.FTN
        ASSN$A.FTN
        COMP$A.FTN
        BRK$A.FTN
        APND$A.FTN
        CCAT$A.FTN
        CTTV$A.FTN
        XLAT$A.FTN
        SCHR$A.FTN
        DL.FTN
        CEQN$A.FTN
        STAR$A.FTN
        OLS.FTN
        SPL.FTN
        FPRT$A.FTN
        CVDT$A.FTN
        CFNM$A.FTN
        FTN.C
        MOST.FTN
        MORE.FTN
        B_MORE.FTN
        APND$A.B
        TLS
        TLS.MAP
      END   LSDIR            57
      SEGDIR
      BEGIN SEGDIR
         (      Ø)
         (      1)
         (      2)
         (      3)
         (      4)
         (     33)
      END   SEGDIR           15
    END   MAMNT            93
  END   MFD         525
  RECORDS USED(DECIMAL)=          525
  RECORDS LEFT=                 14298
  DSKRAT OK
```

SECTION 6

FORMATTING STORAGE DEVICES

This section describes the utilities for formatting both disks (and disk partitions) and magnetic tapes. The disk formatting utility is MAKE; the tape formatting and label writing utility is LABEL.

OVERVIEW OF MAKE

MAKE is the system utility for the creation and formatting of disks and disk partitions, including both user and paging disks. MAKE will create and structure any disk or storage module supported by PRIMOS. The disk or disk partition is formatted with the following PRIMOS features:

- Disk record availability table (user-specified name)
- MFD
- BOOT
- CMDNC0
- DOS
- BADSPT (if there are badspots on the disk)

The MAKE program writes the bootstrap (BOOT) into Record 0 of the disk.

MAKE may be run from a command file.

Preliminary

The operator must first construct the number of the physical disk (or partition of the disk) from the tables in Appendix D. This number tells the system the type of storage device, the unit on which it is mounted, and the offset and size (if a partition). This physical device number must be added to the table of assignable disks with the DISKS command at the supervisor terminal.

The operator logs in on a user terminal to a UFD on another physical device. The disk to be formatted by MAKE is assigned to this terminal by the ASSIGN DISKS command, using the same physical device number as for DISKS. To prevent accidental erasure of data on a disk because a physical device number was mistyped, the following procedure is recommended.

- 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. (Some disk drives may have switches labelled WRITE PROTECT.) The DISKS and ASSIGN DISKS commands are not given when MAKE is run under PRIMOS II.

The disk to be created must not have been connected to PRIMOS by the STARTUP command.

RUNNING MAKE

To run MAKE, type the command MAKE. The response is:

```
MAKE, REV 17.2
BUILDING NEW PARTITION.
PHYSICAL DISK:
```

The user must then type the number of the physical disk to be created. An impossible number causes the message ILLEGAL DEVICE NUMBER (MAKE) to be typed out and the request to be repeated. A complete guide to the construction of physical device numbers (including partitions) is given in Appendix D.

MAKE computes the number of records on the disk pack from the disk number. In case of ambiguity, MAKE asks resolving questions, answerable by YES or NO, such as: 40 MB STORAGE MOD?

When the ambiguity, if any, is resolved MAKE asks:

```
SPLIT DISK?
```

If part of the disk is to be used for paging (see Section 20), answer YES, otherwise NO. This feature permits the system to maintain bad disk track information on the paging partition as well as on the file system partition. If answered YES, MAKE then asks:

```
PAGING RECORDS (DECIMAL)
```

The user responds by typing the number of records to be used for paging. This number is determined from the number of users to be run under PRIMOS. (See Section 20 for details).

The intent of split disks is to allow paging partitions to have bad spots recorded in the badspot file. Consequently, most of the disk should be paging records; a maximum of 20 to 25 records for the file system partition is all that is needed for the badspot file.

Note

It is not recommended that partitions on storage modules be split. Instead, create a 1-head paging partition on the module.

If the user desires to make a split disk, the recommended procedure is to first run MAKE with the SPLIT DISK question answered NO, then rerun MAKE with the SPLIT DISK? question answered YES. The reason for this action is to write correct checksums on the entire disk initially, not just on the file system part of the disk. This action is also important if it is desired to copy the split disk at a later time using the COPY command.

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

```
DISK          FILE-RECORDS    PAGE-RECORDS(DECIMAL)
number        number          number
```

and asks:

```
PARAMETERS OK?
```

If the number is correct, type YES in response to the OK? query. If not, type NO, and MAKE will return to the PHYSICAL DISK: question. MAKE then asks the question:

```
PACK NAME?
```

Enter a valid filename which will be the name of the disk record availability file (DSKRAT). The pack name must be PAGING if the disk is being split in order to maintain bad track information on the paging partition. If you are making a paging partition which is not to be split, the name must not be PAGING.

Note

The name of the DSKRAT file (packname) may be changed later, for example, by the CNAME command.

For all devices except diskettes (floppy disks), MAKE asks:

```
BADSPOTS ON DISK?
```

Badspots are part of a disk pack that cannot hold data. On some disk packs, this information is recorded on the inside bottom cover in the form of a list of pairs of numbers. These pairs (track number, head number) represent parts of the disk determined by the disk manufacturer to be probable badspots. If the list exists, the user must type YES response to the BADSPOT question. MAKE then prints:

TRACK=

The user responds by typing the track (decimal) of the first badspot. MAKE then prints:

HEAD=

The user responds by typing the head (decimal) of the first badspot. This dialogue between MAKE and the user continues allowing the user to record the track and head of all badspots on a given disk. The TRACK, HEAD pair 0, 0 terminates input of badspots. MAKE then prints a list of the badspot HEAD and TRACK numbers at the terminal and asks:

PARAMETERS OK?

If they are OK, the user types YES. At this point, MAKE writes a file containing the badspot information in the MFD with the filename BADSPT. The file named BADSPT is used by the utility commands FIXRAT and COPY. MAKE also initializes the DSKRAT file to ensure that badspot records are not available for file system use. A NO answer returns to the TRACK= question.

Note

PRIMOS supports a maximum of 16 bad spots on paging partitions; the sum of the bad spots on the primary paging partition (PAGDEV) and the alternate paging partition (ALTDEV) must not exceed 16.

If there are no badspots on the disk, the user responds by typing NO to the query BADSPOTS ON DISK?, and no BADSPT file is created.

MAKE then asks:

VIRGIN DISK?
VERIFY DISK?

Virgin Disks

If the user answers YES to the first question, MAKE writes the records of the disk or partition. The first word of each record is set to the record address. This action also writes a valid hardware checksum for each record.

If the user answers NO, MAKE does not initialize the records but writes BOOT, DSKRAT file, and MFD and proceeds to verification. The records need not be initialized if all the records have been initialized by a previous run of MAKE. However, it is strongly recommended that the user answer YES to the VIRGIN DISK ? question at each invocation of MAKE.

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. During this process the current track number is displayed on the control panel lights. Upon completion MAKE prints the following message and begins disk initialization:

    FORMAT COMPLETED

Disk Initialization

MAKE creates and writes to the disk the bootstrap (BOOT), disk record availability table, and the MFD. If requested by the user, 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 is not virgin, this step is omitted by the user.

Verification

If the verification question was answered YES, MAKE reads every record in the file system part of the disk or partition to verify that each record can be read. If the answer was NO, MAKE is complete.

During the reading and writing of all records, MAKE displays the track number it is processing in the DATA lights. When done, MAKE prints:

    DISK CREATED

and returns to the operating system, which types OK,. If any of the badspots affected the current partition or platter, the message: LOST RECORDS is printed. If any disk write errors occur, MAKE retries nine times. Each error results in an error message of the form:

    DISK WT ERROR device # PRIMOS record # status-word

If the write is not successful after 10 tries, MAKE aborts, prints the message UNRECOVERED ERROR, and returns to the operating system. If a DISK-NOT-READY status is detected, a single disk error message is typed with a status of 177776. The software then retries to write, waiting for the disk to become ready. If a read is not successful, MAKE gives up and prints:

READ ERROR, RECORD = record-number

MAKE then asks:

ADD TO BADSPOT FILE?

If the user responds by typing YES, MAKE prints the track and head of the badspot and modifies the BADSPT file and DSKRAT appropriately. If a badspot is found on record 0 to 15 MAKE prints the error message:

CANT ADD RECORDS<=16 TO BADSPOT FILE. (MAKE)

and returns to PRIMOS level. 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.

If the user runs MAKE, then answers YES to VIRGIN DISK? and VERIFY DISK?, it is possible to find out immediately if there is any problem in the file structure part of the disk pack.

Final Operations

After the disk has been created, the user should unassign it with the UNASSIGN DISK command. Afterwards the operator should use the operator terminal to remove this disk from the table of assignable disks with the DISKS command. 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 FUTIL to copy the UFD DOS on a master disk to UFD DOS on the newly created disk. The BOOT file in the MFD that is read from the disk by the control panel boot 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

DESCRIPTION OF THE BADSPT FILE

The BADSPT file is a saved memory image. The file may be examined and modified by RESTORing it and referencing the file with the debugging command PSD. BADSPT is restored into consecutive memory locations starting at location '1000. A BADSPT file is created only if badspots exist; this can be verified by issuing the PM command following the command line:

RESTOR BADSPT

Pairs of numbers, starting at '1000, constitute the track and head number of each badspot on the disk. The file BADSPT can be no longer than one record. Thus, BADSPT records badspots. To add a badspot to the BADSPT file, place the track and head data in locations given symbolically by EA+1 and EA+2 (The value of EA is the second parameter printed by the PM command); then, give the command:

SAVE BADSPT 1000 new-end-address

where new-end-address has the value EA+2.

The BADSPT file may be used in conjunction with all disks except fixed head disks and diskettes.

Caution

The BADSPT file must not be deleted, or overwritten, or copied TO. This file is used by FIXRAT and COPY to process badspots.

OVERVIEW OF LABEL

LABEL initializes magnetic tapes just as MAKE initializes disks. LABEL writes either IBM (9-track EBCDIC or 7-track BCD) or ANSI (9-track ASCII) level 1 volume labels followed by dummy HDR1 and EOF1 labels. LABEL can also be used to read existing VOL1 and HDR1 labels. ANSI labels are written in accordance with the American National Standards Institute standard ANSI X3.27-1978. IBM labels are written in accordance with IBM's specifications (IBM manual GC28-6680-5). Any non-standard labels such as 7-track ASCII or user-defined labels cannot be read or written.

Using LABEL

To read existing labels type the command:

    LABEL MTn [-TYPE type]

To write labels type the command:

    LABEL MTn [-TYPE type] -VOLID volumeid [-OWNER owner] [-ACCESS access]
              [-INIT]

MTn — The tape drive where the tape to be labelled is located. n is a number between 0 and 7. This keyword is required and must be the first on the command line.

    -TYPE A = 9-track ASCII   (ANSI)
               (this is the default)
    -TYPE B = 7-track BCD   (IBM)
    -TYPE E = 9-track EBCDIC   (IBM)

volume-id — A 1-6 character string which uniquely identifies this tape reel. If less than 6 characters are specified, they are blank-padded on the right. The keywords -VOLUME or -VOL may be substituted for the keyword -VOLID.

owner — 1-14 characters long for ANSI labels, 1-10 characters long for IBM labels. If less than 14 (or 10) characters is specified, they are blank-padded on the right. If this keyword is omitted, the default is the user's login name. The keyword -OWN may be substituted for the keyword -OWNER.

access — A single character defining access to this tape. ACCESS is not used by Prime software but is included for completeness. If it is omitted it is left blank on ANSI labels. ACCESS is ignored for IBM labels.

-INIT — Necessary keyword for previously unformatted tapes.

ERRORS USING LABEL

Improper use of the LABEL command causes an error message to be printed. These errors are the result of bad syntax in the LABEL command itself or a system magnetic tape I/O error.

Syntax Errors

▶ ***DUPLICATE KEYWORD DETECTED

The same keyword was typed more than once. (Error 1)

▶ ***INVALID TAPE UNIT SPECIFIED

Something other than MT0-MT7 was typed. (Error 2)

▶ ***VOLUME ID SPECIFIED IS TOO LONG

The volume id cannot be longer than 6 characters. (Error 3)

▶ ***OWNER ID SPECIFIED IS TOO LONG

The owner id cannot be longer than 14 characters. (Error 4)

▶ ***INVALID LABEL TYPE SPECIFIED

Label type must be one of the characters "A", "E", or "B". (Error 5)

▶ ***NO MAGNETIC TAPE UNIT SPECIFIED

A magnetic tape unit is required. (Error 6)

▶ ***VOLUME ID WAS NOT SPECIFIED

When writing labels, a volume id is required. (Error 7)

▶ OWNER ID SPECIFIED IS TOO LONG FOR TYPES B OR E

The owner id for IBM labels cannot be longer than 10 characters. (Error 8)

▶ ***UNABLE TO WRITE TAPE LABEL ON THIS TAPE

A mag tape read error occurred and the label was not written. (Error 9)

▶ ***UNABLE TO READ TAPE LABEL ON THIS TAPE

A mag tape read error occurred and the label was not read. (Error 10)

▶ ***LABEL OPERATION ABORTED

Error 9, 10, 12, or 14 occurred and the label was not read. (Error 11)

▶ ***LABEL READ WAS NOT TYPE x

The label read was not of the type specified. (Error 12)

▶ ***ACCESS IGNORED FOR IBM LABELS (WARNING ONLY)

This is a warning only - processing continues. (Error 13)

▶ ***VOL1 LABEL ALREADY EXISTS

ANSI standards prohibit the re-writing of VOL1 labels. (Error 14)

▶ UNRECOGNIZED KEYWORD. string (CMDL$A)

An invalid keyword (string) appeared on the command line. (Error 15)

System Errors

Error	Meaning
MTn NOT ASSIGNED	Use command AS MTn before the LABEL command
subr EOF	END-OF-FILE on the magnetic tape
subr EOT	END-OF-TAPE
subr MTNO	The tape drive is not operational
subr PERR	Parity error on the tape drive
subr HERR	Tape drive hardware error
subr BADC	LABEL improperly called mag tape subroutines

(In the above errors, subr is the name of the mag tape subroutine that reported the error. See Reference Guide, PRIMOS Subroutines, PDR3621 for more information regarding these errors.)

If LABEL successfully writes a label, the message "TAPE LABEL WAS WRITTEN SUCCESSFULLY" is displayed. On read operations, LABEL prints out the volume and owner ids, creation date, access (ANSI tapes only), and other information.

HELP FACILITY

The command LABEL -HELP causes LABEL to print out a description of the command similar to that found in this document.

For a complete description of tape labels and their use, refer to the IBM publication GC28-6680, OS TAPE LABELS and the ANSI publicaton X3.27-1969, "American National Standard Magnetic Tape Labels for Information Interchange".

SECTION 7

BACKING UP

INTRODUCTION

Backing up is a procedure for making copies of current data files and programs. These copies are then available to be restored, in part or in full, in the event the files on the system are lost or broken. If desired, backup procedures may be contained on command files and invoked by the COMINPUT command. Backing up:

- Aids in maintaining the integrity of user data
- Provides recovery in case of loss of on-line data
- Gives the user the ability to restore earlier versions of files or programs
- Allows inactive files to be removed from the system

File losses may be considered 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 by fire, being dropped, etc.
- Operator error (running MAKE on a disk which contains current data, using a FUTIL UFDDEL from the MFD, etc.).

Causes of minor losses include:

- Power failure during write operations which may cause pointer mismatch, discrepancies with DSKRAT file, etc.
- User deletion or truncation of the wrong file by mistake (most common cause)

Scheduling

The exact scheduling of backups depends upon the installation, how often files are changed, and how important current information is. Example: One development system at Prime is a Prime 500 with two 300MB storage modules and one 80MB storage module. On Monday, Wednesday, and Friday mornings the active partitions on one of the 300MB modules are backed up onto a storage module with COPY; on Tuesdays and Thursdays the other 300MB storage module is COPYed. FIXRAT is run on the 300MB module not being backed up that day. The 80MB module is not as active and is not backed up during the week.

Two sets of backup disks are used to provide coverage between the weekly magnetic tape backups.

Each weekend the entire system is copied onto magnetic tape with the MAGSAV utility. Weekly tapes are kept for 2 months; the first tape of each month is kept for two years. This degree of backup protection is probably well in excess of that needed by the average system.

On the other hand, at another PRIME installation, MAGSAV is used to perform incremental backups, and a complete copy is done only one weekend per month. This procedure saves time and takes less 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 fire-proof vault.

BACKING UP ON DISK OR MAGNETIC TAPE

Backups may be disk-to-disk (using COPY) or disk-to-tape (using MAGSAV); both procedures are described in this section. The latter procedure is useful to the user of a small system which may have only one or a limited number of disks. Each method has particular advantages, which are made use of in the scheduling example above.

Disk-to-Disk Advantages

- Faster data transfer rate. Typically a fully-used 300MB module can be COPYed in about 1 hour. (Times may be scaled for smaller mass storage devices.) It takes 10-15 minutes to completely fill one 2400 foot reel of magnetic tape at the nominal speed of 45 inches per second; backing up on magnetic tape will take about 6 hours for a full 300 MB disk.

- Random access of data. Files in a storage module or disk can be accessed rapidly by PRIMOS using the directory tree structure. To locate a file on a magnetic tape backup, first the correct tape must be located and then this tape searched by MAGRST to find the desired file. The random access feature is even more convenient when more than one file is to be restored from the backup.

Disk-to-Tape Advantages

- Lower relative cost. Although it takes about 25 magnetic tapes (at 800 bpi) to backup a full 300 MB module, the cost of these tapes is less than the cost of another module.

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

COPY (DISK-TO-DISK)

COPY is an external command that copies and verifies a disk. COPY copies between physical devices (disks or partitions) under PRIMOS. Under PRIMOS, both disks must be ASSIGNed before invoking COPY. Since COPY duplicates a disk track-by-track (with badspot consideration) it is considerably faster than a directory-driven copying procedure. However, COPY cannot perform UFD compression or pointer cleanup. Use the FIXRAT utility for these procedures.

Running COPY

After the user types COPY at the terminal, the COPY command responds by printing a series of questions to which the user replies at the terminal.

Device Specification

When asked, the user must specify the device to be copied from (FROM), and the device to be copied to (TO). Depending upon the device number entered, ambiguity-resolving questions as: 1.5M WORD PACK? or 40MB STORAGE MOD? may be asked. The size of the FROM device and the TO device must be equal; the user must be sure to specify disks or partitions of the same size when replying to the FROM and TO queries, except for the cartridge module device. See CMD COPY discussion at the end of this section. If not, COPY prints the message: REC LENGTH AND NR RECS MUST BE = FOR BOTH DEVS and repeats the FROM question. The FROM and TO parameters are physical device numbers (see Appendix D).

If FROM or TO is not a valid physical disk number, or if the user replies NO to the query PARAMETERS OK?, COPY repeats the series of questions and waits for reply from the user. If the new parameters are acceptable, COPY initiates the copy operation.

Note

COPY does not allow rewriting of the same disk. For example, specification of the same physical disk number in reply to the FROM and TO queries is an illegal specification of COPY parameters.

Caution

A TO disk number must not be a disk connected to PRIMOS by the STARTUP command. It is good practice, when running COPY under PRIMOS II, to place all active disks that have a manual write-protection feature in WRITE PROTECT before initiating the COPY command, except for the disk to be written to (TO disk). It is good practice when running under PRIMOS, to place all disks assigned to the user terminal at which the COPY command is to be initiated to WRITE PROTECT, except the TO disk, before initiating the COPY.

Copying Method

COPY copies disk records from the FROM disk to the TO disk and, when done, verifies the copy by reading each record from both disks and performing a word-by-word comparison in memory. During this process, COPY displays the track number it is processing in the DATA lights on the processor control panel, bits 2-16. Bit 1 is off during the copy operation and on during the verify operation. When done, COPY prints DONE and returns to PRIMOS, which prints OK. If any disk read errors occur during the copy, the read is retried nine times. Each error results in an error message of the form (all numbers in the error messages are octal):

```
DISK RD ERROR device-number PRIMOS-record-number status
```

If the read operation is not successful after ten tries, PRIMOS ignores that record and prints the message:

```
DISK RD ERR, DISK=device-number record-number

ERROR IGNORED, COPY CONTINUED
```

Then, PRIMOS continues the copy operation. If any disk write errors occur, COPY 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 ten tries, COPY aborts, prints the error message UNRECOVERED ERROR, and returns to PRIMOS. 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 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 E for complete list of status words.)

When the COPY is finished, both writing and verifying, the message:

DONE

is printed at the terminal.

Badspots

COPY can copy FROM a disk or partition that has badspots. COPY looks for the file BADSPT in the MFD (see Section 6). If the COPY command finds the file BADSPT in the MFD, COPY interprets the BADSPT file and skips reading bad tracks on the disk to be copies FROM.

COPY skips writing tracks containing badspots on the TO disk. Those users who desire to use a disk containing badspots as the TO disk during a COPY operation should perform the following: create the FROM disk with the MAKE command and specify to MAKE the badspots on both the FROM and the TO disks. If the FROM disk already has something useful on it, the information must be backed up before invoking the MAKE command. Then, the information must be retrieved following the MAKE of the FROM disk, using either the FUTIL or the MAGRST command.

If the BADSPT file exists on the FROM disk, but has bad contents (e.g., from being overwritten or truncated), the message:

BAD BADSPT FILE, IGNORED

is printed; and COPY attempts to copy the disk as if the file BADSPT did not exist. It is worthwhile to try to copy a pack with badspots, for the purpose of backup. COPY skips writing tracks containing badspots on disks it copies TO.

Example: In this example COPY is used under PRIMOS II. Operation under PRIMOS is similar.

```
OK: COPY
GO
COPY 16.0
FROM PHYS DISK= 50
1.5M WORD PACK? NO
TO PHYS DISK= 51
1.5M WORD PACK? NO
FROM, TO, RECORDS = 000050, 000051, 6496
PARAMETERS OK? YES
```

DONE
OK:

Checksums

When copying between a disk written with a 4000 controller (obsolete) and a disk written with a 4002 cntroller, the controllers will generate different checksums. The command:

COPY NOCHECKSUM

copies the disk without checking checksums, thereby permitting the disk to be copied. During this procedure, there is a small risk that bad information will be copied without detection.

Restoring Data From Disk - FUTIL

Files, trees, and UFDs may be restored to an active disk from a backup disk with the FUTIL utility. The user must be familiar with the TO, FROM, COPY (and possibly TRECPY and UFDCPY) commands of FUTIL (see Reference Guide, PRIMOS Commands, FDR3108).

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 STARTUP command at the supervisor terminal. (Use the tables in Appendix D to construct the physical device number for the partition.)

The operator should then log in as a user. The backup, having been created by COPY, will have the same packname as the system disk to which the restoration will be done. Therefore, the user cannot specify the TO and FROM directories in FUTIL with the pathname as: <packname>UFD>etc. Instead the form of the pathname <logical-device-number>UFD>etc. must be used, because names are duplicated.

The logical device numbers may be obtained by giving the PRIMOS command STATUS DISKS. A table will be printed on the terminal whose first three columns are DISK, LDEV, and PDEV -- the packname, logical device number, and physical device number respectively. This table gives the logical device numbers corresponding to the known physical device numbers for the backup (FROM) and system (TO) disks.

Example: The system disk, containing two partitions SOFTWR and DBTEST, is mounted on unit 1. It is necessary to restore the tree structure of the UFD=DBADM in the MFD on the DBTEST partition. The backup disk is mounted on unit 0. To determine the logical device numbers, the STATUS DISKS command is given:

```
OK, STATUS DISKS

DISK    LDEV  PDEV  SYSN
SOFTWR   0    3462
DBTEST   1   71061
DBTEST   2   71063

OK,
```

Invoke FUTIL, define the FROM and TO directories and then restore the desired files and or directories with COPY, TRECPY, or UFDCPY as appropriate. Exit to PRIMOS when restoration is completed:

```
OK, FUTIL
GO
> FROM <1>MFD>DBADM
> TO <2>MFD>DBADM
> UFDCPY
> QUIT

OK,
```

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

Cartridge Module Device (CMD) - COPY

Multi-head partitions on the non removable section of a CMD may be backed up by copying to a set of removable CMD cartridges. The partition is copied one disk surface at a time. The cartridges must be restored in the order they were copied to a partition equal in size to the original partition. If the original partition being backed up had badspots, these will be transferred to the final partition. See the discussion of badspots earlier in this section.

Operation under PRIMOS: Power down the entire CMD prior to changing the removable cartridge. If part of the CMD is being used for paging, COPY should be run only under PRIMOS II. The message:

```
SURFACE n READY
```

prompts the user to change the cartridge. Power down the CMD and insert the next cartridge. Then power up the CMD and type YES to continue the COPY. If YES is typed before the disk is ready, COPY 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. See Figure 7-1 for an illustration of CMD copy.

When restoring a partition, if a surface is mounted in the wrong order, the program will inform the user which surface he has mounted in error, and then request the correct surface again.

Operation under PRIMOS II: As under PRIMOS, COPY prompts when to change the removable cartridge. If the disk is not ready, COPY waits until the disk becomes ready.

MAGSAV/MAGRST (DISK-TO-TAPE/TAPE-TO-DISK)

Complete descriptions of these commands will be found in Reference Guide, PRIMOS Commands.

General Information for Magnetic Tape - File Utilities

MAGSAV and MAGRST are utility programs that move files on any disk including storage modules, to a magnetic tape and vice versa. The files may be SAM, DAM, segment directories, UFDs, or an entire disk. Whenever a directory is specified, the directory and all components (the subtree) are transferred. Under PRIMOS or PRIMOS II, MAGSAV and MAGRST may be run from the supervisor terminal.

Logical Tapes

A logical tape may span multiple physical tapes, or a single physical tape may contain multiple logical tapes. The header record contains the tape name, data, real number, and revision number.

A disk file appears on tape as a record containing a pathname, followed by as many data records as are required for the file. The pathname contains the path from the file specified by the user to the current file. When an entire disk is saved, all pathnames begin in the MFD.

When physical END OF TAPE is encountered in either MAGSAV or MAGRST, a message is logged on the user terminal and a new tape unit is requested. The new unit may be the same as the old unit.

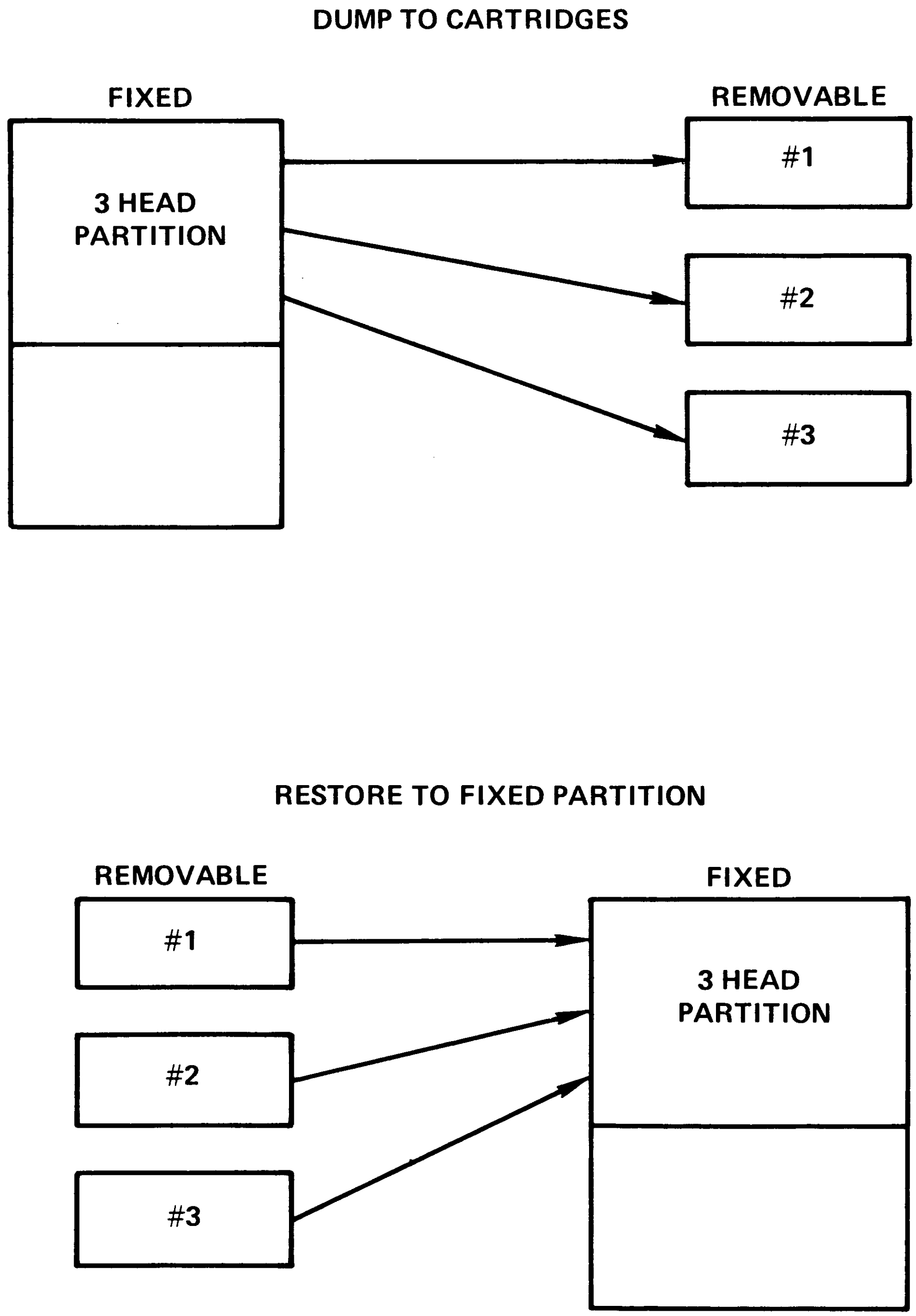


Figure 7-1. CMD Copy

Errors

Tape read or write errors are retried before being considered unrecoverable. Recoverable errors are handled by the PRIMOS system. Both recovered and unrecovered errors are logged. The first record on a tape is not retried. The total number of recoverable errors, if any occur, will be printed when the end of the tape is reached.

Assigning Tapes

When running MAGSAV under PRIMOS, the magnetic tape drive must be ASSIGNed. Files or directories in simultaneous use by other users must not be accessed by MAGSAV. An attempt to do so results in a backup tape with files that contain either partially written or partially updated information. Furthermore, there is a high probability that MAGSAV or other user programs will abort with the message:

```
FILE IN USE
```

since the two programs compete over which program is to gain access to the same file.

Backing up on Magnetic Tape - MAGSAV

MAGSAV saves files and directories from all PRIMOS-supported disks and disk partitions onto magnetic tape for backup and storage.

MAGSAV is invoked by:

```
MAGSAV [-7TRK] [-VAR] [-UPDT] [-INC]
```

-7TRK Use 7-track tape format; default is 9-track.

-VAR Increases tape record size to 2048 words and allows variable-length records, up to 2048 words, to be written. This is useful for large files as it decreases the amount of tape used for record headers and inter-record gaps. If this option is selected, MAGSAV prints the record size after the REV stamp of the MAGSAV dialog.

-UPDT Update. Set the DUMPED switch in the UFD entry for files and directories that are MAGSAVed; the default is not to set the switch.

-INC Incremental dump. Only those files and directories with a reset (=0) DUMPED switch are MAGSAVed. The default is to MAGSAV all files and directories.

Note

When a file is modified, its DUMPED switch is reset (=0). When the update option (-UPDT) is used with MAGSAV, the DUMPED switch is set (=1) for each file or directory that is MAGSAVed. If the MAGSAV program is run with the incremental save option (-INC), only files that have a zero DUMPED switch will be MAGSAVed. (i.e., only files that have been modified since the last time the MAGSAV program was run, will be MAGSAVed.

MAGSAV requests information in the following order:

Request	Response
TAPE UNIT:	The proper response is the physical unit number of the tape (0-7).
ENTER LOGICAL TAPE NUMBER:	The response is 1 for the first logical tape, 2 for the second, etc. MAGSAV rewinds the tape, then positions itself correctly. A response of 0 implies the tape is already positioned correctly and MAGSAV takes no action.
TAPE NAME:	Any six-character name.
DATE (MM DD YY):	The response format is MMbDDbYY where b represents a space and MM=month, DD=day and YY=year. The date is checked for validity and rejected if it is not valid. For example, 07 35 03 would be rejected. A (CR) answer enters the current system date (PRIMOS).
REV NO:	An arbitrary integer.
NAME OR COMMAND:	Enter a filename to be saved or an action command. filename is the name of the file to be saved on tape. $A pathname password changes the home UFD to pathname. $Q ends logical tape and exits to PRIMOS. $R ends logical tape, rewinds physical tape, and exits to PRIMOS. $I [filename] [level] writes an index into filename (default is terminal) to the level

specified (default is 2).

For example, $I FILEA 3 writes, into FILEA, an index of the MFD, any UFDs, and any entries in the UFDs.

$UPDT ON turns on update. The DUMPED switch will be set for all files and directories MAGSAVed. (Same as option -UPDT).

$UPDT OFF turns off update (default).

$INC ON turns on incremented dump (same as option -INC).

$INC OFF turns off incremented dumps (default).

$VALID ON checks each entry name to see that it conforms to file name rules.

$VALID OFF does not check entry names (default).

The prompt NAME OR COMMAND: is printed when writing or other action has been completed to allow further action or termination of the logical tape. Incorrect entry causes prompt to repeat.

To save an entire disk, attach to the MFD, enter MFD in response to the query NAME OR COMMAND: To save a UFD, attach to the MFD and give the name of the UFD that is to be saved; or attach to the UFD, and issue the SAVE * response. To save a file in a UFD, attach to the UFD and give the name of the file. MAGSAV also saves a disk that contains nested segment directories. Example:

```
OK, AS MT1
OK, MAGSAV
GO
REV. 16.2
TAPE UNIT (9 TRK): 1
ENTER LOGICAL TAPE NUMBER: 1
TAPE NAME: PDR311
DATE (MM DD YY):
REV NO: 0
NAME OR COMMAND: S1
NAME OR COMMAND: S2
NAME OR COMMAND: S3
NAME OR COMMAND: S4
NAME OR COMMAND: S5
NAME OR COMMAND: S6
NAME OR COMMAND: S7
NAME OR COMMAND: S8
NAME OR COMMAND: S9
```

```
NAME OR COMMAND: APPA
NAME OR COMMAND: APPB
NAME OR COMMAND: $R
OK,
```

Restoring Data From Magnetic Tape - MAGRST

MAGRST restores information from a magnetic tape into the PRIMOS file system. The format is:

MAGRST [-7TRK]

-7TRK Use 7-track tape format; default is 9-track.

All restore operations take place in the home UFD. MAGRST asks for the tape unit and logical tape number . MAGRST then prints the name, date, and revision on the user terminal and asks:

READY TO RESTORE: Enter the action to be taken.

YES restores the entire logical tape.

NO requests another tape unit/logical tape combination.

PARTIAL allows restoration of part of a tape (MAGRST will ask for files to be restored).

$I [filename] [level] creates an index to the level specified (default is 2) and writes it to filename (or to the terminal, if filename is not specified). The files and directories are restored.

NW [filename] [level] performs the same function as $I... but files and directories are not restored. The default value of level is 100. The option is useful to determine what is on the tape.

TREE NAME: This prompt is typed when a partial restore is requested. The pathname specified is restored from the tape to the disk. The prompt is repeated after each restoration until a null line (carriage return) is entered. The null line signals the end of restoration.

After each file is restored, the message:

FILE COMPLETE

is printed at the terminal. For a partial restore, files that have bad records are omitted. The pathnames of these files are printed along with an error message. The message:

RESTORE COMPLETE

is printed when the end of logical tape is reached.

MAGRST checks for conflicting file types when a file is going to be overwritten. Conflicts generate an error message, and the file is skipped.

MAGRST need not search all logical tapes when restoring sequential logical tapes. After MAGRST has exited to PRIMOS, the magnetic tape is not rewound. Instead, it is positioned at the location before the beginning of the next logical tape in sequence. In the case of sequential logical tapes, run MAGRST again and specify Ø to the prompt LOGICAL TAPE NO:. Then, the next logical tape is restored without rewinding and reading through the preceding logical tapes.

Index

MAGRST can index a tape and direct the listing of the index to a disk file rather than the user terminal. To use this feature, follow the the $I command with a filename, and then with the number of index levels.

Example of index with MAGRST:

```
READY TO RESTORE: $I TAPE#1 5
```

In this case, MAGRST restores the tape and writes a 5-level index to the file TAPE#1.

Example of Index generation without restoration:

```
OK, MAGRST
GO
REV. 16.2
YOU ARE NOT ATTACHED TO AN MFD
TAPE UNIT (9 TRK): 1
ENTER LOGICAL TAPE NUMBER: 1
NAME: PDR311
DATE(MM DD YY): 02-27-78
REV NO:      0
REEL NO:      1
READY TO RESTORE: NW 1
*** STARTING INDEX ***
S1
S2
S3
S4
S5
S6
S7
S8
S9
APPA
APPB
*** END LOGICAL TAPE ***
*** INDEX COMPLETE ***

OK,
```

In this case, MAGRST does not restore the files and directories.

SECTION 8

PRINTER OPERATOR (PROP)

Printer Operator (PROP) controls the actions and parameters of the printer spooling phantoms. It contains commands to define and modify such variables as paper size and the number of print lines per page, as well as operator commands to start and stop the phantom.

To facilitate interaction with PROP, each printer phantom has a unique name of 1-16 characters. Each name is the suffix for a group of files which control the action of the phantom. Table 8-1 lists these files and their meaning.

The command line has the following format:

```
PROP { label -command }
     { -STATUS        }
```

where label is the name of the phantom and command is a PROP command.

ENVIRONMENT

As mentioned above, each printer phantom has a name associated with it which identifies the files defining its environment. The PROP commands which create, modify and display this environment are listed in Table 8-2. Figure 8-1 illustrates the use of one of the commands, -DISPLAY and Figure 8-2 illustrates the use of another command, -MODIFY.

Environment definition mode assigns values to the parameters that make up an environment. CREATE sets up an environment that is not currently defined, while MODIFY changes an already defined environment. If MODIFY is used on a running phantom, it may be followed by one of the HANG or STOP arguments to determine when the environment changes will be made. Table 8-3 lists the sub-commands for CREATE and MODIFY.

The commands which display and delete existing environments include STATUS, DISPLAY and DELETE. STATUS prints a list of the currently defined environments and indicates which ones are currently being used by a phantom. DISPLAY prints a detailed description of a particular environment, while DELETE deletes a currently defined environment.

OPERATIONS

The PROP commands which operate the printer phantoms are: START, RESTART, STOP, HANG, ABORT, CONTINUE, DROP and BACK. Table 8-4 lists these commands. The spool phantom(s) normally run from UFD SPOOLQ and are run as part of the normal system start-up procedure. For example, the command:

PROP PRØ -START

starts the phantom spooler program for parallel interface (MPC) Printer Ø.

Note

> The login name under which the spooler phantom is run (usually SYSTEM) is the name under which the operator must login to use the operator commands of the spooler. If the operator, logged in under a different name, attempts to perform operator functions, the PROP program will return an error message and ignore the command.

PROP attaches to the first available local SPOOLQ directory and creates a file, R.label. This file remains open for the duration of time the phantom is running, thus providing a means for PROP to determine whether or not the phantom is running.

Operator communication is handled through the message files A.label, R.label and M.label (see Table 8-1).

QUEUE STRUCTURE

The spooler uses the UFD=SPOOLQ to store files in the print queue. If multiple systems are connected over a network, one of these directories should exist for each system. In addition to the temporary print files, a control file, Q.CTRL, used by the queueing mechanism is in this UFD. This file contains information about each entry in the spool queue, queue top and bottom pointers, and a locking flag and counter to provide integrity when multiple readers and one writer are allowed to access the file simultaneously.

The queue structure permits a maximum of 2ØØ entries. Any attempt to exceed this will yield an error message from the spool program.

The queue file is a fixed length, "circular" file (i.e., when the last physical entry is written, it wraps around to the first entry). There is one six-word file header and 2ØØ 4Ø-word data entries. Each data entry has a one-word header, which is set to 1 if the data entry is valid and Ø if the entry is null (invalid). The head pointer points to the next entry to be popped from the queue and the tail pointer points to the next entry to be added. Because the head and tail pointers are equal when the file is empty as well as when it is full, an 'empty' flag was included in the header. If the flag is set to 1, there is one or more entries in the queue; if Ø, the queue is empty.

```
PROP ENGPR0 -DISPLAY
[PROP rev 17.2]

DEVICE: PR0
PAPER:
DEST:
    1
    2
    3
    CAROUSEL
    RESEARCH
MESSAGE:


COMOUT: ON
UPCASE: OFF
PRINT:  ON
PLOT:   OFF
LENGTH:     38
LARGE:      30
LIMIT:   30000
UPPER:      40
LOWER:       0
HEADER:      2
WIDTH:     108
LINES:       0
```

Figure 8-1. Illustration of PROP Environment ENGPRO

```
OK 06:46:19  0.281  0.327  level 6+
PROP ENGPR0 -MOD
[PROP rev 17.2]

> PAPER WHITE
> LENGTH 54
> WIDTH 88
> LINES 66
> FORM WHIT
> FILE
Wait... Acknowledged.

OK 06:48:21  0.742  0.706  level 6+
PROP ENGPR0 -DISPLAY
[PROP rev 17.2]

DEVICE: PR0
PAPER:  WHITE
FORM:
   'WHIT  '
DEST:
    1
    2
    3
    CAROUSEL
    RESEARCH
MESSAGE:


COMOUT: ON
UPCASE: OFF
PRINT:  ON
PLOT:   OFF
LENGTH:     54
LARGE:      30
LIMIT:   30000
UPPER:      40
LOWER:       0
HEADER:      2
WIDTH:      88
LINES:      66

OK 06:48:39  0.421  0.363  level 6+
```

Figure 8-2. Illustration of Modification of ENGPRO

	Word	Contents
Queue File Header:	0	lock flag
	1	lock counter
	2	head pointer
	3	tail pointer
	4	entry size
	5	empty flag
Queue Entry Data:	0	.
	1	user name
	2	.
	3	.
	4	.
		file name
	17	.
	18	.
	19	.
	20	form type
	21	.
	22	month
	23	day
	24	year
	25	time
	26	options
	27	defer time
	28	file size
	29	.
	30	pt file name
	31	.
	32	raster scan size (plot)
	33	.
	34	.
		reserved
	38	.
	39	.

Figure 8-3. Queue File Format

Table 8-1. Environment Filenames.

Filename	Function
O.label	Contains cominput created by PROP for the phantom, label.
A.label	Contains the last operator request received by the phantom, label. If A.label is open, it means that the request has been received but not yet acted on.
R.label	Contains operator requests which have not yet been received by the phantom, label.
M.label	Contains messages from the phantom, label to the system console.
E.label	Contains the environment definition for the phantom, label.
O_label	Contains COMOUTPUT for the phantom, label.

Table 8-2. Environment Commands.

Name	Function
-CREATE	Sets up a new phantom environment with the name given in the PROP command. See Table 8-3 for a list of the sub-commands that set up the environment.
-MODIFY	Modifies the phantom environment named in the PROP command. If MODIFY is used on a started phantom, use one of the -HANG or -STOP arguments (see Table 8-4) to control when the changes will be made. See Table 8-3 for a list of the sub-commands that modify the environment.
-DELETE	Deletes the phantom environment named in the PROP command.
-STATUS	Prints a list of the currently defined environments and whether or not they are running.
-DISPLAY	Prints a detailed description of the phantom environment named in the PROP command.

Table 8-3. Environment Subcommands

PLOT [ON / OFF]	If ON, scan the queue for PLOT files. If OFF, ignore PLOT files in the queue.
PRINT [ON / OFF]	If ON, scan the queue for PRINT files. If OFF, ignore PRINT files in the queue.
DEVICE [PR0 / PR1 / PR2 / PR3 / CENPR / CE2PR / PLOT / AMLC n]	Output to this device. If AMLC is selected, it must be followed by an octal line number. The AMLC line will be used as configured on the system console.
UPCASE [ON / OFF]	If ON, convert all lower case characters to upper before printing. If OFF, do no conversion.
HEADER [0 / 1 / 2]	Set the number of header pages. A setting of 2 gives a trailer page.
PAPER [name / '____']	Mount name forms. Name is from 1-6 characters. All FORM synonyms are deleted.
LENGTH [n / 38]	Print n lines per page.
LINES [n / LENGTH+13]	N is the number of physical lines per page.
WIDTH [n / 108]	N is the number of physical columns on a page. Used for formatting header and/or trailer pages.
LIMIT [n / 30000]	Do not print any files bigger than n disk records.
UPPER [n / 63]	Do not look for SPOOLQs on any logical disk with a number higher than n, where n is a decimal number.
LOWER [n / 0]	Look for SPOOLQs starting at logical disk n, where n is a decimal number.
LARGE [n / 30]	Print files whose length in records is less than n before larger files.

Table 8-3. Environment Subcommands (continued)

COMOUT [ON / OFF]	If ON, keep a COMOUTPUT file of all phantom actions. The name of the file is SPOOLQ>0_label. If OFF, turn COMOUTPUT off and append to the file if it already exists.
MESSAGE text	Print text on every header page. Text is one line of up to 80 characters. If text is omitted, a null message will result.
DEST synonym	Defines a printer name synonym.
UNDEST synonym	Deletes a printer name synonym.
FORM synonym	Defines a printer name synonym.
UNFORM synonym	Deletes a printer name synonym.
QUIT	Exits environment definition mode. Any parameter changes are ignored; the file E.label remains unchanged.
FILE	Exits the environment definition mode. If CREATE was used to enter environment definition mode, the environment file E.label is created and defaults used whenever the user did not specify a value. If MODIFY was used to enter environment definition mode, the environment file E.label is updated.

Table 8-4. Phantom Operation Commands

Name	Function
-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 issued paper change to occur immediately.
-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	Continue printing/scanning. Takes the spooler out of HANG mode.
-STOP {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 logged out.
-START	Starts the spooler printing/scanning. Used to start the spooler initially or after STOP.
-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.
-RESTART	Restarts the spooler after it has been halted by running out of paper, a paper jam or similar cause. Printing of the file restarts from the beginning of the file.
-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.

SYNONYMS

The phantom environment may contain synonyms for both the form type (-PAPER and -FORM subcommands) and the printer name (-DEST subcommand) which map to the same name at the phantom level. The phantom matches the spooler -FORM name with the environment -PAPER and -FORM names, as well as the spooler -AT name with the environment -DEST names. For example, while form type 'WHITE' might be mounted, if it has synonyms 'PET', 'NARROW' and '8_x_11', then any request with any of those names after the spooler -FORM option is eligible for printing. For example:

```
SPOOL RPE-T-442 -FORM 8_x_11
```

prints on the printer with WHITE forms mounted. The only noticeable difference between doing the above and specifying -FORM WHITE is that the word 8_x_11 will be found somewhere on the header page.

SECTION 9

BATCH FOR THE OPERATOR

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 the Batch monitor
- Stopping the Batch monitor
- Monitoring the Batch subsystem
- Controlling users' jobs when requested to do so

STARTING THE BATCH MONITOR

Three commands are needed to start the Batch monitor:

1. PHANTOM BATCHQ>PH_GO
2. CHAP -usernumber priority timeslice
3. BATCH SYSTEM -START

The first two commands usually are included in the C_PRMO command file (Section 20), and thus are normally part of the system start-up routine. However, they may be given by the operator if the Batch monitor is stopped and restarted while the system is running. In this case, the operator replaces "usernumber" with the user-number of the Batch monitor's phantom. "Priority" and "timeslice" should generally be set to "1 24" (default) to keep Batch jobs at the same or lower priority than interactive jobs. (The monitor's priority and timeslice represent the greatest priority and timelsice any Batch job can have. Job priority and timeslice can be lowered for each queue with Batgen's rlevel and timeslice subcommands: see Section 12 for details.)

The Batch System -Start Command

When C_PRMO finishes, the operator sets the system time and date with the SETIME command. When date and time have been set, the started phantom initializes the Batch monitor and sends the message:

    BATCH Waiting for BATCH SYSTEM -START

The operator then gives the command:

    BATCH SYSTEM -START

This command starts the monitor and returns the message:

Start-up request issued.

Note

The BATCH program may return the message:

Process not started.

This means either that the phantom has not been started, that the phantom was started so recently that it is not yet ready to accept the startup command, or that date and time have not been set. Wait for the *BATCH* Waiting for BATCH SYSTEM -START message to appear at the supervisor terminal. Then give the BATCH SYSTEM -START command again.

When the monitor has finished its start-up routine and is ready to process users' jobs, it returns the message:

BATCH Operator Start-up.

The Batch system is then accessible to users.

Note

The monitor may return the message:

BATCH Start-up request previously processed.

This means that the monitor is running already and has disregarded the new start-up command.

STOPPING THE BATCH MONITOR

To stop the Batch monitor, give the command:

BATCH SYSTEM -STOP

The Batch monitor will respons "Stop request issued." If it responds, "Process not started," then the monitor is not running or is just starting up (see above).

When the monitor sees the STOP command, it sends the message, "*BATCH* Operator stop" to the supervisor terminal, and then logs itself out.

Do not LOGOUT the monitor (using LOGOUT ALL or LOGOUT -nn), SHUTDN ALL, or SHUTDN the partition on which BATCHQ resides while the Batch monitor is running; these actions 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. Usually, the repairs can be made by giving the

PH BATCHQ>PH_GO command, which runs *FIXBAT, or by running *FIXBAT interactively. If *FIXBAT cannot repair the database, the System Administrator must run the C_BDIF or C_RSET command file. (Details on these files, as well as on running *FIXBAT, are given in Section 12.)

Example

Here 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. (The long form of the PRIMOS prompt is in effect. See the writeup on RDY in the Prime User's Guide for details.)

```
OK 14:01:36 5.606 0.721
batch system -stop
[BATCH rev 17.2]
Stop request issued.

OK 14:02:05 1.378 0.500

*** 58 14'02
*BATCH* Operator stop.
batch -display
[BATCH rev 17.2]
Warning: jobs are not being processed at this time.

No queues have waiting or held jobs.
No running jobs.
    .
    .
    .
OK 14:05:23 10.418 1.66
ph batchq>ph_go
PHANTOM IS USER 63
OK 14:07:00 1.793 0.818
*** 63 14'07
*BATCH* Waiting for BATCH SYSTEM -START
chap -63 2 50   /* 40 decimal
batgen -status   /* Make sure queue file is okay.
[BATGEN rev 17.2]

Queue:       Status:
------------ ---------
Express      unblocked
Normal-1     unblocked
Normal-2     unblocked
Background-1 unblocked
Background-2 unblocked


OK 14:07:32 0.984 0.281
```

```
batch system -start
[BATCH rev 17.2]
Start-up request issued.

OK  14:07:47  0.390  0.360

*** 63 14'07
*BATCH* Operator start-up.
```

MONITORING BATCH

To determine the general status of the Batch system (and monitor), give the command:

```
BATCH -DISPLAY
```

This will output the number of waiting and held jobs per queue, and the file name, user name, and queue name for each curently executing job.

```
BATCH -DISPLAY
[BATCH rev 17.2]

Number of waiting and held jobs:

Queue        Jobs
------------ ----
Normal-1     5
Normal-2     4
Express      2
Background-1 2
Background-2 10

Total=       23

Currently running jobs:

 User  Jobid#  # Queue
------ ------ -- ------------
BURLEY #00001 62 Normal-1
CROW   #10002 60 Normal-2
GRUBIN #30000 63 Background-1
WEBB   #40000 61 Background-2

OK  13:48:55  1.366  0.395
```

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 id number given the job by the Batch monitor. When the job completes (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 too many inquiries.

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, the JOB -STATUS command displays the job name, job-id, queue, and status 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 17.2]

 User   Jobid#   State              External name            Queue
------ ------ --------- ------------------------------ --------

ELS    #00002 executing trepro.cmd                      Normal-1
BURLEY #00003 waiting   o_batchtest
BURLEY #00004 waiting   o_batchtest

BURLEY #10001  held     o_batchtest                     Normal-2
CROW   #10003 waiting   rev.emf

WEBB   #20008 executing mergtest.nw                     Express
WEBB   #20009 waiting   mergtest.cl

WEBB   #30003 waiting   o_sort8                         Background-1
WEBB   #30004 waiting   o_sort9
WEBB   #30005 waiting   o_sort10

GRUBIN #40012  held     subsys.sdf                      Background-2
GRUBIN #40013  held     subsys.sdf
NEILS  #40016 waiting   o_sqrl
```

By giving the job-id in the JOB -STATUS or -DISPLAY command (e.g., 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-- e.g., JOB SYS5 -DISPLAY.) For example:

```
JOB #20000 -DISPLAY
[JOB rev 17.2]

Job mergtest(#20000), user WEBB executing (queue Express).
Submitted today at 1:39:30 p.m., initiated today at 1:49:25 p.m.
Funit=6, priority=9, cpu limit=3, elapsed limit=3.
```

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. He 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. For details, see Section 12.

CONTROLLING BATCH JOBS

The operator has nearly full control over all jobs in the batch subsystem. While logged in under SYSTEM, he can perform any operation on a user job that the user himself could perform, with the following restrictions:

- He must refer to all user jobs by their job-id (instead of job-name).
- He cannot ABORT or RESTART any jobs unless he is either at the supervisor terminal or is referencing his own jobs.

If the operator attempts to ABORT a job from a terminal other than the supervisor terminal, the abort will fail. If he 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 #40000 -ABORT
[JOB rev 17.2]
Job s_(#40000) aborted.

OK 13:51:58  1.987  0.396
```

```
***58 13'52
*BATCH* Job s_ for SMITH(#40000) aborted.

***58 13'52
*BATCH* Executing subsys.sdl for GRUBIN(#40001).
```

SUMMARY OF OPERATOR JOB COMMANDS

	Command	Use
JOB jobid	-CANCEL	Cancel a held or waiting job.
	-ABORT	Abort a running, held, or waiting job.
	-RESTART	Terminate, then restart a job.
	-HOLD	Hold a job in the queue.
	-RELEASE	Release a held job so that it can run.
	-STATUS	Display the status of a job.
	-DISPLAY	Display status and parameters of a job.

The -HOLD and -RELEASE options are only available to the operator. 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. But 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 with the command:

```
JOB job-id -RELEASE
```

SECTION 10

SYSTEM HALTS

GENERAL

Under certain unusual circumstances (hardware or software malfunction), PRIMOS will execute a HLT instruction and halt. A system halt is indicated by the red light on the control panel. This light is directly above the rotary switch. It will light up when the system processor stops.

PRIMOS

Checks

Hardware-related halts are called checks; see Reference Guide, System Architecture for a complete discussion of checks. When PRIMOS halts due to a check, an octal address is displayed in the control panel lights. This is called a coded halt.

Checks indicate exceptional (sometimes serious) hardware conditions. If a check occurs, four words of information are saved in a check header (PB high, PB low, keys, and modals), and control is transferred to the word following the check header. This word is displayed in the data lights. Check headers are in segment 4 of PRIMOS and their locations are not expected to move. Checks currently defined are:

Symbol	Header	Lights	Description	Action
PWRFL_	200	206	Power failure	D, C
MEMPA_	270	277	Uncorrected memory parity error	X
MCHK_	300	306	Machine check	D, R, W
MMOD_	310	316	Missing memory module	D, C

Memory Errors

For software-related halts, the PRIMOS load map (M_PRMOS) and the contents of the data lights are used to find the halt location.

After the machine halts find the halt address:

1. Turn the rotary switch to STOP/STEP.

2. Place the ADDRESS/DATA switch to ADDRESS. Write the address displayed in the lights in the logbook.

Determine the segment number of the halt:

1. Turn the rotary switch to FETCH Y.
2. Place ADDRESS/DATA switch to ADDRESS
3. Push the DATA CLEAR switch.
4. Enter '14 in the lights.
5. Place sense switches 1 and 4 up.
6. Place the ADDRESS/DATA switch to DATA.
7. Push START. Write the segment number displayed in the lights in the logbook.

Using the segment number and address, find the symbolic address corresponding to the halt loction in the PRIMOS loadmap. Then perform the action(s) listed below for that halt. Record the location and the results of the action(s) in the logbook.

Halt Location	Halt Reason	Action(s)
AMLCI_	Bad AMLC Interrupt	D, W
BDMEM_	Bad memory during Cold Start	S
BOOT0_	Halt after SHUTDN ALL	C
MCHK_	Machine check	See Checks above
MEMH2_	Halt after automatic mapout of bad page	W
MEMPA_	Uncorrected Memory Parity	See Checks above
All others		D, C

The action codes are described later in this section.

ACTIONS FOR PRIMOS HALTS

C Cold Start. Bring the system up as if it had been shut down normally as described in Section 2.

D Crash magnetic tape dump. Whenever the operator or administrator is in doubt about the nature of a halt, it is good practice to take a magnetic tape dump, especially before doing a warm start which will erase the machine conditions that existed at the time of the halt. Mount a scratch tape on drive unit 1. Push the MASTER CLEAR switch. Restart at address '776 (See Appendix C). The tape will write and then rewind. Label the tape with date and time of halt (and machine, if more than one). Record this information in the logbook. The tape should be turned over to the person whom the System Administrator has designated to examine unusual halts.

R 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 up.

5. Place 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.

Note

Register '27 is for the Prime 750 only.

S Turn the rotary switch to RUN. Push the START key. If the halt is a parity error during cold start, this action will automatically map out the bad page. The halt is in segment 14 of PRIMOS.

W Warm Start.

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.

Caution

Never perform a warm start before deciding whether or not a dump of machine conditions are necessary. If they are, perform the dump, then perform the warm start.

Note

Warm start runs the memory parity finder before restarting PRIMOS. Thus, the warm start may run for twenty seconds or more before the WARM START message appears at the user terminals; no 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.

X Uncorrected memory parity. Access locations Ø, 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:

location	register	contents
Ø	X	user getting parity error
1	A	page number
2	B	offset within page

Note

If user number is 1 (supervisor) then a Cold Start must be done.

If user number is not 1, turn the rotary switch to RUN and then press the start key. This will automatically map out the bad page. Following mapout, the system will halt at MEMH2_ and must be warm started (see W).

PRIMOS II

PRIMOS II may be halted by the operator to abort a long listing or to recover from a bad STARTUP command. 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.

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, 4 up).
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:
```

PART III - THE ADMINISTRATOR

SECTION 11

BUILDING AND EXTENDING THE SYSTEM

This section is an outline of the contents planned for future versions of this PDR. Certain parts of this Section appear in their entirety. Where available, references have been made to other sections of this guide or to other documents where the information is available.

PRIMOS

Versions of PRIMOS

PRIMOS II: This is the sinlge-user operating system. See section 19 for details.

PRIMOS: Building PRIMOS (the multi-user operating system) is treated in detail in Section 20. Configuring the operating system is described in Section 17.

Paging Space

See section 20 on PRIMOS for discussions on calculating paging space requirements. Section 6 on disk formatting explains how to create split disks.

MAKE - DISK FORMATTER

If the system is being bootstrapped, or if the storage module is to be formatted, it may be necessary to build MAKE by running the command file C_MAKE before formatting any disks. During this process, the file *BOOT is created. *BOOT is an image of the record 0 bootstrap, as it would appear on the storage module. This run file may be used to place the storage module bootstrap (See Appendix A) on disk volumes other than storage modules without rerunning MAKE. *BOOT may also be used to change the start-up parameters for the supervisor terminal.

To save *BOOT as BOOT, perform the following:

```
RESTOR *BOOT
SAVE BOOT (with the following parameters)
```

Parameter	448-record	1040-word record
SA	3011	3011
EA	3667	5103
P	0	0
A	0	0
Bx	Controller Control Wd	
KEYS	SOC Control WD 1	

SOC Control WD 2

The SAVE command for the storage module is:

SAVE BOOT 3011 5103 0 0 110 27 74006

The SAVE command for all other disks is:

SAVE BOOT 3011 3771 0 0 110 27 74000

*BOOT

MAKE contains nine words which correspond to the SAVE parameters described above. To enable MAKE to create bootstraps (see Appendix BOOT) for systems which do not use TTY as the supervisor terminal:

1. A FIXRAT

2. ED BOOT

   Change the nine words beginning at location START

   FILE

3. Run the command file C_MAKE

4. FUTIL *MAKE to CMDNC0 as MAKE

The serial interface controller control words (Port 1) are:

Speed (Baud)	Serial Interface Control Words	SOC Port Sel and Speed (SOC Control Word1)	SOC Character Definition and Delay (SOC Control Word2)
110	110	27	74006
300	1010	76	34006
1200	2010	373	34006
9600	3410	3735	34006

SHARED LIBRARIES

These libraries, which are all or partially shareable, are the FORTRAN library (the I/O routines only) and MIDAS for all users; and COBOL and/or FORMS for those users who have purchased these packages.

Installation of the shared libraries is the default. Small systems with few users and only one MIDAS user or one COBOL user or where the FORTRAN formatted I/O routines are seldom used, may see no benefit from shared libraries.

Features of Shared Libraries

Each user of shared library routines uses private segment '6001 in addition to the segments otherwise required by programs. Segment '6001 is used for the impure portion of the shared libraries and represents a reduction in the size of the user's load file but not in the size of the single user working set at run time. This additional segment may be compensated for by a corresponding reduction in the number of segments in the run file. (The MI option of SEG's Loader is used for this; see LOAD and SEG Reference Guide, IDR3524).

Several benefits result from using the shared libraries.

- User run files will be smaller, reducing the time required to restore the SEG runfile. User interaction with the program will begin sooner.
- Reduction of system load with respect to private segments and private memory image sizes. If properly used, paging may be reduced. This is of importance for users with many large V-mode programs making extensive use of the shared library routines.
- Installation of a new revision of the library does not necessitate program reloading. Installation of a rebuilt shared library is all that is required to make the modified library available to all users of the shared library.

PRIMOS should be shut down and rebooted when a shared library is installed.

Installation

Shared libraries occupy segments '2014 and '2050 and must be installed each time the system is cold started. The runfiles are resident in UFD SYSTEM of the Master disk. Copy these runfiles to UFD SYSTEM on the system disk. The commands that install the runfiles at startup time may be incorporated into the C_PRMO file (as in the example in Section 2) or called from C_PRMO by COMINPUT command. The commands are included in C_PRMO.TEMPLATE in UFD PRIRUN.

This command file installs memory image files in segments '2014 and '2050 and runs the programs required to inform the operating system that shared libraries are activated. Once the libraries are installed, users with programs loaded using the special shared library object files may run V-mode progams accessing these shared libraries. If the shared libraries are not installed, programs expecting the shared libraries to be resident will get an error message from the operating system whenever an attempt is made to access a shared library routine.

Shared Library Usage

Either all or none of the libraries must be shared. If one of the shared libraries is to be used, all appropriate shared libraries must also be used. If the user wishes to use the shared FORTRAN library and also requires MIDAS or COBOL, the shared MIDAS and COBOL libraries must also be used. After the new V-mode run file has been created, and the shared libraries installed, the user's programs may be run as before.

Administration

The shared libraries files, in UFD LIB, are PFTNLB, VKDALB and, for those purchasing COBOL and/or FORMS, VCOBLB and VFORMS.

If the shared libraries are not to be used system-wide, then those users planning to use them must modify their command files to use the non-shared library files.

Rebuilding and Reinstallation

Each of the shared libraries is represented by a set of runfiles and an installation program. If only one of the libraries must be replaced it is necessary to rebuild that library only. These command files put all the necessary files into UFD SYSTEM so that installation is easily accomplished by running the command file C_SHLB in that UFD.

Caution

A library should not be replaced while being used. As programs using the shared libraries execute, links are made to the appropriate shared library routines in such a way that altering the memory image in use by the program can cause random and unpredictable behavior. Changing a shared library (replacing its memory image in segment '2014) has the effect of making such an alteration to the user's memory image. Install new shared libraries only when bringing up the system with a cold start.

It is safe to replace the memory image files in UFD SYSTEM at any time as these are only loaded into memory when the explicit SHARE commands are given.

MIDAS

See the MIDAS Reference Guide, PDR3061.

SPOOLER

See Section 8 for operation, and Appendix SPL for installation options.

NETWORKS

To install networks, copy NETCFG from the Master Disk:

```
FUTIL
>F net-ufd>CMDNC0
>T CMDNC0 password
>C NETCFG
>QUIT
```

net-ufd is X.25 if you have the version which interfaces to the public data network (PDN); otherwise it is PRINET. password is the owner-password of CMDNC0.

Remove the obsolete CONFIG directives MYNAME, NET, FAM, RLOGIN from the configuration file, and replace them with the single CONFIG directive NET ON. The myname parameter of the one-line CONFIG command (or directive) is also obsolete. Specifying an obsolete CONFIG directive related to networks results in a cold start error message.

The network configuration file, NETCON, must be created with the NETCFG command (See Section 18 and the PRIMENET Guide). Once all questions describing the network have been answered in the dialog with NETCFG, the binary file NETCON will be placed in CMDNC0. NETCON will contain the information formerly supplied by the CONFIG directives NET, FAM, MYNAME, and RLOGIN.

The SMLC CONFIG directives are not recommended when configuring networks, as they disable all SMLC mapping from the configuration file. The SMLC directives are intended for those sites that use SMLCs without networks.

Configuration and Installation of FAM

The source, object, run, and command files for the File Access Manager FAM are contained in the directory PRINET>FAM. The files in UFD FAM, of special importance to the FAM installer are:

PH_FAM	Phantom command file
#FAM	Run file

To install the FAM:

1. Create a UFD called FAM (which may be logged into). This UFD must not have a password.

2. Copy the files PH_FAM and #FAM to this newly created UFD from the Master Disk, using FUTIL.

To enable the FAM either:

1. LOGIN under the username of FAM:

```
OK,  LOGIN FAM
FAM  (xx) LOGGED IN AT ...
OK,  SEG #FAM
GO
```

FAM will now run, and no further commands will be read from the terminal.

OR

2. Run the FAM as a phantom:

```
OK,  PH FAM>PH_FAM
PHANTOM IS USER ...
OK,
```

See NETCFG (Section 18) to see how FAM is enabled to communicate with a particular remote node. If remote nodes are not specified properly with NETCFG, FAM will terminate with the message '****ST 26' at the terminal which enabled FAM. The message 'FAMSTOP AT 000026' will be printed at the supervisor (user 1) terminal.

VERSIONS OF SEG

See LOAD and SEG Reference Guide, IDR3524.

SYSTEM SECURITY

Adding to UFD CMDNC0

See section 3.

Adding new directories to the MFD

See section 3.

Protecting the volume-name file (DSKRAT)

It is recommended that the access rights to the disk record availability file be set to READ ONLY. This prevents possible loss of the file (and the partition) by unauthorized or inadvertent modification of the file. The procedure is:

```
A MFD owner-password logical-disk
PRO volume-name 1 1
```

owner-password — The owner password of the MFD.

logical-disk — The logical disk number (octal) on which the MFD resides.

volume-name — The name of the logical disk on which the MFD resides; it is also the name of the disk record availablity file. volume-name is created as DSKRAT by MAKE. It may have been changed, using CNAME; check in your system logbook.

SECTION 12

BATCH FOR THE ADMINISTRATOR

INTRODUCTION

Batch is the most flexible of PRIMOS' command file utilities. It makes phantom execution of jobs easier for the user, while giving the operator and system administrator greater control of the environment and execution of the jobs. It does this by allowing the System Administrator to define from one to six Batch queues from which user jobs can run as phantoms. These phantoms can be set to run "in the background" of the system: that is, to run concurrently with interactive jobs, but at somewhat lower priorities. In this way, they use only small amounts of CPU time when interactive use is heavy, but use large amounts of CPU time when interactive use is light or absent. Furthermore, Batch jobs may be held in their queues by operators, then released to run at appropriate times. For example, extremely long jobs, such as file updates and backups, can be set up as Batch jobs during the day, then run under operator control at night.

Each Batch queue is a separate entity, defined by the System Administrator to be particularly hospitable to certain types of jobs. Queues are defined by nine parameters:

- Name
- Default CPU time limit
- Maximum CPU time limit
- Default elapsed time limit
- Maximum elapsed time limit
- Default PRIMOS file unit for command input
- Default value for priority of job within queue
- Relative run-time priority
- Timeslice

Queues designed for short jobs have a fairly high scheduler priority and a short timeslice; queues designed for normal jobs have slightly lower priorities and normal timeslices. Queues designed for long jobs have low priorities but large timeslices. The queues for short jobs will thus run fastest, as they can operate during times of heavier interactive use. The other queues will take fuller advantage of periods of lighter activity.

Queues and Users

Users submitting jobs may specify the queue, the CPU limit, the elapsed time limit, the file unit, and the priority within the queue they want for their jobs, or they may specify none of those things, thus requesting the Batch monitor to place their jobs in the first available queue and assign to them the queue's default values for time limits, file unit, and priority. The System Administrator must make that first available queue a reasonable default queue, or else must ensure that users know which queues they should use and what the time limits and default values of those queues are.

By using the BATGEN -STATUS and BATGEN -DISPLAY commands, users can see what queues are available and what their characteristics are. They can then submit their jobs to the appropriate queues.

How Batch Allocates Phantoms

A Batch subsystem can consist of a single queue with no limits (except for user-defined ones) placed on jobs running within it. The system then simply runs jobs sequentially, their only priorities being those requested by the users. (Priorities range from 9 down to 0.)

Alternatively, the system can contain from two to six queues. In this case, the Batch monitor checks each queue in turn, beginning with queue number one. If it finds a job waiting to run, and a phantom is available, it runs the job. If six queues have jobs, and six phantoms are free, then one job from each queue is started. When the last of these jobs has been started, the monitor begins checking each queue again, to see if any jobs have finished or aborted. If a job has finished, the monitor does the necessary finishing-off (marking the job as completed or aborted, deleting its temporary files, etc.) and then checks the queue for another waiting job.

A slightly different situation arises if there are fewer available phantoms than queues. For example, if there are three queues, but only one phantom available to run jobs, the monitor will run all the jobs from queue one before running a job from queue two; and it won't run jobs from queue three until queues one and two are both empty.

Care and Upkeep of the Batch Subsystem

The System Administrator has the responsibility for:

- Installing the Batch subsystem
- Defining the Batch environment
- Modifying the environment when necessary
- Insuring that FIXBAT is run periodically to keep queues updated and orderly
- Deleting and replacing queues as they fill up

These tasks are discussed in this section.

The System Operator has the responsibility for:

- Starting and stopping the Batch monitor
- Holding and releasing jobs for users who request this service
- Monitoring the Batch subsystem

These tasks are discussed in section 9.

INSTALLING THE BATCH SUBSYSTEM

The Batch system directly involves three top-level UFDs on each system. These are CMDNC0, BATCH, and BATCHQ. CMDNC0 is used to hold only the commands ($$, BATCH, BATGEN and JOB) themselves. BATCH contains the source code for the Batch subsystem, including the command files necessary to build and install itself. BATCHQ contains the command files used to initialize the Batch database (which will also reside in BATCHQ).

As shipped, these UFDs have no passwords; or more precisely, they have passwords composed of six blanks: '   '. You may want to create owner passwords for them, in order to increase security within the Batch system.

To install (or rebuild) and password the Batch subsystem, take the following steps:

1. Decide what passwords to use (if you want passwords). (The password for BATCHQ must be exactly six characters long, including trailing blanks).

2. Make sure the system time and date have been set with the SETIME command.

3. Make sure the Batch monitor isn't running. If it is running, give the command, BATCH SYSTEM -STOP and wait for it to log out.

4. Since the Batch subsystem needs access to BATCHQ and its sub-UFDs it must know the BATCHQ password. To give it the new password, do the following:

```
OK, ATTACH BATCH <owner-password>
OK, ED B$LIBF
EDIT
FIND C ATCH$B
C ATCH$B, BATCH, JCB
LOCATE BATPAS;*2
       INTEGER BATPAS(3),OPASS(3),NPASS(3),LSTNAM(17)
       DATA BATPAS/'<old-password>'/
CHANGE /<old-password>/<new-password>/
       DATA BATPAS/'<new-password>'/
FILE

OK,
```

Remember that the new password must be exactly six characters long (including trailing blanks).

5. Attach to the BATCH UFD (if you're not there already) and invoke the command file C_BATCH, with the commands:

```
ATTACH BATCH
COMINPUT C_BATCH
```

6. After the C_BATCH command file has finished, attach to the BATCHQ UFD and invoke the C_BDIF command file if you are either building a new Batch subsystem or if you want to reinitialize the BATDEF file. Use the C_RSET command file if you want to preserve the current BATDEF file. The commands are as follows:

```
ATTACH BATCHQ [owner-password]
CLOSE ALL
COMINPUT C_BDIF (or C_RSET)
```

C_BDIF and C_RSET perform the following actions:

- They make sure the Batch monitor is not running. (If the monitor is running, STOP it, and then try the C_BDIF or C_RSET file again.)

- They use FUTIL to TREDELETE CIFILE and Q-CTRL sub-UFDs.

- They recreate the CIFILE and Q-CTRL sub-UFDs, delete all T$xxxx files in BATCHQ, delete the QUEUE and EXECUT files, run *INIT to recreate empty QUEUE and EXECUT files. (C_BDIF also deletes the BATDEF file.)

- C_BDIF then runs BATGEN to generate an empty BATDEF file. C_RSET leaves the BATDEF file intact. (Two non-fatal messages - "File in use. IN.USE" from FUTIL and "Creating new Batch definition file: BATDEF (BATGEN)" from BATGEN - appear during this process. Ignore them.)

- Finally, C_BDIF and C_RSET create a file called "VALID.", which indicates (by existing) that the Batch database is valid. If this file is later deleted, either by accident or by a Batch program designed to signal an invalid database, the Batch subsystem will cease to work. If this happens, run *FIXBAT. If that fails, reinitialize the database by running C_BDIF or C_RSET.

If you are not using passwords, the Batch subsystem is fully installed at this point.

7. If you are using passwords, this is the time to attach to the BATCHQ UFD and set its password to new-password.

8. Now attach down to the CIFILE and Q.CTRL sub-UFDs and set passwords in them. (These sub-UFDs contain the information you least want users to see.) The Batch subsystem does not need to know the passwords of these sub-UFDs, since it can get them (and always does) when it is attached to BATCHQ as an owner.

Note

> Invoking the C_RSET or C_BDIF files will recreate the Q.CTRL and CIFILE sub-UFDs, which resets their passwords to blanks. Whenever these command files are run, the sub-UFDs should be repassworded by the System Administrator.

9. Protect the PH_GO and *FIXBAT files in BATCHQ to "7 1" protection, because the PH_GO is referenced as a non-owner when the Batch monitor is started up, and the first thing it does is run BATCH>*FIXBAT. You may also want to protect BATDEF to "7 1" protection, so that users can monitor queue availability and parameters via the BATGEN -STATUS and -DISPLAY commands.

The BATCH UFD should also be passworded so that users who understand the Batch subsystem (or who have read this document) may not look at the B$LIBF file to determine the password. No software needs to know the BATCH password. CMDNC0 should also have a password, although this is not a requirement.

Note

Once BATCHQ (and/or CMDNC0) have been passworded, the C_BATCH command file will no longer work. Therefore, if you need to rebuild the Batch subsystem after you have passworded either or both directories, you must either remove the passwords while you are rebuilding the subsystem, or you must encode the passwords in the C_BATCH file as follows:

```
OK, ED C_BATCH
EDIT
F TO BATCHQ
TO BATCHQ
A  passwd-1
TO BATCHQ passwd-1
F TO CMDNC0
TO CMDNC0
A  passwd-2
TO CMDNC0 passwd-2
FILE
```

DEFINING AND MODIFYING THE BATCH ENVIRONMENT

Once Batch has been installed, define its environment. You do this by defining from one to six queues via the BATGEN command. This command has the form:

BATGEN pathname

Usually, pathname will be BATCHQ>BATDEF (or, preferably, 'BATCHQ password>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. For an example of this, see the section on Cleaning Up Queues, below.

Once pathname has been read and validated, BATGEN types a prompt character and waits for a BATGEN command. 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. The number is assigned by the Batch system and reflects the order in which queues are first used. To establish a queue as the number-one queue, 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 ($), and waits for a subcommand (described below). If queuename is already in use, it returns a fatal error message, "Queue queuename already exists." ADD subcommands are discussed immediately following this list of commands.
MODIFY queuename	Instructs BATGEN to modify an existing queue. If queue queuename exists, MODIFY responds "Enter queue characteristics:" prints a prompt ($), and waits for subcommands (described below). 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 table.
DELETE {queuename | ALL}	Flags an existing queue (or all queues) for deletion. Queue(s) 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.

ADD and MODIFY Subcommands

Subcommands for the 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. The ADD/MODIFY subcommands function as follows. (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 of his own. 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 who asked for 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 5 seconds for default and maximum CPTIME. Unless both CPTIME limits are explicitly given, they will be set to 5 seconds 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, and its system defaults are both 20 minutes.

▶ FUNIT number

This subcommand sets a default file unit for command input for any 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 vis-a-vis 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. (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 run-time 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 CHAP command; see Secton %CMD%.) 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 for himself.) 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 timeslice value for jobs in the queue. A queue's timeslice may be smaller than the monitor's timeslice and be effective; but if it is larger, it will be ignored, and the monitor's timeslice will be used for each job in the queue. (Again, the user has no control over this value.) Timeslice values represent tenths of a second. They may range from 1 to 99, but probably should not go below 20 unless job priority is unusually high. System default is 20, equaling 2 seconds.

The two commands, below, terminate subcommand sessions and return you to BATGEN command level.

▶ RETURN

Saves the new characteristics for future display and/or filing.

▶ 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."

BATGEN Example

In the following sample BATGEN session two batch queues are defined that might be useful in a university environment. The first, EXPRESS, 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. User input is underlined; comments are offset by '/*'. Except for the illustrative error, all input could come from a command file:

```
OK, batgen BATCHQ>BATDEF
GO
[BATGEN rev 17.0]
> add EXPRESS
Enter queue characteristics:
$ cptime 2          /* 2 seconds max CPU time (default)
$ etime 5           /* 5 minutes max elapsed time (default)
$ priority 4        /* default queue priority is 4
$ return            /* save queue definition
> add EXPRESS       /* wrong queue name
Queue EXPRESS already exists.
> add PAYROLL       /* start new queue definition
Enter queue characteristics:
$ cptime NONE NONE /* no limits
$ etime  NONE NONE
$ funit 126         /* unit 126 for command input
$ priority 9        /* default queue priority is 9
$ rlevel 2          /* lower rlevel by 2 units
$ return
> display           /* display queue definitions

Queue name = EXPRESS, unblocked.
Default cptime=2, etime=5, priority=4;
Maximum cptime=5, etime=20; 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 EXPRESS    /* modify queue definition
Enter queue characteristics:
$ cptime 3          /* set cpu time limit to 3 seconds
$ rlevel 1          /* lower rlevel by 1 unit
$ timeslice 10      /* set timeslice to 10 units
$ return
> display EXPRESS   /* check modification

Queue name = EXPRESS, unblocked.
Default cptime=3, etime=5, priority=4;
Maximum cptime=5, etime=20; Funit=6;
Delta rlevel=1; Timeslice=10;
```

```
> file               /* establish batch queue definitions

OK,
```

USING FIXBAT

FIXBAT (FIX BATch) is an off-line utility designed to:

- Handle the start-up 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 using PH_GO. The System Administrator can choose 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 files 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.)

Invoking FIXBAT

FIXBAT resides as a program, *FIXBAT, in the BATCHQ UFD. To run FIXBAT:

1. Log out the Batch monitor (if it is running).
2. Log into the SYSTEM UFD.
3. Attach to the BATCHQ UFD (giving the owner password if you passworded the UFD).
4. Resume FIXBAT.

If you try to start FIXBAT while the Batch monitor is running, FIXBAT returns with the error message, "Batch process still in progress."

The format for the resume command is:

```
RESUME *FIXBAT [options]
```

There are three options:

Option	Meaning
-DAYS n	Remove all cancelled, completed, or aborted jobs which are n or more days old from the Batch queues. (n must be a positive integer between 1 and 60.)
-QUIET	Do not send a message to the terminal when FIXBAT removes a job from the queue. (This option is useful only if the -DAYS option is also given.)
-STARTUP argument	Tells FIXBAT to start the BATCH monitor. This option should not be specified when FIXBAT is being run interactively, as it will produce the error message "Unable to process BATCH jobs." 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: Rename the current comoutput log "OLDLOG" (deleting any existing "OLDLOG"). Create a new comoutput file named O_LOG. SPOOL: Spool the current comoutput file, calling it BATCH.LOG. Create and open a new O_LOG file. DELETE: Open O_LOG as a comoutput file. (The file is truncated when it is opened, destroying the existing contents.) NOLOG: Take no action with regard to comoutput files.

The O_LOG file, when generated, contains an identifying first line, suitable as a header line for a spool file. It contains the time of day, the date (and the day), and the FIXBAT revision number. After that comes two blank lines, then comes the log trail of what FIXBAT did.

This may include "comments" like "Fixing database", "Deleted T$0001", "Deleted C00041", etc., which are not errors, simply notifications that certain files deemed useless by FIXBAT were deleted.

It may also include the information on any jobs that were deleted from the queues (unless -QUIET was specified). In addition, any strange file formats (such as partial queue entries) are noted here.

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.

After FIXBAT runs, the PH_GO file will then RESUME *MONITOR and the monitor rev number will be typed out, followed by a log trail of its activities.

One thing to note is that most of FIXBAT's output when -STARTUP is specified, is preceded by the time of day on the line. This is useful for determining the timing of operations related to a user complaint that "something went wrong". For example, running FIXBAT interactively might result in the line:

```
Deleted T$0000
```

being typed out, whereas if FIXBAT had been running from PH_GO, the line might have looked like:

```
06:43:52 Deleted T$0000
```

Note

> The DELETE and NOLOG arguments to the -STARTUP option are not recommended, as they make it difficult for the System Administrator to analyze any Batch problems that may occur. However, when problems do occur, the Batch subsystem will try to create a file named "ERROR." in the BATCHQ UFD giving some information on the error.
>
> Changing the argument to -STARTUP may result in old comoutput files (OLDLOG or O_LOG), being left around, or recent log files being deleted. It is recommended that whenever a change like this is made, that the System Administrator make sure that any existing files are taken care of before the Batch monitor is started up with the new PH_GO file.

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 in PH_GO), however, it fixes the database only if one (or more) of three conditions is met:

- It fixes the database if -DAYS n has been specified, in order to remove old jobs from the queue.
- It fixes the database if it cannot find the "VALID." file in the BATCHQ UFD. (The absence of this file indicates an invalid database.)

- It fixes the 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.)

How FIXBAT works: The course of FIXBAT's activities is as follows:

1. FIXBAT begins by deleting temporary files out of BATCHQ. Temporary files are defined as files with six-character names beginning with "T$" and with the last four characters successfully converting to decimal numbers using CNVA$A. (See Subroutine Reference Guide for a description of the APPLIB CNVA$A routine.) These files are generated by the Batch monitor to "bootstrap" Batch jobs, including attaching to the home UFD. If that attach fails, the temporary file stays around. Usually, the Batch monitor deletes these files itself. When a temporary file is deleted, the message "Deleted <name>" is output.

If FIXBAT decides to fix the database, it prints the message "Fixing database," and takes two further actions:

2. It protects all command files in CIFILE to "7 0" protection. A command file is defined here as a file with a six-character name beginning with "C", where the second character is between "0" and "9" or "A" and "V" (inclusive), and with the last four characters successfully converting to a decimal number using CNVA$A. Between this step and step 3, FIXBAT will go through the Q.CTRL queue files, and whenever it finds an active job, it will attach back to BATCHQ>CIFILE and protect the command file of the job to "1 0" protection.

3. It deletes all command files in CIFILE, ignoring "Insufficient access rights" errors, and printing a "Deleted <name>" message if successful. This results in all command files not belonging to active jobs being deleted.

Deleting the Old Batch Job Entries: When FIXBAT deletes old Batch job entries from the queue files, it physically removes the job entry from the queue, and writes the next job entry over the deleted one, repeating this procedure (in effect filling up the hole made), until the end of the queue file is reached.

It will only perform this operation, however, if a -DAYS argument was specified on the command line.

The mechanism 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. The job must have completed, aborted, or been cancelled in the

same or previous year as the current year.

3. The job must have completed on a date such that there are at least <n> full days between that date and the current date, non-inclusive. This means that if a job completed on April 10, 1979, and the current date is April 12, 1979, 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 outputs the final information on that job in a similar format to the information returned by a "JOB -DISPLAY" command, unless the -QUIET option was specified on the command line.

Note

If a deleted job is displayed, the queue name may be blank. This occurs if the user did not explicitly specify a queue. Also, the queue name may not resemble the queue name as defined in BATGEN with regard to upper/lower-case mapping. Examples: for queue "COBOL", a deleted job may have "(queue COBOL)", "(queue cobol)" or "(queue)" output.

FIXBAT error messages and responses

While FIXBAT is running, it may output 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, it 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 command file C_RSET, or, as a last resort, C_BDIF should be invoked. These files are described earlier in this section.

Cleaning Up Queues

Each Batch queue can hold 10,000 jobs. When it is filled, it types error messages ("Queue full") to users submitting jobs. The System Administrator must then delete and redefine the queue so that new jobs may be submitted to it. (Although FIXBAT can delete inactive jobs, it does not reclaim job numbers. When the number 10,000 comes up, the queue is "full," no matter how many jobs it actually contains.

There are three ways to remove jobs from queues:

1. Run the C_BDIF or C_RSET command files, as explained in Installing the Batch Subsystem. This is the fastest way to work, as it will empty all queues. (Running C_BDIF will also wipe out BATDEF.) Be warned, however, that 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 C_BDIF or C_RSET.

2. Delete one queue at a time, using the BATGEN "DELETE queue-name" command. This method assures 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 filling job submissions. Users attempting to submit jobs to the queue are give 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 completed or aborted is the queue actually deleted. Then its database is deleted, the queue is removed from the BATDEF file, and a 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 runs "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 different from or identical to the old one - may be placed in the BATDEF file. (If BATDEF originally contained less than six queues, the new queue could be added before the old one was 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, wait for the monitor to remove the queue from BATDEF, and then ADD.)

3. 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.) ADD also any new queues you wish to create.

   3. FILE your new BATDEF file with the BATGEN command:

      FILE 'BATCHQ password>BATGEN'

   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.

SECTION 13

SYSTEM RESOURCE ALLOCATION

This section is an outline of the contents planned for future versions of this PDR. Where available, references have been made to other sections of this guide or to other documents where the information is available:

NUMBER OF USERS
- Terminal
- Remote
- Phantom

USER SEGMENTS
SHARED SUBSYSTEMS
COMMAND UFD (CMDNC0)
PARTITIONING DISKS

BATCH VS. CX

The decision to implement Batch, CX, or both, belongs to the System Administrator. We recommend implementing Batch alone, for the following reasons:

- Implementing Batch and CX together creates a security problem. Batch runs command files under their users' login names, and reserves the name SYSTEM for the operator or administrator. CX runs all jobs as SYSTEM. A CX phantom file, logged in as SYSTEM, would therefore have access to all Batch users' jobs and to the Batch database and could cause damage to the Batch subsystem.

- Batch offers considerably more flexibility to users, operators, and administrators, than CX does. Major differences are as follows:

	BATCH		CX
1.	Batch runs any non-interactive command file, including command input, phantom, and CX files.	1.	CX runs only those non-interactive command files which begin with an ATTACH command and end with CO -TTY, CX -E, or LOGOUT.
2.	Each job runs under its user's login name, allowing easier identification of jobs and giving information on system usage.	2.	All jobs run as SYSTEM.
3.	The home UFD may be specified when the job is submitted, or changed while the job is waiting in the queue.	3.	The home UFD must be specified within the command file; it can be changed only by editing the file.
4.	The System Administrator can define the environment for each queue, allotting high run-time priorities and short timeslices to some queues, low priorities and long timeslices to others. The System Administrator can also set maximum and default CPU time limits and elapsed time limits to be applied to jobs within a queue.	4.	The System Administrator has no control over the CX environment.
5.	The operator can monitor Batch jobs and queues, force-logout or restart executing jobs, cancel jobs waiting to run, and hold jobs within the queue, keeping them active but preventing them from running until he releases them.	5.	The operator can force-logout executing jobs.

6. Users may set CPU limits and elapsed time limits on their own jobs, within the framework set up by the Administrator. They may also set a value for their jobs' priorities within the queue.	6. Users may set CPU time limits and queueing priorities for their jobs
7. Users may change parameters on their jobs while the jobs wait in the queue. They may also cancel waiting jobs, and abort or restart executing jobs.	7. Users may drop jobs from the queue.

SECTION 14

DEFAULTS AND PARAMETERS

This section is an outline of the contents planned for future versions of this PDR. Certain information is presented in its entirety here. Where available, references have been made to other sections of this guide or to other documents where the information is available.

TRANSLATORS

Translators process source programs into object code capable of being loaded into a run-time memory image by one of Prime's loaders. The translators also perform other related operations (error message printing, concordance generation, etc.) governed by command line keywords.

The FORTRAN 77 and PL/I subset G compilers have their defaults set with driver programs. These programs are supplied in the UFDs F77>TOOLS and PL1G>TOOLS on the Master Disk. The default information is stored in a data file supplied in UFD SYSOVL. The Administrator can move these programs and/or the data files to other directories if desired. If this is done, make sure that:

- The directory in which the driver programs are stored should be password-protected and the driver programs set to allow non-owners NO rights. This prevents unauthorized execution to change defaults.

- The data file protection should be set to allow non-owners both READ and WRITE access. These files are in a non-ASCII format for security reasons. They are modified by the driver programs.

To change defaults, use the following sequence of commands:

```
A directory owner-password
R translatorDF driver-pathname -options
```

directory	UFD in which the driver programs are resident. Supplied as TOOLS.
owner-password	Owner password of directory.
translator	Name of the compiler - F77 for FORTRAN 77, PL1G for PL/I subset G.
driver-pathname	Pathname of the datafile. Supplied in UFD SYSOVL. The file is F77DATA for FORTRAN 77 and PL1GDATA for PL/I subset G.

options New default options for the compiler.

If the driver is resumed without the driver-pathname and options, it prints instructions and a list of valid keyword options.

PL/I subset G (PL1G)

The driver program is PL1G>TOOLS>PL1GDF; the data file is SYSOVL>PL1GDATA. If the program is executed with no arguments, the result is as below:

```
OK, RESUME PL1GDF

Usage: PL1GDF driverfilename [options]
    where "options" are selected from the following:

-BINARY      -LISTING    -NESTING  -OPTIMIZE  -XREF    -EXPLIST
-OFFSET      -BIG        -SILENT   -RANGE     -UPCASE  -DEBUG
-STATISTICS  -PRODUCTION -32I
```

See The PL/I subset G Reference Guide for the details of the compiler options.

FORTRAN 77 (F77)

The driver program is F77>TOOLS>F77DF; the data file is SYSOVL>F77DATA. If the program is executed with no arguments, the result is as below:

```
OK, RESUME F77DF

Usage: F77DF driverfilename [options]
    where "options" are selected from the following:

-BINARY-LISTING-NESTING-OPTIMIZE-XREF-EXPLIST
-OFFSET-BIG-INTS-DYNM-SILENT-RANGE
-UPCASE-DEBUG-STATISTICS-DO1-PRODUCTION-LOGS
```

See the FORTRAN 77 Reference Guide for the details of the compiler options.

In the FORTRAN, COBOL, RPG II, and PMA translators these keywords set the values of the A and B registers which control the operations actually performed. To change the defaults, use the following sequence of commands:

```
A CMDNC0 owner-password
RESTOR translator
SAVE translator [3/A-register] [4/B-register]
```

translator — The translator utility: FTN, COBOL, RPG, or PMA.

A-register — New value of the A register. If omitted, the current value is unchanged.

B-register — New value of the B register. If omitted, the current value is unchanged.

FORTRAN (FTN)

The Prime-supplied FORTRAN compiler defaults are:

A register: '1707

Input file on disk (-INPUT pathname)
No listing file (-LISTING NO)
Binary file on disk (-BINARY YES)
Print error messages at user terminal (-ERRTTY)
Generate 32R mode code (-32R)
No global trace (-NOTRACE)

B register: 0

Static allocation of local variables (-SAVE)
Short integers (-INTS)
No concordances (-NOXREF)
Generate floating point skip instructions (-FP)
Do not generate code for segment-spanning arrays (-NOBIG)
Do not optimize DO loops (-STDOPT)
Do not generate code for debugger (-NODEBUG)

See The FORTRAN Reference Guide for values of the A and B registers.

COBOL

The Prime-supplied COBOL compiler defaults are:

A register: '2777

Input file on disk (-INPUT pathname)
Listing file on disk (-LISTING YES)
Binary file on disk (-BINARY YES)
Generate 64V mode code (-64V)
Suppress expanded listing (-NOEXPLIST)

See The COBOL Reference Guide for values of the A register; the B register is not used.

RPG II

The Prime-supplied RPG II compiler defaults are:

A register: '3777

Input file on disk (-INPUT pathname)
Listing file on disk (-LISTING YES)
Binary file on disk (-BINARY YES)
Print error messages at terminal (-ERRTTY)
Print concordances (-XREF)
Print current status of compilation (-STATUS)
Suppress column index banner (-NOBANNER)
Suppress source program sequence checking (-NOSEQCHK)
Suppress octal listing of object data (-NOOBDATA)

See The RPG II Programmer's Guide for values of the A register; the B register is not used.

Note

The RPG II compiler interprets an A-register setting of 0 as '3777 (the default setting).

Assembler (PMA)

The Prime-supplied Assembler defaults are:

A register: '0777

Input file on disk (-INPUT pathname)
Listing file on disk (-LISTING YES)
Binary file on disk (-BINARY YES)
Print errors at terminal (-ERRLIST)
Suppress expanded listing (-NOEXPLIST)
Generate complete concordance (-XREFL)

See The Assembly Language Programmer's Guide for values of the A register; the B register is not used.

LOADERS

See the Reference Guide, LOAD and SEG for complete details of Prime's loaders.

V-Identity (SEG)

Stack size: '6000 words
Default library: PFTNLB and IFTNLB (FORTRAN libraries)

R-Identity (LOAD)

Memory location: '122770 to '144000
Default library: FTNLIB (FORTRAN library)
Mode: D32R
Sector Zero Base Area:
  Base start at location '200
  Base range '600 words
COMMON: Top at location '077777

SYSTEM AND NETWORK PARAMETERS

Configuration Defaults

Default values for many parameters are established by the operating system upon start up. These value can be altered by including the appropriate directive in the configuration file. These parameters, along with their defaults and the directive to alter these defaults are given in the table below. (See Section 17 for details).

Parameter	Default	CONFIG Directive
ABBREV processor	YES	ABBREV
AMLC line input buffer	'140	AMLBUF
AMLC line output buffer	'300	AMLBUF
AMLC programmable clock baudrate	9600	AMLCLK
ASR terminal input buffer	'200	ASRBUF
ASR terminal output buffer	'300	ASRBUF
Carrier check operations interval	2	AMLTIM
Configure network	NO	NET ON
DMQ AMLC buffer	'40	AMLBUF
Event logger file size	'10000	LOGREC
File system read/write lock	1	RWLOCK
Implicit logouts allowed	YES	LOGLOG
Inactivity timeout (seconds)	'1750	LOUTQM
Logout on AMLC line disconnect	NO	DISLOG
Max. per-user guaranteed file units	'176	FILUNT
Maximum per-user file units	'200	FILUNT
Min. grace time for terminal lines	0	AMLTIM
Modem disconnect operations rate	1800	AMLTIM
Number of prepaged pages	3	PREPAG
Phantom users, number	0	NPUSR
Print configuration directives	NO	TYPOUT
Print LOGIN/LOGOUT messages	YES	LOGMSG
Remote users, number	0	NRUSR
Restart after power failure	NO	UPS
Segments per user process	'40	NUSEG
SMLC lines	OFF	SMLC
Supervisor terminal baud rate	'100	ASRATE
System erase character	"	ERASE
System kill character	?	KILL
Total available file units (all users)	'4000	FILUNT
Total virtual address space (segments)	'300	NSEG
Wired memory size printout	NO	WIRMEM

SECTION 15

USAGE ENHANCEMENTS

This section is an outline of the contents planned for future versions of this PDR. Where this information is available, references have been made to other sections of this guide or to other documents.

EXTERNAL LOGIN PROGRAM

Overview

An external login program is a program that is used to augment the LOGIN facility provided by PRIMOS. When a user of the system LOGS IN or LOGS OUT, any existing program LOGIN in UFD CMDNC0 is RESUMEd. This program may be custom-written by a given installation to perform special LOGIN/LOGOUT functions, such as accounting or restricting system access.

Logging In

To interact with a Prime computer system, a user must be identified to PRIMOS by logging in. To do this, the LOGIN command must be issued with the User File Directory (UFD) name and the user's password (if a password has been assigned). For example,

```
LOGIN JDOAKS SECRET
```

In this example, LOGIN is the command name, JDOAKS is the UFD name and SECRET is the password.

Upon receiving the LOGIN command line, the PRIMOS LOGIN facility performs the following operations:

1. PRIMOS on the line. The LOGIN facility looks at the switch LOGOVR, which is configured at PRIMOS system start-up time (by the LOGLOG directive). If LOGOVR is set, the previous user is required to LOGOUT before the present user is allowed to LOGIN. If the LOGOVR switch is not set, the previous user is immediately logged out and the pending LOGIN command proceeds.

2. PRIMOS initializes accounting time meters (e.g., Elapsed Time, CPU Time, I/O Time).

3. PRIMOS attempts to attach the user to the specified UFD. If the ATTACH is successful, the user is logged into the system. If the ATTACH is not successful for any reason (e.g., UFD NOT FOUND, NO RIGHT, BAD PASSWORD, etc.), the LOGIN process terminates and control returns to a state where PRIMOS is ready to receive a subsequent LOGIN.

4. PRIMOS sends LOGIN messages to the user terminal and to the supervisor terminal.

5. PRIMOS runs an external login program in UFD=CMDNC0 (if an external login program exists).

Logging Out

A user notifies PRIMOS of the finish of an interactive session by issuing the LOGOUT command. Upon receiving the LOGOUT command, PRIMOS takes the following actions:

1. Sends a message to the user terminal and the supervisor terminal.

2. Closes all open file units.

3. Unassigns storage devices.

4. Unassigns AMLC lines.

5. Resumes the external LOGOUT program.

6. Cleans up after resuming the external LOGOUT program.

Writing an External Login Program

Purpose: Different site environments may require operations and functions not provided by the standard PRIMOS LOGIN and LOGOUT facility. An external login program provides the system administrator the capability to augment operations on a per-site basis, without having to modify any PRIMOS internal code.

Examples of External LOGIN Usage: Some suggested tasks that an external login program can perform are:

- Validation of user identity, project, password, etc.

- Validation at LOGIN time for users logging in via telephone (e.g., passwords, restricted lines, etc.).

- Monitoring usage of valuable system resources (e.g., Connect Time, CPU Time, I/O Time).

- Printing system status information at the user terminal.

- Providing additional user services. For example, executing programs or command files to initialize the users process, such as setting the terminal characteristics.

External LOGIN Programming Guidelines

The external login program may regulate the use of the operating system in addition to the LOGIN facility provided by PRIMOS. It may access confidential system information such as the valid user names, passwords, and per-user accounting information. Therefore precautions must be made when writing the external login program, to prevent inadvertent or malicious misuse of the program. The following are factors to be considered by a systems administrator when writing an external login program.

- To validate user identity, project, password, the external login program must ATTACH to a directory that contains the names of all users on the system at the specific site. This directory must also contain all associated validation information. Access to this directory must require passwords; which appear as either data or parameters in the external login program. If a user has access to the external login program, access may be gained to the validation information of the entire system. Therefore, the external login program must be placed in a memory segment that is protected by the system ring mechanism, or in a directory that requires both owner and nonowner passwords. The passwords should be known only to the system administrator and designated key personnel.

  For example, the external login program may reside in the UFD named CMDNC0 where the owner has read, write, and delete rights to the external login program; the nonowner has no rights at all.

- User passwords are read from the system directory to perform user validation. This information is stored within some locations in memory. If a user is logged in, these memory can be accessed using a system debugger (such as VPSD). To prevent unauthorized access to another users validation information, the external login program must destroy this information before completion of execution. This must occur as soon as the validation data is no longer needed by the program.

- All files that are opened by the external login program must be closed before the program completes execution.

- CONTROL-P for interrupting program (BREAK) is inhibited when the external login program begins execution; it must be re-enabled when the external login program completes execution. When a user issues the LOGIN command, PRIMOS attempts to ATTACH the user to the specified UFD. If this is successful, then the external login program is resumed. If the user is allowed to QUIT in the middle of execution of the external login program, the following undesirable conditions result:

Files left open.

User validation information in memory where the user can access it.

An incomplete validation process i.e., the user is logged in without having gone through the validation checks.

- The home UFD must not be changed by calls to the PRIMOS subroutine ATCH$$ in the external login program. Before this program terminates, it must reset the current UFD to the home UFD. The calling sequence is:

  CALL ATCH$$(K$HOME,0,0,0,0,CODE)

- If user input from the user terminal is required during the LOGIN process, a timeout condition must be set to avoid an indefinite wait for user response. For example, if a password is required for a login, the user is given a finite number of chances to enter the correct password. If the correct password is not given in this number of trials, the external login program should log that user out.

- The external login program may forcibly log a user out if the user does not pass the LOGIN validation process. This is done by calling the LOGO$$ subroutine (see PRIMOS Subroutines Reference Guide for usage details).

- The The external login program is designed by the system administrator to meet the needs of the site. Site environments and requirements change from time to time; the external login program must be easy to modify and to understand.

The program is RESUMEd with BREAK inhibited to prevent the user from exiting LOGIN via CONTROL-P or BREAK. The LOGIN program should perform a CALL BREAK$(.TRUE.) before exiting.

Note

The BREAK$ subroutine maintains a BREAK inhibit counter, rather than a single flag. External login programs (which are entered with BREAKs inhibited) should enable breaks before calling EXIT. That is, they should always enable BREAKs one time more than the number of inhibits; otherwise the user will be logged in with BREAKs inhibited. (If this happens, causing any error will re-enable BREAKs.)

The command line that called LOGIN is available via the library subroutines RDTK$$ and CMREAD. Accounting information is available via the library subroutine TIMDAT. The LOGIN program exits via a call to the library subroutine EXIT and must not encounter any uncontrolled errors that result in EXIT being bypassed.

Sample External Login Program

The program below allows the System Administrator to designate which UFDs cannot be logged into. It also records each user's TIMDAT information for for accounting purposes. Whenever the password of UFD=CMDNC0 is changed the data line:

```
DATA CMDPAS/'PASSWD'/
```

must be changed also.

The program can be compiled, loaded, and installed using the command file below:

```
FTN LOGIN
LOAD
LO B_LOGIN
LI
SA *LOGIN
QU
FUTIL
TO CMDNC0 owner-password        /* Put the owner-password here
C *LOGIN LOGIN
Q
```

```
C---Sample External Login Program.    Barry Burke   11/01/78
C---                                  Business Data Services
C---                                  PRIME Computer, Inc.
C
C SYSCOM>KEYS.F        MNEMONIC KEYS FOR FILE SYSTEM (FTN)    31 MAY, 1977
      NOLIST

      INTEGER*2         LOGDAT(15), INFO(8), BUFF(20), CODE, TYPE, NWR,
     +                  CMDPAS (3), I,J,LOGOUT

      LOGICAL BRKON,BRKOFF,FANTOM,LOGIN,NAMEQ$

      PARAMETER         BRKON=.FALSE.,BRKOFF=.TRUE.

      DATA     CMDPAS/'PASSWD'/          /* YOUR 'CMDNCO' PASSWORD

C
C---START.
C--- INHIBIT BREAKS (CONTROL 'P') ONCE MORE TO INSURE SECURITY.
C
            CALL BREAK$(BRKOFF)
            LOGOUT = $60

C
C--- NOW GET SOME VITAL INFO FROM USERS COMAND LINE.
C
      CALL RDTK$$(3,INFO,BUFF,20,CODE)        /* RESET COMMAND LINE POINTER
      CALL RDTK$$(K$READ,INFO,BUFF,20,CODE) /* READ 1st TOKEN

      LOGIN = (NAMEQ$('LOGI',4,BUFF,4))      /* CHECK FOR LOGIN/LOGOUT

C
C--- NOW CHECK IF THIS A PHANTOM USER.
C--- PHANTOM USERS WILL ONLY HAVE THE WORD 'LOGIN' IN THEIR COMMAND
C--- LINE ON LOGIN.
C
      IF (LOGIN) CALL RDTK$$(K$READ,INFO,BUFF,20,CODE)   /*READ NEXT TOKEN
      FANTOM = .FALSE.
          IF (INFO(2) .EQ. 0)     FANTOM=.TRUE.

C
C--- NOW GET SYSTEM DATA FOR THIS USER (FROM TIMDAT).
C
        CALL TIMDAT(LOGDAT,15)

C
C--- HERE WE ATTACH TO 'CMDNCO' TO DO SOME SECURITY CHECKING.
C--- IN THIS UFD WE KEEP THE FOLLOWING FILES:
C---                    LOGOUT = COMMAND FILE WITH THE WORD 'LOGOUT' IN IT.
C---                    BADONE = FILE WITH UFD-NAMES *NOT* ALLOWED TO LOGIN TO.
C---                    LINPAS = FILE WITH LINE #'S AND PASSWORDS FOR EACH.
C---                    LOGFIL = FILE IN WHICH IS STORED EACH USER'S LOGIN
C---                             LOGOUT TIMDAT INFO.  THIS IS USED AS THE DATA
C---                             FOR A WEEKLY ACCOUNTING PACKAGE.
```

```
C
C
C
C--- ATTACH TO CMDNCO
C
      CALL ATCH$$('CMDNCO',6,K$ALLD,CMDPAS,K$IMFD,CODE)    /* DO *NOT* SET HOME.
      IF (CODE.NE.0) GOTO 10       /* ON ERROR GIVE USER ERROR MESSAGE
      CALL ERRPR$(K$IRIN,CODE,'External Login',14,'ATCH$$',6)
      GO TO LOGOUT           /* AND LOG HIM OUT (HE WILL CALL)
C
C--- SET UP ENTRY FOR LOGFIL.
C
10    LOGDAT(16) = 'LO'                /* LOGOUT KEY
      IF (LOGIN)  LOGDAT(16) = 'LI'    /* LOGIN

C
C--- NOW OPEN FILE.
C---        SUBROUTINE 'OPENF' IS ACTUALLY A CALL TO SRCH$$, BUT INSTEAD
C---           OF SUPPLYING A 'TYPE' VAR., A NUMBER SPECIFYING THE NUMBER OF
C---           RETRIES TO ATTEMPT ON A FILE-SYSTEM ERROR. THE MAIN REASON
C---           FOR DOING THIS IS BECAUSE MORE THAN ONE USER MAY BE ATTEMPTING
C---           TO WRITE TO THE FILE AT THE SAME TIME, RESULTING IN 'FILE IN USE'
C---           ERROR
C
         CALL OPENF(K$RDWR,'LOGFIL',6,1,5,CODE)
         IF(CODE.NE.0) GOTO 30         /* ON ERROR, SKIP THIS

C
C--- POSITION TO THE END OF THE FILE.
C
      CALL PRWF$$(K$POSN+K$PREA,1,LOC(0),0,1000000,NWR,CODE)
C
C--- AND WRITE ENTRY
C
      WRITE(5,1000) LOGDAT
1000  FORMAT(3A2,8I6,I2,4A2)
      CALL SRCH$$(K$CLOS,0,0,1,0,CODE)      /* CLOSE THE FILE.

30    IF (LOGIN) GOTO 35
      CALL EXIT              /* WAS A LOGOUT, LET HIM GO.
      GOTO 999               /* TO KEEP HIM FROM GETTING BACK IN

C
C--- NOW FOR SOME USER - VALIDATION.
C
C--- 1)     IS LINE PASSWORDED?
C
35     CALL OPENF(K$READ,'LINPAS',6,1,5,CODE)
       IF(CODE.EQ.0) GOTO 40

       CALL TNOU('Ext. Login - No password file!',30)  /* SOME-BODY STOLE IT!
       GOTO LOGOUT           /* SECURITY HAS BEEN BREACHED...NOBODY GETS IN!

C
```

```
C--- FILE IS OK.
C
C
C--- THE FILE 'LINPAS' HAS ENTRIES WHICH LOOK LIKE THIS:
C
C    ##, PASSWD
C    01, GORK
C    17, FOO
C      .
C      .
C      .
C
40    CALL RDLIN$(1,BUFF,20,CODE)
      IF(CODE.NE.0) GOTO 70

      IF((RS(BUFF(1),8)-:260)*10+(RT(BUFF(1),8)-:260).NE.LOGDAT(12))
     +  CALL SRCH$$(K$CLOS,0,0,1,0,CODE)

C
C--- HERE USER # IS PASSWORDED....VERIFY WITH USER.
C
      DO 50 I=1,3
      CALL TONL
      CALL TNOUA('Password?  ',11)

      CALL COMANL
      CALL RDTK$$(K$READ,INFO,BUFF(6),3,CODE)
      IF(NAMEQ$(BUFF(3),6,BUFF(6),6))  GOTO 70   /* MATCHED.
      IF (I.EQ.3) GOTO 55
50    CALL TNOU('Password INVALID, try again.',28)
55    CALL TONL
      CALL TNOU('ACCESS PROHIBITED',17)

C
C--- LOG THE USER OUT, HE IS NOT AUTHORIZED TO USE THIS LINE.
C
60    CALL LOGO$$(0,0,0,0,0000000,CODE)   /*BYE-BYE

C
C--- 2)     DO WE ALLOW THIS UFD TO BE LOGGED INTO?
C
70    CALL SRCH$$(K$CLOS,0,0,1,0,CODE)
      CALL OPENF(K$READ,'BADONE',6,1,5,CODE)
      IF(CODE.NE.0) GOTO 100     /* THIS IS AN OPTIONAL FILE

80    CALL RDLIN$(1,BUFF,20,CODE)
      IF(CODE.NE.0)  GOTO 100       /* E-O-F
      IF(NAMEQ$(LOGDAT(13),6,BUFF,6)) GOTO 90 /*MATCHED...NO-LOGIN TO THIS UFD
      GOTO 80

90    CALL TNOU('Not a valid LOGIN name....',26)

         GOTO 55
```

```
C
C--- USER PASSED VALIDATION TESTS, WELCOME HIM IN:
C
100   CALL SRCH$$(K$CLOS,0,0,1,0,CODE)
      CALL TNOU ('Welcome to PRIME Computer.',26)

C
C--- NOW GET USER BACK  TO HIS HOME UFD.
C---  WE CHECK HERE FOR THE EXISTENCE OF A 'C_LOGIN' FILE.  IF PRESENT
C---  WE WILL START IT UP AS A COMMAND FILE.
C---  WE ALSO CHECK FOR A '*LOGIN' PROGRAM,AND WE'LL RUN THIS FOR HIM
C---  IF FOUND.
C
C***  NOTES:
C***    1)  PHANTOMS CAN'T USE THIS FEATURE, AS THEY ARE ALREADY RUNNING
C***        FROM A COMMAND FILE, AND IT WOULDN'T MAKE SENSE TO START UP A
C***        PROGRAM AUTOMATICALLY FOR THEM.
C
C***    2)  WHEN CONTROL IS PASSED TO ONE OR THE OTHER (OR BOTH) OF THESE
C***        FILES, BREAKS WILL STILL BE INHIBITED ONCE. A CALL TO 'BREAK$'
C***        BY THE *LOGIN PROGRAM, OR THE COMMAND 'TERM -BREAK ON' BY
C***        THE 'C_LOGIN' FILE IS REQUIRED TO RE-ENABLE THESE.
C
      CALL ATCH$$(K$HOME,0,0,0,0,CODE)
      CALL BREAK$(BRKON)
      CALL SRCH$$(K$CLOS,0,0,1,0,CODE)

      IF(FANTOM) GOTO 900
      CALL COMI$$('C_LOGIN',7,6,CODE)
      IF(CODE.NE.0)  CALL SRCH$$(K$CLOS,0,0,6,0,CODE)
      CALL SRCH$$(K$EXST,'*LOGIN',6,1,TYPE,CODE)
      IF(CODE.EQ.0) CALL RESU$$('*LOGIN',6)

900   CALL BREAK$(BRKON)
999   CALL  EXIT
      END
C      SUBROUTINE OPENF
C      FUNCTIONS EXACTLY AS SRCH$$ BUT HAS AN
C      EXTRA PARAMETER (NUM) DEFINING THE NUMBER OF ATTEMPTS
C      AT AN OPERATION BEFORE GIVING UP
C
C
      SUBROUTINE OPENF(KEY,NAME,NAMLEN,UNIT,NUM,CODE)
      INTEGER KEY,NAME(16),UNIT,TYPE,CODE,I,NUM
C
C SYSCOM>ERRD.F       MNENOMIC CODES FOR FILE SYSTEM (FTN)     6 SEPT, 1977
      NOLIST
C
      DO 20 I=1,NUM
      CALL SRCH$$(KEY,NAME,NAMLEN,UNIT,TYPE,CODE)
      IF (CODE.EQ.E$FIUS) GOTO 10
      RETURN
10    CALL SLEEP$(0001010)     /* SLEEP FOR 1.01 SECONDS
20    CONTINUE
```

```
RETURN
END
```

MANAGING DISK STORAGE

Monitoring for Charges with FIXRAT, FUTIL

FIXRAT - see Section 5 for operation
FUTIL - see Reference Guide, PRIMOS Commands, FDR3108

What to do When Partition is Full

Looking at partitions - Section 4
Deleting files - Section 3
Creating new partitions - Section 6

CHANGING CONFIGURATION WHEN ADDING MEMORY

RUNNING SPOOLER ACROSS THE NETWORK

PART IV - REFERENCE

SECTION 16

PRIMOS COMMANDS FOR THE OPERATOR

This section describes commands that are normally used by the operator at the PRIMOS supervisor terminal. Some of these commands may also be available for use under PRIMOS II; if so, this is indicated in the command description.

PRIMOS INTERNAL COMMANDS

After PRIMOS is started the PRIMOS II terminal becomes the PRIMOS supervisor terminal. Normally, the supervisor terminal is used only to STARTUP and SHUTDN disks, configure devices, check status, and to collect a record of LOGIN, LOGOUT, and other messages. External commands and RESTOR, RESUME, and START may be invoked from the supervisor terminal under PRIMOS.

Caution

The supervisor terminal must remain attached to CMDNC0 in order that external commands may be invoked from user terminals.

► ADDISK [PROTECT] pdev-1 [pdev-2] ... [pdev-n]

ADDISK searches the table of logical devices for an available table entry location; if one is found, ADDISK starts up the device specified by pdev. ADDISK repeats this procedure for each argument specified. If a device specified by pdev is already started, no action is taken. If there are no free logical devices, the error message:

NO ROOM

is printed at the supervisor terminal. Logical device numbers may be freed by using SHUTDN to remove devices from the table. If a non-existent 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).

PRIMOS supports 62 logical devices; since STARTUP will start up at most eight logical devices, ADDISK must be used to start up devices in excess of eight.

Write protection: The option, PROTECT, is specified to assign write protect for pdev.

PROTECT may only be specified for disks that are added locally via the ADDISK or STARTUP command.

The status of the write protection assignments may be changed for a given pdev by respecifying the ADDISK or STARTUP command with or without the PROTECT option.

If a subsequent ADDISK command is issued for the same disk, and PROTECT is not specified, the write-protect feature is disabled. (An ADDISK PROTECT to an already protected disk does not change the 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.

Networks: If a local computer system is part of a network, devices on other systems on the network may be made available to users on the local system by the command:

ADDISK nodename pdev-1 [pdev-2]...

nodename — The network name of the system on which the physical devices are located.

pdev-n — The physical device numbers of the device on the remote system which are to be made available to local users. These devices must be connected to their local system by STARTUP or ADDISK before being made available to any other system.

Remotely added devices have the write-protection status assigned them at their local system. The write-protection status of a device cannot be changed remotely.

▶ AMLC [protocol] line [configuration] [lword]

The following protocols are available:

- TTY (normal 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 lower case alphabetic characters to upper case for output; uses normal terminal protocol for input)
- TTYHUP (translates lower case alphabetic characters to upper case for high-speed output; uses normal terminal protocol for input)

Terminal protocol is used by lines controlling interactive terminals. With terminal protocol, all input from the terminal is echoed if the line is set for full duplex, and a CARRIAGE RETURN - LINE FEED is echoed following CARRIAGE RETURN. Bit 8 of each character (the ASCII code parity bit) is forced on. CONTROL-P or BREAK are interpreted as a QUIT if the terminal is connected to PRIMOS as a user terminal. If connected to PRIMOS as an assigned AMLC line, CONTROL-P or BREAK are not interpreted. A CARRIAGE RETURN input by the terminal is transmitted as a NEW LINE to the program requesting input. Input is no longer echoed if the line input buffer becomes full. Terminal protocol is identical to the protocol used to control PRIMOS user terminals.

If the System Administrator has enabled the watchdog disconnect timer (via the AMLTIM config directive), users at terminals may have a limited time to log in after their carrier becomes active. If they do not log in within this time, their lines are disconnected.

Transparent protocol is used by lines connected to peripheral devices or other computers. With transparent protocol, no input is echoed, no response is made to the input of a line feed or carriage return, and there is no transformation of carriage return to line feed.

The high-speed protocols (TTYHS and TRANHS) are used by lines connected to peripheral devices that can run at greater than standard terminal speeds, 9600 baud at the terminal. These protocols are the same as those described above with one exception: for output only, the line's character time interrupt flag is set when the output buffer contains more than 40 characters, and it remains set until the output buffer contains fewer than 40 characters. The protocols have a burst mode effect on the output device.

The translating protocols, TTYUPC and TTYHUP, are used to avoid sending lower-case output to terminals or peripheral devices that cannot print lower-case characters. TTYUPC is used for standard-speed lines, TTYHUP for high-speed lines.

With a line using the high-speed protocols, a drastic increase in system overhead can result depending upon the Baud rate and the number of lines in the group. The user must be careful not to assign protocols to lines that normally have their character-time-interrupt flag always set; as, for example, the last line in each group of lines. If the protocol is not given, the transparent protocol is assigned by the operating system. The line number is an octal integer less than or equal to the number of terminal users (ntusr) plus the number of remote users (namlc) specified by the CONFIG command. (See Figure 16-1.)

```
-----------------------------------------------------------------
|   |           |           |           |           |           |
-----------------------------------------------------------------
1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
```

Bits	Assignment
1-4	Line number (bit 4 is LSB)
5	Not used
6	Data Set Control Bit (1=on)
7	1=Loop line, 0=do not loop

8-10

Octal Value	Standard Line Speed (Data Rate)
0	110 Baud
1	134.5 Baud
2	300 Baud
3	1200 Baud
4	programmable clock
5	75 Baud*
6	150 Baud*
7	1800 Baud*

Note

Speeds marked with * are assignable by hardware jumpers. The speeds shown are the default values. Other choices are 75, 150, 600, 1800, 2400, 4800, 9600 or 19200 Baud.

If octal 4 is specified, the line speed is that set by the programmable clock (i.e., by the AMLCLK config directive). This directive takes values from 29 to 19200 baud. If no clock speed has been set, the default value of 9600 baud is used.

11	Not used
12	0= 1 stop bit, 1=2 stop bits
13	0=enable parity, 1=disable parity
14	0=odd parity, 1=even parity
15-16	Character length:

0	0	5 bits
1	0	6 bits
0	1	7 bits
1	1	8 bits

Figure 16-1. Setup of Line Configuration

The baud rate of the last AMLC line in a group (i.e., the last line on the last board) is the input interrupt rate for all other AMLC lines in that group. (The standard rate is 110.) Changing the baud rate of this line can have adverse effects on the rest of the system. For this reason, it is recommended that users not be allowed to assign this line (and perhaps change its baud rate). Instead, this last line should be assigned to SYSTEM or made non-assignable.

Line: The AMLC line number is an octal number from 0 to the maximum allowed value.

Config: The config argument, which sets the line configuration, is an octal number that corresponds to the bit pattern illustrated in Figure 16-1. Three commonly used values are shown below. They are for data sets with parity disabled and 8-bit character length.

config	baud rate
2213	300
2313	1200 (default)
2413	9600

lword, an optional parameter, is a 16-bit octal integer constructed as follows:

Bit		
1	Set	Line is half-duplex
	Reset	Line is full-duplex
2	Set	Don't echo LINE FEED for RETURN
	Reset	Do echo LINE FEED for RETURN
3	Set	Recognize X-OFF (CTRL S or '223)
	Reset	Don't recognize X-OFF (CTRL S or '233)
4	Set	X-OFF seen
	Reset	Terminal in output mode
9-16	--	Number of user to which AMLC is connected

The user number is the number printed at the terminal upon LOGIN or LOGOUT, or printed by the STATUS command indicating user number. STATUS prints this number as a decimal value but its octal equivalent must be used in the AMLC command. If the rightmost eight bits (9-16) of lword are zero, the AMLC line is not associated with any user space and is available to be assigned.

Terminal characteristics such as FULL DUPLEX/HALF DUPLEX, XOFF, etc. may be set at the user terminal with the PRIMOS-level TERM command.

The AMLC command may be used to configure ASSIGNed AMLC lines as well as terminal AMLC lines. After the system is running, users may assign/unassign the AMLC lines through the following commands:

ASSIGN AMLC [protocol] line [configuration]

UNASSIGN AMLC line

Where the parameters are the same as for the AMLC command.

The system I/O clock rate is that of the top line number of the actual configuration (each AMLC board has 16 lines). This is independent of the number of users for whom the system is configured.

There may be from one to four AMLC boards. Each board has up to four ports (C,D,E,F). Each port has four cable connectors (J1,J2,J3,J4). Line numbers may be calculated with the formulae below:

AMLC Board(x)	Address	AMLC Port	y	Cable connector	z
1	54	C	1	J1	1
2	53	D	2	J2	2
3	52	E	3	J3	3
4	35	F	4	J4	4

Line number	Formula
Physical line number off AMLC	16(x-1)+4(y-1)+z-1
User number	16(x-1)+4(y-1)+z+1
Actual line	16(x-1)+4(y-1)+z
AMLC command line (line) - octal	20(x-1)+4(y-1)+z-1
Configuration number - octal (for NAMLC directive; see the CONFIG command in Section 17)	20(x-1)+4(y-1)+z+1

Note

Certain connectors are not used, depending upon the number of AMLC boards.

▶ ASSIGN

The operator may use the ASSIGN command to assign magnetic tape drives or other peripheral devices to users. The ASSIGN command is discussed in Section 3.

▶ BATCH SYSTEM { - START | - STOP | - DISPLAY }

The BATCH command starts and stops the Batch monitor. It also allows operators and users to monitor usage of the Batch subsystem. It is fully described in Section 9.

▶ 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 the System Administrator read-enables the BATDEF file by setting its protection to 7 1, users can use the BATGEN command to monitor queue availability and parameters.

BATGEN is described in Section 12.

▶ CHAP { -userno [priority] [timeslice] | ALL }

CHAP is an internal operator command that changes a user's timeslice and priority level in the ready list.

userno — The number of the user to be modified in the form -nn. The priority and timeslice of user 1 cannot be modified.

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 timeslice value in tenths of a second.

If specified as 0, the timeslice is reset to the standard value. If omitted, the timeslice 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 timeslice is reset to the standard value. If ALL is specified instead of -userno, the specified changes are made to all users. The default value is '24 (20 decimal)- a two-second timeslice.

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 Section 17.

COPY

COPY is an external command that copies one disk to another and verifies the copy. It is fully described in Section 7. COPY may be used from any terminal.

DISKS [NOT] pdev-0 [pdev-1] ... [pdev-7]

The DISKS command may be given only from the supervisor terminal. The DISKS command adds or removes the specified physical disk(s) to/from the Assignable Disks Table. 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 subsequently 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:

```
DISKS 20250 50250 60250 70250 10020 110250 20252
```

adds the specified disk and partitions (20250, etc.) to the Assignable Disks Table. These partitions may now be ASSIGNed by the users or operators. The command sequence:

```
DISKS NOT 20250
UNASSIGN 20250
```

removes the physical disk partition 020250 from the Assignable Disks Table and unassigns that partition.

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.

▶ ELIGTS tenths

ELIGTS is an internal command which modifies the eligibility time slice for system users.

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 another scheduler queue, the eligibility queue, and the user time slice is decremented by the ready time slice. This 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 user time slice is exhausted, at which time the user is entered in the low priority queues.

▶ FIXBAT

FIXBAT is a utility for checking the batch queue database integrity. It is normally supplied in UFD BATCHQ as *FIXBAT. A complete description is in Section 12.

▶ FIXRAT

FIXRAT is an external command that checks the PRIMOS file integrity on any physical device or partition. It is described fully in Section 5. FIXRAT may be invoked from any terminal.

▶ JOB

The JOB command allows the operator to monitor, hold, release, change, cancel, or abort a user's Batch jobs. It is discussed in Section 9.

▶ LOGOUT { -usernumber | ALL }

Logs out the process specified by usernumber or logs out ALL users connected to the system (except the FAM, if it is a running process). Files are closed, devices deassigned, the UFD is detached, and assigned segments returned to the supervisor. LOGOUT ALL also sets MAXUSR to 0; no one can login until MAXUSR is explicitly reset. See the MAXUSR command.

▶ LOGPRT

LOGPRT is an external command that prints the contents of the event logging file. It is described in detail in Section 4. It is normally supplied in UFD SYSTEM as *LOGPRT.

▶ LOOK [-userno] [segno] [access] [mapseg]

LOOK is an internal operator command that provides access to any segment in the system.

userno Number of the user owning the segment (default is user 1).

segno Number of the segment to be examined. The 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. The 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. 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.

▶ MAKE

MAKE creates a structure for any PRIMOS-supported disk pack or partition. It is described fully in Section 6.

▶ MAXSCH n

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 fast I/O wait queues.

The parameter n is the value of the variable MAXSCH in SUPCOM; the default value is 3.

▶ MAXUSR n

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. When it is necessary to limit the number of users on the system, the MAXUSR command can be used. Thereafter, no more than n users will be allowed to login. If the number of users is already above n, no user will be forcibly logged out, but no new users will be allowed to login and no phantoms will be started. The value of n is ignored for phantoms started from the supervisor terminal (user 1); other users attempting to login or start a phantom 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 zero will prevent new logins just before the shutdown. LOGOUT ALL logs out all users and sets MAXUSR to zero.

▶ MESSAGE

MESSAGE is an internal command with the following facilities:

From any user terminal -

- Send a message to the supervisor terminal.

From the supervisor terminal -

- Send a message to all users on the local node.
- Send a message to a specified user on the local node.
- Send a message to a supervisor terminal at a different network node.

The format for user-to-operator message is:

MESSAGE
text-of-message

text-of-message is a one-line message. Two lines are printed at the supervisor terminal:

*** uu hh'mm
text-of-message

uu is the user number; hh'mm is the time of day in hours and minutes.

The format of an operator-to-user message is:

MESSAGE ALL [NOW]
text-of-message

MESSAGE -uu [NOW]
text-of-message

MESSAGE node-name
text-of-message

The operator can send messages to all users or to a single user. When the parameter ALL is specified, the message is sent to all users. The parameter -uu is a dash followed by the user number. When the parameter -uu is specified, a message is sent to the user specified.

If the optional argument NOW is not specified, the message is stored in a Broadcast Buffer (ALL) or a Single User Buffer (-uu). 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.

If the argument NOW is specified, the message is printed immediately. This is undesirable if the user is in the middle of a sensitive operation. When NOW is specified, stored messages are not affected. Also, when NOW is specified, the format of the message at the user terminal is:

*** BULLETIN ***

If the operator attempts to send a message to a single user before the previous message to a single user has been received, the error line:

MESSAGE NOT SENT

is printed at the supervisor terminal.

Users who login on a remote node receive messages from both the local and the remote node.

If node-name is included after MESSAGE, the message will be printed at the supervisor terminal of that node of the computer network. If the node-name given is not connected to the local node then the error message "Illegal name" will be printed after the operator enters the message to be sent.

Note

The node name of the originating system is not printed at the receiving system. The sending operator should include the originating node-name as part of the message text.

To cancel a stored message, a null line must be entered as the text of message.

▶ NETCFG

NETCFG builds the network configuration file. It is described in detail in The PRIMENET Guide.

▶ NETPRT

NETPRT is an external command which prints the contents of the network event logging file. It is described in detail in section 4. It is normally supplied as NETPRT in UFD PRINET>CMDNC0 or in UFD X.25>CMDNC0.

▶ OPRPRI $\left\{\begin{matrix} 1 \\ 0 \end{matrix}\right\}$

OPRPRI is an internal command that allows the SHARE command to be issued from the supervisor terminal. This command is implemented as a check and balance to prevent inadvertant or unauthorized use of the SHARE command that might adversely affect the system.

1 Removes safeguard and allows the SHARE command to be given.

0 Resets safeguard against issuing SHARE command inadvertently. This is the default.

▶ PROP

PROP is an external command invoking the spool queue management utility for the system printer. It is described fully in Section 8.

▶ REMOTE { PERMIT | DENY } [option]

The REMOTE command enables the supervisor terminal user to permit or deny access to local file system disk partitions from specific or from all remote nodes.

PERMIT Permits access to specific or all local disks by specific or all remote nodes.

DENY Denies access to specific or all local disks by specific or all remote nodes.

options are described below; their values can be:

```
nodename pdev-1  [pdev-2 ... pdev-9]
nodename -ALL
-NET     pdev-1  [pdev-2 ... pdev-9]
-NET     -ALL
```

Examples

The following examples illustrate how this command is used to permit access from remote nodes (to deny access, the DENY keyword is used).

REMOTE PERMIT nodename pdev-1 [pdev-2 ... pdev-9]

Permits the operator at system nodename to STARTUP or ADDISK any of listed devices (at least one device must be specified in this list). All local devices specified in this list must already be started-up with a previous ADDISK or STARTUP command.

REMOTE PERMIT nodename -ALL

This command permits the operator of system nodename to use the STARTUP or ADDISK commands 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-9]

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.

▶ REPLY -usernumber -TAPE option

The operator uses the REPLY command to reply to users' mag tape requests. Options are:

Option	Meaning
GO	Operator has mounted a tape and/or assigned a drive. User sees message, "Device MTpdn assigned."
pdn	(Physical device number) User has sent an "ASSIGN MTX" request, and operator has assigned him drive number pdn. User sees message, "Device MTpdn assigned."
ABORT	Operator cannot fulfill request for some reason (device not available, tape not found, etc.) User sees message, "Magtape Assignment Request Aborted (ASSIGN) ER!"
RESEND	Operator requests system to display user's message again at supervisor terminal.

The operator may request the display of all outstanding requests by giving either the command, "REPLY -TAPE RESEND", or the command " REPLY -ALL."

Outstanding tape requests are displayed at three-minute intervals at the supervisor terminal. The operator may change the time interval between displays by giving the command,

REPLY -REPEAT seconds

Seconds is a decimal number specifying the repeat interval in seconds.

The degree of operator control over magnetic tapes is set using the SETMOD command.

▶ SETIME -mmddyy -hhmm

The SETIME command sets date and time. It can be entered at any time during system operation. The SETIME command must be given following a startup of PRIMOS before any users are allowed to login.

mmddyy are digits that represent the month, day, and year (last two digits), and hhmm are digits that represent the time in hours and minutes. The two arguments to SETIME must be separated by spaces and start with a dash as the first character. For example:

```
SETIME -050178 -1630
```

sets the date and time: May 1, 1978, 4:30 PM.

▶ SETMOD { -OPERATOR | -USER | -NOASSIGN }

The SETMOD command must be given by the operator from the supervisor terminal. It sets the mode for mag tape assignments, as follows:

Option	Meaning
-OPERATOR	All mag tape assignments must be made by operator. All user ASSIGN commands for mag tape drives result in messages at the supervisor console; the operator must then make the assignments and reply to users with the REPLY command.
-USER	Users can assign tape drives at their own terminals. Users can also request operator assistance by using the "MTX," "-TPID," "-track," "-protect," and "-density" options of the ASSIGN command. The system is in USER mode (the default mode) when cold started.
-NOASSIGN	Tape drives cannot be assigned. User issuing an ASSIGN command sees message, No magtape assignment permitted. (AS) ER! This mode of operation may be used to indicate the temporary absence of the operator in an environment where the operator makes all tape assignments.

▶ SHARE [pathname] segment-number [access-rights]

SHARE is an internal command for the installation of 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. Segments '2000 to '2077 are available to hold shared programs. Those shared programs can be executed by any user. For shared segment assignments, see Table 3-1.

It is also possible to change the access rights of supervisor segments '1 to '12. 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 not recommended.

filename	An optional parameter naming a runfile in the current UFD. If filename 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-'12 and '2000-'2077 should be specified. Specification of a segment number outside these ranges will cause unpredictable results.
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.

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.

▶ SHUTDN [pdev-0 [pdev-1 ... pdev-n]]
ALL

The SHUTDN command performs tasks necessary to shutting down the PRIMOS system in an orderly manner.

SHUTDN ALL performs a complete PRIMOS system shutdown. All user files are closed, physical disks are disconnected from the system, and the PRIMOS systems shut down by inhibiting interrupts, exiting page mode, stopping the system clock, and halting. SHUTDN with no arguments specified performs the same functions for PRIMOS II. Under PRIMOS, the SHUTDN ALL command from the supervisor terminal causes the prompt:

REALLY?

The operator must answer YES for the shutdown to occur.

If the SHUTDN command is issued with a list of physical devices (pdev-0 ... pdev-n), the listed devices are closed 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 STARTUP command. The STATUS command can be used to list the devices currently started up. Unlike the STARTUP command, the pdev's do not have to be given in logical drive order.

Caution

Do not shut down the physical device associated with logical 0, or PRIMOS will lose the command directory (from memory, not disk). To recover, STARTUP the disk and ATTACH CMDNC0.

Note

If the shutdown is a normal one, we recommend using the BATCH SYSTEM -STOP and LOGOUT ALL commands before using SHUTDN ALL.

Example of Selective Shutdown

Assume the initial STARTUP command was:

STARTUP 51 50 56 54

To replace the pack on physical device 56 (logical device 2): give the command SHUTDN 56, stop the drive, replace the pack, and restart the drive.

Networks

Remote devices connected to the system can be disconnected by the command:

SHUTDN nodename pdev-1 [pdev-2] ...

nodename	Network name of the system on which the devices are physically mounted.

pdev-n — Remote devices, made available to the local system, which are to be disconnected.

Note

If a device is shut down at its local system it is no longer available for use on any other system in the network.

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.

▶ STARTUP [PROTECT] comdev [pdev-1...pdev-8]

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-1... 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 non-existent 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.

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 only be specified for disks that are added locally via the ADDISK or STARTUP command and for disks that have a settable write-protection feature.

The status of the write protection assignments may be changed for a given pdev by respecifying the ADDISK or STARTUP command with or without the PROTECT option.

If a subsequent STARTUP command is issued for the same disk, and PROTECT is not specified, the write-protect feature is disabled. (A STARTUP PROTECT to an already protected disk does not change the protection.) If a STARTUP 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 inadvertantly written upon.

Networks: Devices on other systems on the network may be made available to users on the local system by the command:

STARTUP node-name pdev-1 [pdev-2] ...

node-name the network name of the system on which the physical devices are located.

pdev-n the physical device numbers of the devices on the remote system which are to be made available to local users. These device must be connected to their local system by STARTUP or ADDISK before being made available to any other system.

Remotely added devices have the write-protection status assigned them at their local system. The write-protection status of a device cannot be changed remotely.

▶ STATUS

STATUS is an internal command for monitoring system usage. In its full form at the supervisor terminal, it prints network node name, PRIMOS file units open, physical devices connected to the system (volume-name, logical device number, physical device number, network node), status of network nodes, the paging and command devices, and logged in users (login name, user number, line number, devices and peripherals). Subsets of this information can be obtained by appending keywords

(DISKS, USERS, etc) to the STATUS command (see below). If the STATUS command is given at a user terminal slightly different information will be obtained (see Reference Guide, PRIMOS Commands FDR3108).

Some typical instances where the STATUS command must 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 and a message is printed at the terminal to that effect.)

The STATUS ALL command prints all the system information shown in Figure 16-2. (At the supervisor terminal the command STATUS will print out the same information as STATUS ALL.)

```
OK, STATUS ALL

USR=SYSTEM  ENC                                      I  }
                                                         } UNITS
NO FILE UNITS OPEN                                   II }

DEVICE  USRNAM  USRNUM  LDEVICE                      III    DEVICES

DISK    LDEV  PDEV    SYSN                           IV
SYSENC   0      460
DRAFT    1    10460
CADDB1   2   100461
CADDB2   3    23060
FREEC1   4      462
FREEC2   5   100463
TESTC1   6    21462
SOFTWR   7     3462   ENB
SPOOLB  10      460   ENB
MISCEL  11    71063   ENB                                   DISKS
SYSEND  12      460   END
TRANS   13    12060   END
TRANS2  14    52061   END
SYSENE  15      460   ENE
CPUGR1  16    12460   ENE
CPUGR2  17    61461   ENE
MFGTFR  20      462   ENE
SYSENF  21      460   ENF
MISCE2  22    12060   ENF
PERIPH  23    52061   ENF
SYSENG  24      460   ENG
SUBSYS  25    11060   ENG
DBGRP   26    61461   ENG
```

Figure 16-2. Information Printed by STATUS Command at Supervisor Terminal

```
SMLC NETWORK                                          V

      NODE        STATE
    ENC           ****
    EN.F2         DOWN

RING NETWORK

      NODE        STATE
    ENC           ****
    ENA           DOWN
    ENB            UP
    END            UP                                     NETWORK
    ENE            UP
    ENF            UP
    ENG            UP
    RES.C1         UP
    TE.CAD         UP
    TE.PCB         UP
    TE.PC2         UP

PAGDEV = 10462 COMDEV = 460                           VI

USER    NO LIN PDEVS                                  VII
MASSE    4  2    462
VMS     27 31  52061 (SYSD   )
FLOWER  44 75   3462 (FROM SYSB TTY #3)                   USERS
SYSTEM   1 76    460 SMLC00
FAM     24 77    460 (2)
SYSTEM  25 77    460 (2)

OK,
```

Figure 16-2. Information Printed by STATUS Command at Supervisor Terminal (continued)

Description of Status Information

Section	Information
I	The user will always be SYSTEM followed by the local node-name, if any. (e.g., SYSC)
II	Prints a list of all PRIMOS file units currently open. In this example there are no open file units.
III	Prints a list of mag tape devices currently assigned, giving physical device number, username of user to whom the device is assigned, usernumber of this user, and logical device number which user has assigned to the physical device. (If user has assigned no logical number with the -ALIAS option of the ASSIGN command, then LDEVICE is the same as DEVICE).
IV	Column 1 (DISK) is the packname of the disk or partition, i.e., the name of DSKRAT file. Column 2 (LDEV) is the logical device number associated with the physical device by the STARTUP command. Logical device 0 must be the command device; the paging device or partition is not included in this list (see V) as it is not directly accessible by user or operator. Column 3 (PDEV) is the physical device number indicating type of device, drive unit, partition size, and offset (see Appendix D). The fourth column (SYSN) tells the node of the network on which the disk is physically mounted. A blank in this column means the disk is a local one; a node name in this column shows that this is a remote disk mounted at that node of the network.
V	There are four types of networks: RING, IPC, PDN, and SMLC; this section indicates which type is in use. (Under PRIMOS more than one may be in operation simultaneously). The node name is given and the state of that node, either UP (in operation) or DOWN (not in operation). The local node is indicated by ****.
VI	Specifies the paging device and command device physical device numbers. The command device is that started up as logical device 0 (see CONFIG and STARTUP).
VII	Lists users currently logged into the system. Column 1 (USER) is the login directory name of the user. Column 2 (NO) is the user number; this is a decimal number and is usually the line number plus 2. Column 3 (LIN) is the AMLC line number of the user terminal (octal). Specially assigned line numbers are:

LIN	Meaning
75	user logged in remotely from another node in the network
76	user is the supervisor terminal using USRASR command
77	all phantom users

Column 4 (PDEVS) lists all disks 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 home UFD, current UFD, or if the user has opened any files on that disk. Currently assigned devices are indicated after disks used with the same device abbreviation 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 shown in this column is:

- remote login to another system on the network (No example here).
- remote login from another system on the network (FLOWER is logged into SYSB from terminal 3 on SYSB).
- user priority (FAM is running at priority 2). Normal user priority is 1; a priority of 1 is not printed. (See CHAP.)
- use of a remote disk (VMS is using a disk on SYSD). (See CHAP.)

Partial Information

Subsets of the total status information may be obtained by appending key words to the STATUS command. (At the supervisor terminal, STATUS ME is identical to STATUS USERS.)

Command	Sections printed
STATUS DEVICES	III
STATUS DISKS	IV
STATUS NETWORK	V
STATUS UNITS	I, II
STATUS USERS, ME	VII

Note

Section VI is printed only by the STATUS ALL (or STATUS) command.

▶ UNASSIGN

The operator can use the UNASSIGN command to deassign a peripheral device assigned to any user. UNASSIGN is discussed in Section 3.

▶ USRASR user-number

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 supervisor commands after associating with a user.

user-number is a user number.

Example:

```
USRASR 4
```

The USRASR command works only if the associated communications line is not enabled on the AMLC. The communications line can be disabled with the AMLC command:

```
AMLC TTYNOP 2
USRASR 4
```

The AMLC command disconnected the user normally attached to AMLC line 2.

To return to operations as a normal supervisor terminal, type:

```
USRASR 1
```

SECTION 17

CONFIGURATION DIRECTIVES

COMMAND FILE WITH CONFIG

PRIMOS can be started up with a command file containing the CONFIG command. The user first creates a command file named C_PRMO and places this file in the UFD CMDNC0. The first executable line in the command file C_PRMO must be a CONFIG command. The command file C_PRMO may contain other PRIMOS internal commands such as STARTUP and MESSAGE, and must terminate with the command line: CO -END. Comment lines may be used anywhere in the command file.

If CMDNC0 has a password, the nonowner password must be blanks and C_PRMO must allow read access rights for the nonowner.

Starting PRIMOS from a Command File

To bring up PRIMOS from the command file C_PRMO, the operator first starts up, under PRIMOS II, the physical disk that is to be logical disk 0 at the supervisor terminal. This logical disk 0 will be the command disk under PRIMOS. The operator then attaches to PRIRUN and RESUMEs PRIMOS.

The PRIMOS operating system then starts running and checks if a command file C_PRMO exists in the UFD named CMDNC0 on logical disk 0. If C_PRMO does not exist, PRIMOS will take commands from the supervisor terminal. If C_PRMO does exist, PRIMOS will read commands from the command file until it encounters the command line, CO -END (or CO -TTY). Subsequently it will return to PRIMOS and take commands from the terminal.

Errors

If the user types a detectable error while issuing the CONFIG command, the system requests that the CONFIG command be retried.

CONFIG EXTENSIONS

Configuration parameters are accepted in the form of CONFIG directives. These directives are kept in a data file in CMDNC0 and are processed by the preloader to set up all the system parameters specified by the CONFIG command. In addition, the configuration directives can be used to set variables in FIGCOM and to override the default network configuration.

Overview of Preloader Actions

The preloader attaches to CMDNC0 and looks for the file C_PRMO. If the file exists, it is opened for command input; if it doesn't, PLEASE ENTER CONFIG is printed at the supervisor terminal. (For an example of the interactive CONFIG directive, see SINGLE-LINE CONFIG COMMAND at the end of this section.) The first ((executable directive is read (from the terminal or from C_PRMO), and a 'CO -END' is issued. The directive is examined to ensure that it is a CONFIG directive.

The preloader then checks if the keyword -DATA is the first name following the CONFIG command. If this keyword is present, the second name following the command is taken as the configuration data filename. The file is opened for input, and configuration directives are processed as described below. The format is:

    CONFIG -DATA configuration-data-filename

No restrictions are placed on configuration-data-filename, the name of the configuration data file; however, it is suggested that the name CONFIG be adopted as a de facto standard.

A one-line CONFIG command can be included anywhere in a configuration data file. However, use of the CONFIG directives described in this section are recommended, rather than the one-line CONFIG commands. The CONFIG command will not be printed unless TYPOUT YES is in effect (see TYPOUT command). One-line CONFIG options are:

0/ntusr	Number of terminal users
1/pagdev	Paging device
2/comdev	Command device
3/maxpag	Number pages physical memory to use
4/altdev	Alternate paging device
5/namlc	Number assignable AMLC lines
6/npusr	Number phantom users
7/nrusr	Number remote users
10/smlcon	Non-zero value enables SMLC

Note

All numerical values in command lines are octal unless otherwise specified.

Networks

All detailed network configuration information is contained in the NETCON file created by NETCFG (see Section 18). To process this information the directive NET ON should be included in the configuration file.

CONFIG Directives

Directives for configuration data files are given below. Comment lines may be inserted in this file; they should begin with the characters /*. All numerical parameters are octal unless otherwise specified.

▶ ABBREV {YES | NO}

The YES option (the default) allows users to use ABBREV files. The NO option prohibits the use of ABBREV files. (This command sets the FIGCOM variable ABBRSW.)

▶ ALTDEV physical-device [records]

Specifies the alternate paging device and, optionally, its size.

<u>physical-device</u> The physical device number of the paging disk or partition.

<u>records</u> The size of the alternate paging device. If specified (and the equivalent parameter is specified in PAGDEV), then the total number of segments needed by PRIMOS (NSEG) will be automatically calculated. (<u>records</u> is a 16-bit non-zero positive integer.)

This directive is equivalent to the one-line CONFIG parameter 4/physical-device.

The alternate paging device is used <u>only</u> if the primary paging device (PAGDEV) has <u>records</u> specified.

▶ AMLBUF line [in-buff-size] [out-buff-size] [dmq-size]]]

Sets terminal I/O buffer sizes.

<u>line</u> The AMLC line number for which buffer sizes are to be set. For terminal and remote users, this value is the physical line number. For assignable AMLC lines, this value is ntusr+nrusr-1 for the first assignable AMLC line, ntusr+nrusr for the second, up to ntusr+nrusr+nalmc-2 for the last assignable AMLC line.

<u>in-buff-size</u> The terminal input buffer size in words. If 0 is specified, buffer size is unchanged. The default value is '140 (96 decimal).

out-buff-size — The terminal output buffer size in words. If 0 is specified, buffer size is unchanged. The default value is '300 (192 decimal).

dmq-size — The size in words for the DMQ buffer (only meaningful if the system has DMQ AMLC controllers). If omitted or specified as 0, its value is not changed. The default value is '40 (32 decimal).

Total size of input buffers plus output buffers may not exceed 64K words. Exceeding this limit produces the message "TERMINAL I/O BUFFERS TO LARGE".

No individual buffer may exceed '3604 (2024 decimal) words. Exceedinq this limit -- or specifying a dmg-size that is not a power of 2 -- produces the message "BAD DMQ AMLC CONFIGURATION".

▶ AMLCLK baudrate

baudrate specifies the desired baudrate for the software programmable clock in the AMLC hardware. The value specified must not be less than '144 octal (100 decimal) nor greater than '22600 octal (9600 decimal). (This operation was performed previously by changing the variable "AMLCLK" in the AMINIT module.)

▶ AMLTIM [ticks] [disctime] [gracetime]

The AMLTIM directive controls three variable event timers.

ticks — The time interval (in decimal tenths of a second) between carrier check operations. At the end of each period, the AMLC process checks each line for carrier loss. If a loss has occurred, the process forces the line's Data Terminal Ready (DTR) signal inactive until the next sample period, thus disconnecting its incoming telephone line. The value for this argument must be greater than zero. Default is 2, i.e., 0.2 seconds.

disctime — The time period (in decimal tenths of a second) for forcing the DTR signal to the modern inactive on lines that do not have active carriers. Specifying a value of zero disables this feaure. Otherwise, the value specified must not be less than the value of ticks and is truncated to the nearest multiple of that value. Default is 1800, i.e., three minutes.

gracetime	The minimum grace period (in decimal tenths of a second) for terminal lines that have active carriers but are not connected to logged-in processes. (The actual grace period varies randomly from gracetime to twice gracetime.) Default value is zero, i.e., the argument is disabled. Specified values (if not zero) must be greater than ticks, and are truncated to the nearest multiple of ticks.)

▶ ASRATE control-word

Sets the supervisor terminal Baud rate.

control-word is an octal integer specifying supervisor terminal Baud rate (default value is '1010).

control-word	BAUD rate (decimal)
110	110
1010	300 (default)
2010	1200
3410	9600

If used, the ASRATE directive must be the first one in the configuration data file. This ensures that any subsequent configuration error messages are printed at the appropriate speed.

The command is equivalent to, and overrides, the B-register setting of *COLDS, the cold start program. If the ASRATE directive is omitted, and if the system includes an SOC controller (and not an OPTION-A controller), the speed of the supervisor terminal (user 1) will be the same as it was under PRIMOS II.

▶ ASRBUF line [in-buff-size [out-buff-size]]

Sets ASR terminal I/O buffer sizes.

line	The ASR line number. At present, only 0 is a valid line number.
in-buff-size	The ASR terminal input buffer size in words. If 0 is specified, buffer size is unchanged. The default value is '200 (128).
out-buff-size	The ASR terminal output buffer size in words. If omitted or specified as 0, buffer size is unchanged. The default value is '300 (192).

▶ COMDEV physical-device

Specifies the system command device, i.e., the device on which the system UFD=CMDNC0 resides.

physical-device is the physical device number of the command disk or partition.

This directive corresponds to the one-line CONFIG parameter 2/physical-device.

The command device must be specified with either the COMDEV or CONFIG command in the configuration data file.

▶ CONFIG options

Specifies basic system configuration.

This directive and its options are discussed at the beginning of this section.

▶ DISLOG {YES | NO}

Sets disconnect logout option.

YES The user will be logged out if AMLC line is disconnected.

NO The user is not logged out if AMLC line is disconnected (Default).

This directive sets the variable DLOGOT in FIGCOM.

▶ ERASE {character | octal-value}

Specifies system default erase character; supplied default is ".

character Printing ASCII character, such as " (which is the default value).

octal-value The octal value of any ASCII character. For example:

```
ERASE @
is equivalent to
ERASE 300
```

For example, ERASE 377 sets the system default erase character to DEL.

This directive sets the variable DEFERA in FIGCOM.

▶ FILUNT reserved-unit max-unit total-unit

Defines the number of file units available to the user, and to PRIMOS.

reserved-unit	Maximum number of file units guaranteed to be available to each user. Default is 16 ('20).
max-unit	Maximum number of units any one user may have open at one time. Default is 128 ('200).
total-unit	Total number of units that may be simultaneously open in the system. Default is 2048 ('4000).

If the FILUNT directive is not specified in the configuration file, the default values are used.

The maximum total number of units that may be open simultaneously by all users is 2048; total-unit may be used to reduce this number. The effect of reducing the total number of file unit table entries in the system is to reduce the amount of virtual memory used by the file management system. PRIMOS attempts to minimize the file unit table entries in use to control the size of the working set. At cold-start, three file units are allocated for each configured user.

The maximum number of units that one user may have open simultaneously is 128. Of the 128 units, 2 are reserved for exclusive use by the system. max-unit may be used to reduce this number, but not below 2. Since the file unit numbers available are 1 to max-unit, much Prime software will not run if this value is below '22 (decimal 18) (for example, FUTIL). The highest numbered file unit available is max-unit-1. It may be desirable, in special circumstances, to restrict max-unit to 16 for compatability with PRIMOS II and PRIMOS III.

The number of file units guaranteed to be available to each user is 16. reserved-unit may be used to increase or decrease this quantity. Since there are not enough file unit table entries to permit all users to have 128 file units open simultaneously (128*128=16384), the PRIMOS file management system subroutine SRCH$$ may return the error code E$FUIU (all units in use). If multiple cooperating processes (users) depend on having a certain number of file units available, the possibility of a deadlock exists. reserved-unit must be specified so that there are sufficient units available to prevent deadlock. That is, total-unit must be greater than or equal to reserved-unit*n, where n is the number of configured users, and total-unit ≤2048.

▶ GO

Marks end of configuration data file; any subsequent lines are ignored. The configuration data file must include a GO directive.

▶ KILL {character | octal-value}

Specifies system default KILL character; supplied default is ?

character A printing ASCII character, such as +, ?, *, etc.

octal-value The octal value of any ASCII character.

For example, KILL + or KILL 253 sets the system default kill character to +. This directive sets the variable DEFKIL in FIGCOM.

▶ LOGLOG {YES | NO}

Allows LOGINs while logged in.

YES Users are permitted to use the LOGIN command while logged in (Default).

NO The LOGIN command is inhibited for a logged-in user.

This directive sets the variable LOGOVR in FIGCOM.

Note

The external login program, if present, is run only once if a user logs-in while already logged-in, provided that LOGLOG YES has been specified in the configuration.

▶ LOGMSG {YES | NO}

Prints LOGIN/LOGOUT messages:

YES LOGIN and LOGOUT messages are printed at the supervisor terminal (default).

NO LOGIN and LOGOUT messages are suppressed.

This command sets the variable NLGPRT in FIGCOM.

▶ LOGREC file-size

Specifies maximum size of LOGREC file.

file-size is the size of the LOGREC file in words. The default value is '10000 (4096 decimal). Specifying a size of 0 disables the quota check. Specifying a negative value suppresses writing to the LOGREC file; this should be used when running a write-protected disk.

If size of LOGREC file is exceeded and quota check is not disabled, the message EXCEEDING QUOTA ON LOGREC will be printed for each additional entry added to the file.

This command sets the variable LRQUOT in FIGCOM.

▶ LOUTQM minutes

Specifies inactivity time for automatic LOGOUT.

minutes is the number of minutes of inactivity allowed, at the terminal, before the user is automatically logged out. The default value is '1750 (1000 decimal). The value must be greater than 0.

This directive sets the variable LOUTQM in FIGCOM.

▶ MAXPAG number-of-pages

Specifies number of pages of memory to validate. Memory validation occurs at each cold start. This directive must be included in order to use more than 256K words (512KB) of memory.

number-of-pages is the number of '2000 (1024 decimal) word pages of physical memory to validate for use.

This directive corresponds to the one-line CONFIG parameter 3/number-of-pages.

▶ NAMLC number-of-lines

Specifies number of assignable AMLC lines in the system.

number-of-lines is the number of assignable AMLC lines in the system; the default is 0.

This directive corresponds to the one-line CONFIG parameter 5/number-of-lines.

▶ NET ON

This directive specifies that networks are to be configured. If this directive is not specified, then networks will not be configured. (See Section 18.)

▶ NPUSR number

Specifies number of phantom users to be configured.

number is the number of phantom users for which the system is to be configured. (Non-negative octal integer.) Default is 0.

This directive corresponds to the one-line CONFIG parameter 6/npusr.

▶ NRUSR number

Specifies the number of processes to be reserved for remote logins.

number is the number of remote users for which the system is to be configured. (Non-negative octal integer). Default is 0. The maximum is '40 (decimal 32).

This directive corresponds to the one-line CONFIG command parameter 7/nrusr.

▶ NSEG number

Specifies the total virtual address space for a system (i.e., the variable NSEG in SEGMENT 4).

number specifies the number of page maps to be allocated during system initialization. There may be fewer page maps available than the number of possible user segments. A 64 user system can allow 64 possible segments to be addressed by each user, but only number segments can actually be used by all users at any given time. The system allows a maximum of 511 decimal, ('777) page maps. The default value of number is 192 decimal ('300).

If the paging space specified by PAGDEV (and ALTDEV) does not allow NSEG segments to be allocated, NSEG is reduced to conform to the paging space requirements.

▶ NTUSR number

Specifies number of terminal users.

number is the number of terminal (local system) users for which the system is to be configured. (Positive octal integer between 2 and 100.)

This directive corresponds to the one-line CONFIG parameter 0/number.

NTUSR is added to NPUSR and NRUSR to determine the total number of users configured on the system.

▶ NUSEG number

Sets the size of each process' descriptor table 2 (virtual address space size). number specifies the number (octal) of segments available to each user process. PRIMOS system reserves room for a total of 4096 ('8000) user segments. The product of number times the total number of users (including phantoms and remote login users) cannot exceed 4096. The default value of number is 40 octal.

▶ PAGDEV physical-device [records]

Specifies paging device and size.

physical-device	The physical device number of the disk or partition on which paging is to take place.
records	The size of the paging disk (a non-negative 16-bit integer). If specified, this value (with the ALTDEV value of records, if any) limits the total number of segments in the system. If records is specified and an alternate paging device has been specified by ALTDEV, then records defines the point at which page space allocation switches from the primary to the alternate paging device.

This directive corresponds to the one-line CONFIG parameter 1/physical-device. The paging device must be specified by PAGDEV or CONFIG.

If records is small (e.g., 1), almost all paging is forced to occur on the alternate device. However, the primary device is always used to page the PRIMOS segments (0-'22) and user 1's segments '6000 and '6002.

See Section 6 for directions on creating badspot files on paging devices.

▶ PREPAG pages

Specifies number of pages to prepage.

pages is a positive octal integer specifying the number of pages to prepage out when a page fault occurs. The default value is 3.

This directive sets the variable PREPGK in PAGCOM.

▶ RWLOCK value

Specifies file system read/write lock setting.

value is an octal integer specifying the system-wide file read/write lock. The default value is 1. Possible values are:

Value	Meaning
0	1 reader or 1 writer (writer has exclusive control)
1	N readers or 1 writer (writer has exclusive control)
3	N readers and 1 writer
5	N readers and N writer

The directive sets the RWLOCK variable in FIGCOM.

Note

Many subsystems (such as SPOOL) do not permit multiple writers.

```
SMLC ON
SMLC CNTRLR  controller-number  device-address
SMLC SMLCnn  controller-number  line-number
SMLC DSC     line strap proc rcev n
```

Enables and configures SMLC lines. If used, it disables any SMLC data in the NETCON file (See Section 18.) There are four formats:

SMLC ON: Enables the SMLC in the default ON configuration; the normal OFF default is SMLC disabled. This directive corresponds to the one-line CONFIG parameter 10/1.

SMLC CNTRLR controller-number device-address: Specifies physical device number(s) of the SMLC controllers.

controller-number	the controller, either 0 or 1.
device-address	the physical device address of the specified controller. The default value is address '50 for controller 0; default address for controller 1 is not defined.

SMLC SMLCnn controller-number line-number: maps logical line numbers onto physical controllers and line numbers.

nn	the logical line number; values are 00 to 07.
controller-number	the controller, either 0 or 1.
line-number	the physical line number of the specified controller onto which the logical line number is mapped. The default values map SMLC00 to SMLC03 onto controller 0, physical lines 0 to 3.

SMLC DSC line strap proc recv n: Specifies data set control configuration for a synchronous line on the SMLC/HSSMLC/MDLC board. Only the BSCMAN process currently interprets the data structures initialized by the DSC option. Arguments are specified as octal numbers:

line — logical line number (0 through 3)

strap — Bit set indicates signals which are strapped ON by the software.

:1 - Data Terminal Ready (DTR)

:2 - Request to Send (RTS)

In addition, speed select in Germany is specified via the :10 bit:

SET - fast

Reset - slow

proc — Indicates data set control procedure (to be used for transmitting data) as follows:

1 - No data set orders. Usually used with DTR and RTS strapped ON, with modems used for four wire full duplex service.

2 - Use data set orders as follows:
issue RTS, wait for CLEAR to Send (CTS), send, drop RTS. Usually used with most half duplex modems.

3 - Use data set orders as follows:
wait for .NOT. Carrier Detect (CD), issue RTS, wait for CTS, drop RTS. Rarely used, but may be necessary with 201 series modems only if lines are very noisy. Try "2" first.

recv — Indicates whether the receiver is to be turned on before or after transmitting:

0 - Turn on receiver before transmitting. This provides the fastest response and should be used if possible.

1 - Turn on receiver after transmitting. This setting must be used with two wire 201 series modems. This setting may be tried on other two wire systems only if problems appear which cannot be solved by other means.

The default setup, if no DSC is specified, is the equivalent of including the following line in the configuration data file:

SMLC DSC line 1 2 Ø

▶ TYPOUT $\begin{Bmatrix} \text{YES} \\ \text{NO} \end{Bmatrix}$

Controls printing of configuration commands.

YES subsequent commands in configuration data file will be printed on the supervisor terminal as they are processed. Printing continues until a TYPOUT NO or GO command is encountered.

NO commands are not printed as they are processed. Printing is suppressed until a TYPOUT YES or GO command is encountered. TYPOUT NO is the default.

Any number of TYPOUT commands can be used in a configuration data file to print selected commands.

▶ UPS number

Sets system to perform restart after power failure. This directive enables an automatic warm start after power is restored following a power failure check. It is designed to be used when an Uninterruptible Power Supply (UPS) is used to maintain power to the CPU and memory.

<u>number</u> is the system UPS action variable. This variable determines what actions are taken after a power failure. Valid values of <u>number</u> are:

177777 no UPS (default).

Ø UPS, but HALT on a warm-start.

>Ø Number of seconds to delay after warm-start (no operator start is required).

The number of seconds to delay after a warm-start is the amount of time it takes for the disk(s) to come up to the proper number of revolutions per minute. Typically, this is about 1 minute (about '1ØØ seconds) for a storage module.

This directive sets the UPSSW variable in FIGCOM.

▶ WIRMEM

This directive causes the size of wired memory, in pages, to be printed at the supervisor terminal during coldstart. This value changes as the system runs. However, it does give some idea of the relative memory cost of the selected configuration.

PRIMOS PRELOADER AND INITIALIZATION ERROR MESSAGES

Below is a list of all error messages generated by the PRIMOS preloader ('PRIMOS') and the PRIMOS and NETWORK initialization sequences. The majority of the CONFIG messages are fatal, and cause configuration to terminate. Any error messages which do not come from the preloader ('PRIMOS'), require that PRIMOS II be booted again from the control panel (i.e., start over from the beginning); the offending directive (or lack thereof) should be corrected before attempting to bring up PRIMOS again.

Error Messages

▶ file-system-error-message config-file (PRIMOS)

A file system error was encountered by the preloader while attempting to open the configuration file config-file.

▶ file-system-error-message CAN'T ATTACH TO CMDNC0 (AINIT)

A file system error was encountered while attempting to attach to CMDNC0 for User 1.

▶ file-system-error-message CMDNC0 (PRIMOS)

A file system error was encountered by the preloader while attempting to attach to CMDNC0.

▶ file-system-error-message C_PRMO (PRIMOS)

A file system error (other than FILE NOT FOUND) was encountered by the preloader while attempting to open the file C_PRMO for command input.

▶ file-system-error-message PRnnnn (PRIMOS)

A file system error was encountered by the preloader while attempting to open or read the indicated PRnnnn file.

▶ BAD directive PARAMETER (AINIT)

One or more of the parameters specified for the configuration directive is invalid.

▶ INVALID CONFIG COMMAND: directive (AINIT)

The directive in the configuration command file is not a recognized configuration directive.

CONFIG Error Messages

▶ BAD DMQ AMLC CONFIGURATION (AINIT)

A DMQ buffer size in an AMLBUF command is not equal to a power of 2, or total size of terminal input, output and DMQ buffers exceeds 64K words.

▶ BAD LINE # IN AMLBUF CMND (AINIT)

An AMLBUF command specifies a line number less than Ø or greater than the number of lines configured for the system.

▶ BAD LINE # IN ASRBUF CMND (AINIT)

An ASRBUF command specifies an invalid line number.

▶ BAD RECORD ADDRESS IS LESS THAN 16. (BADSP$)

A bad record is found to have an address less than or equal to 16. The first 16 records of a partition contains the MFD, BOOT, and other special file.

▶ END OF FILE. MISSING 'GO' CMND (PRIMOS)

The configuration data file does not include the required GO command.

▶ FILUNT INVALID (AINIT)

The FILUNT directive specifies incorrect information for proper configuration.

▶ FIRST COMMAND MUST BE CONFIG

The command typed in response to the 'PLEASE ENTER CONFIG' prompt or the first executable command in C_PRMO is not a CONFIG command. (Pre-loader)

▶ ILLEGAL ALTDEV

The device specified as the alternate paging device is not legal. (Pre-loader)

▶ ILLEGAL COMDEV

The device specified for the command device is not legal. (Pre-loader)

▶ ILLEGAL PAGDEV

The device specified for paging is not a legal paging device. (Pre-loader)

▶ INCORRECT 'BADSPT' FILE FORMAT (BADSP$)

The file BADSPT has an incorrect file format. For example: Starting address (SA) of the saved format file is not equal to '1000.

▶ MISSING NTUSR, PAGDEV, OR COMDEV

The configuration data file did not specify these required parameters. (Pre-loader).

▶ NRUSR INVALID (AINIT)

The number of remote users specified by an NRUSR directive exceeds the maximum number of configurable remote users -32 ('40).

▶ NTUSR, NPUSR, OR NRUSR INVALID (AINIT)

The value of NTUSR, NPUSR, or NRUSR is incorrect.

▶ NTUSR+NPUSR+NRUSR TOO BIG (AINIT)

The number of terminal plus phantom plus remote users exceeds the maximum number of configurable users.

▶ RESTART PLEASE

This message appears following any error message printed by the PRIMOS initialization logic (AINIT). The system will halt at location BOOT0_ in segment 6. PRIMOS II must be reloaded. The offending command in the configuration data file must be corrected.

▶ SEEK FAILURE ON ALTDEV (AINIT)

The initial seek to cylinder 0 on the alternate paging device failed.

▶ SEEK FAILURE ON PAGDEV (AINIT)

The initial seek to cylinder 0 on the primary paging device failed.

▶ SMLC CTRLR # OUT OF RANGE (AINIT)

An SMLC command specifies an invalid controller number.

▶ SMLC LINE # OUT OF RANGE (AINIT)

An SMLC command specifies an invalid line number.

▶ SUM OF BAD SPOTS ON THE PRIMARY AND ALTERNATE PAGING DEVICE EXCEEDS 16

PRIMOS supports a maximum of 16 defective tracks (bad spots) on both primary and alternate paging partitions.

▶ TERMINAL I/O BUFFERS TOO LARGE (AINIT)

The total size of the terminal input and output buffers exceeds 32K words. (AMLBUF and ASRBUF commands)

▶ TOO MANY BAD SPOTS IN 'BADSPT' (BADSP$)

Number of bad spots entries in the file BADSPT exceeds 16. PRIMOS supports a maximum of 16 bad spots in both primary and alternate paging partitions.

▶ TPIOS ERROR

An I/O error occurred while preloading the paging device. (Pre-loader)

▶ USE physical-device FOR PAGING?

The disk physical-device has been specified as the paging device, but is formatted as a standard PRIMOS disk. A reply of YES is required to enable paging on physical-device. (Pre-loader)

NETWORK INITIALIZATION ERROR MESSAGES

▶ file-system-error-message NETCON (AINIT)

A file system error has occurred while opening or reading the network configuration file.

▶ BAD NETCON FILE FORMAT (AINIT)

The network configuration file has an illegal format. Recreate the network configuration file using the most recent version of NETCFG.

▶ NETWORK NOT CONFIGURED (AINIT)

This message is no longer issued.

▶ NO TABLE SYSGEN'D FOR RING # n (NETFIG)
NO TABLE SYSGEN'D FOR IPC # n (NETFIG)
NO TABLE SYSGEN'D FOR SMLC # n (NETFIG)

There are too many nodes of the specified line type. Recreate the network configuration file specifying fewer nodes of that type.

▶ TOO MANY IPC's (AINIT)
TOO MANY RING NODES (AINIT)
TOO MANY SMLC's (AINIT)

The network configuration requires more table space than exists in the operating system. These errors can only be generated if the -NOCHECK option was used with the NETCFG command.

▶ TOO MANY NETWORK NODES
ONLY n NODES ALLOWED (NETFIG)

There are too many total nodes configured. Recreate the network configuration file specifying fewer total nodes.

▶ WARNING — revision text NETWORK CONFIGURATION FILE (NETFIG)

The network configuration file was created for a previous version of the operating system. If there are no subsequent errors then networks have been configured successfully. In any case, the network configuration file should be recreated with the most recent version of NETCFG.

SINGLE-LINE CONFIG COMMAND

The single-line CONFIG command (rather than the configuration data file) can still be used to configure the system. However, the single-line command cannot specify as many features as the configuration data file directives. Users are urged to convert to the data file method of configuring the system.

The command format is:

CONFIG ntusr pagdev comdev [maxpag] [altdev] [namlc]
[npusr] [nrusr] [smlc]

ntusr — An octal integer defining the number of terminal users, including the supervisor (e.g., for a 4-user system, ntusr = 5, for a 7-user system, ntusr = 10).

pagdev — A physical device number specifying the device to be used for paging. (See Appendix D for possible values.)

comdev — A parameter that specifies the physical device number initially assigned as logical device 0. This is the device on which UFD=CMDNC0 is searched when a user invokes an external PRIMOS command. If comdev and pagdev are the same, the disk is considered to be split into a file system and a paging part. The boundary between the partitions is defined by the DSKRAT header, and it may be set by the MAKE program.

maxpag — An optional parameter defining available physical memory storage. It corresponds to the last sector number (octal) to be used. The values for maxpag and associated storage used are as follows:

Blank or 0	All of Memory up to 256K; at least 64K ('100K)
'200	64K ('100K)
...	...
'1000	256K ('400K)

altdev — CONFIG may specify either one or two disk devices on which paging is to take place. The alternate paging device cannot be a split disk.

namlc — An optional parameter defining the number of assignable AMLC lines.

<u>npusr</u> An optional argument defining the number of phantom users.

<u>nrusr</u> An optional parameter specifying the number of remote users who can run on the system.

<u>smlc</u> An optional parameter enabling the SMLC.

For example, the one-line CONFIG command:

```
CONFIG 60 100460 460 1000 6/4
```

specifies a system with 48 ('60) terminals and 4 (6/4) phantom users. The paging device (100460) and the command device (460) are partitions of a storage module on drive unit 0. The value of MAXPAG ('1000) specifies '400K (256K) of available physical memory storage.

The 6/ prefix is a feature of the command line which allows the omission of parameters. In this case, parameters 4 and 5 have been left at the default values (the first parameter, 60, is parameter 0).

SECTION 18

COMMUNICATIONS

PRIMENET

This section covers what you need to know as an administrator for a Prime computer installation which uses PRIMENET. It tells how to configure a network-connected system, how to turn on PRIMENET, and how to enable remote users and the File Access Manager (FAM).

Operations Across the Network

Several operations which involve PRIMENET are not described here because they are performed in almost exactly the same way as they would be under non-network circumstances. These network operations are described with the corresponding local operation.

Configuring the Network (NETCFG)

The first step in configuring a PRIMENET-connected computer involves using the interactive program NETCFG. NETCFG conducts a dialog with the user to obtain the system names and network addresses of the several computers comprising the network, information about any Public Data Networks to which the computer may be connected, and similar information. It produces a file called NETCON which contains (in coded form) the information gathered from the dialog.

PRIMOS uses the NETCON file CMDNC0>NETCON to establish network parameters and connections at system startup time. Since NETCON files in directories other than CMDNC0 have no effect on PRIMOS operation, you may use NETCFG to set up or to review several different network configurations without their interfering with each other or with PRIMOS.

A complete explanation and an example of the use of NETCFG appear in the PRIMENET Guide (IDR3710).

CONFIG Directives

Include the directive NET ON in the configuration data file to cause PRIMENET configuration at system startup. If NET ON is not included, PRIMENET will not be brought up on the system being started.

Use the directive NRUSR in the configuration data file to configure the number of remote (via network) users allowed. This directive is not required if no remote users are to be allowed. Between 0 and '40 (decimal 32) users may be specified.

See Section 17 for complete details about the configuration directives.

Starting the File Access Manager (FAM)

Start the FAM as described under FAM operation in Section 3. Make sure you change the priority to 2 with the CHAP command. When you have determined the phantom user number of the FAM, you can put both the PH FAM>PH_FAM command and the CHAP -nn 2 command into C_PRMO for automatic startup.

DISTRIBUTED PROCESSING TERMINAL EXECUTIVE (DPTX)

The Distributed Processing Terminal Executive (DPTX) is a support and emulation communications subsystem operating on Prime 350 or higher. It allows users at Prime terminals to communicate with IBM host computers (emulation) and users at IBM3277 terminals to communicate with Prime computers (support). In addition, the supported IBM3277 terminals can "pass through" the Prime computer and communicate with an IBM host computer. DPTX is a separately priced software package. It is completely documented in the Distributed Processing Terminal Executive Guide (DPTX Guide).

The System Administrator's responsibilities for DPTX are:

1. Creating the DPTX terminal and controller configuration table(s)

2. Starting and Monitoring the DPTX processes

3. Stopping and Shutting down DPTX

Creating DPTX Terminal Address Tables

The DPTX Terminal Address Tables contain the required information for all support and emulation terminals. This is a block-structured source file containing information on:

- Terminal Addresses

- Controller Addresses (support only)

- Terminal Characteristics

- Terminal function (emulation or support)

This source file is created using the Editor (ED) and is compiled by the DPTCFG program to create a binary file for use in step 2 as the terminal address table in DPTX startup. The content of the source file and the operation of the DPTCFG program are described in the DPTX Guide.

Starting DPTX

DPTX is started by initializing the DPTX processes and the DPTX Terminal Address Table from the supervisor terminal after PRIMOS has been brought up. The procedure is:

1. Initialize the DPTX Terminal Address Tables (required). The DPTX Terminal Address Tables are initialized by the command:

   DPTX -ON

   or

   DPTX -DATA pathname

   The -ON parameter is used if no output filename was specified when the tables were compiled. The -DATA parameter is used if an output file name was specified during table compilation. Pathname is the filename. This must be initiatized first before any of the processes are started.

2. Initialize the BSCMAN process (required). The BSCMAN (Bisync Manager) is the primary software link between PRIMOS and the IBM host. This process is implemented by the phantom PH_BSCMAN.

3. Initialize the emulation and support processes. These processes are implemented as phantoms. The processes and their phantoms are:

   Emulation: PH_EM

   where: EM means emulation process

   Support: PH_TM and PH_DH

   where:

   TM means Traffic Manager process

   DH means Data Handler process

   Support passthrough: PH_EM, PH_TH and PH_DH

   Each phantoms is started by typing:

   PH PH_xx

   where xx is a process abbreviation.

4. Initialize the PRTDSC process. The PRTDSC process may be implemented as an interactive program from a user terminal or as phantom. As an interactive program it may be started from a Prime terminal or an IBM3277. The command for a phantom is:

   PH PH_PRTDSC

   In either case PRTDSC provides a way for host originated data to be recorded on a hard-copy terminal or Prime line printer. The initialization procedures and Process descriptions are in the DPTX Guide.

Stopping DPTX

DPTX can be turned off by issuing a DPTX -OFF command from the supervisor terminal. In this case, PRIMOS continues to operate but DPTX cannot be restarted. To stop DPTX and allow it to be restarted, shutdown PRIMOS and then start it up again with DPTX initialization as described above.

SECTION 19

PRIMOS II OPERATION

This section describes how to load and start PRIMOS II and summarizes the essential operator tasks during a single-user operating session.

OVERVIEW OF PRIMOS II OPERATION

What is PRIMOS II

PRIMOS II is the single-user version of PRIMOS, intended primarily as a bootstrap for PRIMOS and to run diagnostics. Ordinarily, PRIMOS II is used only momentarily during startup of PRIMOS (the bootstrap brings PRIMOS II into memory; See Section 2). However, in some situations it may be desirable to run the computer as a single-user system under PRIMOS II: for backups, file operations with absolute security, disk formatting, magnetic tape operations, etc.

Certain operational aspects of PRIMOS II differ from the time-shared versions. For example:

- It is not possible to interrupt a program with BREAK (CONTROL-P). Instead use the control panel to HALT and then restart by STARTing at a specified starting location. (See Appendices C and B.)

- PRIMOS II occupies the user's address space - the same physical memory as the user. (See Table 19-1.) This places some restrictions on program loading to ensure that PRIMOS II and the loader are not overwritten.

Versions of PRIMOS II

PRIMOS II is supplied on the master disk in a real- and virtual-memory version. The real-memory configuration runs stand-alone on any Prime CPU with high-speed arithmetic. The virtual-memory configuration runs in user space under PRIMOS on Prime 300 (and higher) CPUs. The real-memory version of PRIMOS II is discussed in this section. See the TDOS64 command in Reference Guide, PRIMOS Commands for information on the virtual-memory version of PRIMOS II.

PRIMOS II is '50000 locations in size, including file-system buffers. It is loaded in 64R addressing mode and requires a CPU with the high-speed arithmetic option.

The locations and names of the PRIMOS II configurations as supplied on the master disk are shown in the following list:

UFD	File	Description	Low	Restart	High
DOS	*DOS64	64K real-memory	130000	170000	177777
CMDNC0	TDOS64	64K virtual-memory	130000	170000	177777

File-system buffers are internal to PRIMOS II. The bottom of PRIMOS II is fixed at location '130000, independent of the number of open file units.

The banner printed when PRIMOS II is loaded is:

PRIMOS II REV 16.8 07/02/79 (AT nnnnnn)

where nnnnnn is the restart location.

Bootstrap Loading

PRIMOS II is started up from the control panel by the procedure described in Section 2. When PRIMOS II is boot loaded from the processor control panel, the bootstrap loads file *DOS64 from UFD DOS. PRIMOS II automatically acquires the supervisor terminal typewriter control words directly from the bootstrap. (See Appendix A.)

On systems with a storage module disk drive, PRIMOS II can be bootstrap-loaded from the storage module. The sense switch settings are xx0114.

See Appendix A for additional details on bootstraps.

Starting Up Additional Partitions

When PRIMOS II is booted in, it automatically starts up the partition it boots in -- that is, the first partition in the disk pack specified in response to the "Physical device=" prompt. To start up any other partition under PRIMOS II, use the STARTUP command. The format of this command is:

STARTUP partition-1 [... partition-4]

Two points must be remembered:

- PRIMOS II can start up a maximum of four partitions.
- Every desired partition must be specified in the STARTUP command, even if that partition has already been started up. For example, if PRIMOS II automatically started up disk 460, and the operator wanted to add disk 010460, the command would be:

STARTUP 460 010460

(If the command were given as "STARTUP 010460", then disk 460 would be shut down before disk 010460 was started up.)

STARTUP OF PRIMOS

Once PRIMOS II is started and running, PRIMOS (Section 20) can be started.

INITIAL PRIMOS II OPERATING SESSION

Attaching to UFD

After PRIMOS II is started up, the user must attach to a User File Directory to execute PRIMOS II commands and create or manipulate files. The MFD has a password to discourage casual or inadvertent use of this important directory. (At MAKE time, a default password of XXXXXX is assigned.)

Caution

Do not attach to the MFD for program development or normal file handling tasks. Be very careful in entering commands while attached to the MFD. FIXRAT can be used to fix the disk if the UFDs DOS, BOOT, or CMDNC0 are intact; but if any of the files in the MFD are damaged, the master disk is compromised.

USING PRIMOS II

PRIMOS II and its supporting software allow the user to:

- Format disks and partitions using MAKE.
- Check file system integrity with FIXRAT.
- Backup disks with COPY.
- Create files and programs (Editor).
- Perform magnetic tape operations with MAGSAV and MAGRST.
- Perform file manipulation with FUTIL (copying, deleting, etc.).

The PRIMOS II commands are described in Reference Guide, PRIMOS Commands (FDR3108); other manuals to be consulted are those for the specific language users and the New User's Guide to EDITOR and RUNOFF (FDR3104).

Note

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 the shared editor.

Some commands use all of memory and will not run under PRIMOS II (e.g., RUNOFF).

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 selector switch to STOP/STEP.
2. MASTER CLEAR.
3. Set sense switches 1, 2, 3, and 4 up. This sets the sense switches to the restart address for the specific version of PRIMOS II being used.
4. Turn selector switch to LOAD.
5. START.
6. Reset sense switches to 0.

SHUTDOWN

Before terminating an operating session with PRIMOS II by loading another operating system or turning off power, enter the following command:

OK: SHUTDN

The SHUTDN command writes, to the disk, data that is buffered in memory.

Changing Disk Packs

To change removable disk packs, shut down PRIMOS II as above. Then power down the disk drive and replace the pack. If PRIMOS is the system in control, SHUT DOWN the physical disk with the SHUTDN command, then cycle down the disk. Restart the disk drive. When the unit is READY, give the STARTUP command appropriate to the operation with the new pack, and continue typing commands.

Turning Power Off

After a shutdown, the CPU can be used to run other software or power can be turned off. The following power-down order is recommended:

1. Disk drives

2. ASR, high-speed tape unit, and other peripheral devices

3. Prime CPU

RESTARTING PRIMOS II

A typical procedure to restart PRIMOS II after a shutdown is:

1. Turn on power and boot PRIMOS II into control.

2. ATTACH to a UFD and resume operation.

Pitfalls to Avoid

Some users are accustomed to keeping the PRIMOS II restart location in the sense switches so that it can be easily restarted. With PRIMOS II, this is not recommended, as it requires Sense Switch 1 to be SET. Setting Sense Switch 1 may produce undesired results with the FTN, PMA, and MDL commands.

SECTION 20

PRIMOS

OVERVIEW OF PRIMOS

PRIMOS is the operating system for the Prime 350 and up computers. It contains the code which manages:

- Time-shared access for up to 63 users
- Segmented virtual address space for programs up to 32 megabytes per user
- Input/output control
- File system
- Interactive terminal access and phantom user non-interactive jobs
- Communications systems

In addition, utilities, such as SEG, and languages, such as FORTRAN, are brought into user memory as needed.

BUILDING PRIMOS

PRIMOS Directory Organization - PRI400

The following conventions are designed to make it easier to perform installations, updates, and other maintenance activities upon the library directories that contain PRIMOS.

Naming Conventions: Program source files are named x.lang, where x is the name of the procedure contained in the file (the first such if there is more than one), and lang is FTN for FORTRAN, PMA for Assembler, and PL1 for PL/P or PL/I. In this way, it is easy to answer the question, "Which sources in this directory are written in PL/I?"

The binary file corresponding to a source file x.lang is named x.BIN.

The listing file, if any, corresponding to source file x.lang is named x.LIST.

An insert file used by a source file x.lang is named y.ins.lang, where y is an appropriately descriptive name. Command files are named C_y, where y is an appropriately descriptive name. Maps are named M_y, where y is an appropriate name.

Library Directory Hierarchy: The structure of the hierarchy containing PRIMOS modules is:

top_level_dir	contains command, system image files (*COLDS, PRnnnn, etc.), and utility runfile files (*MAPGEN, etc.)
KS	contains kernel source files
FS	contains file system source files
NS	contains network subsystem source files
CS	contains communications subsystem sources
R3S	contains ring 3 operating system sources
SE	contains DPTX devices support sources
PLPLIB	contains PLP library sources used by operating system
OBJ	contains all object modules

No command file should contain dependencies on a particular name or location within the hierarchy of top_level_dir.

PRI400>INSERT: All $INSERT files which are used in compiling or assembling source programs are in PRI400>INSERT. Source program statements which were previously:

```
$INSERT DEVCOM
```

are now

```
$INSERT *>INSERT>DEVCOM.INS.xxx
```

where xxx is FTN, PMA, etc.

If PRIMOS source programs are to be compiled or assembled in a directory other than PRI400, a subdirectory INSERT must exist in the present home directory and must contain any $INSERT files required.

PRI400>UTILS: The sources for certain utilities used by PRIMOS are in PRI400>UTILS. These include PRIMOS (the PRIMOS preloader), MAPGEN (the page map and cold start image generator), and the version of VPSD that is loaded with PRIMOS for debugging purposes. The command files for generating these utilities are found in PRI400.

Command File Usage

To build PRIMOS, attach to UFD PRI400 and run the command files which load the source binaries creating the memory image files.

1. Run C_R3LOAD which uses the Ring 3 source binaries in directory OBJ to create the memory image for segment '13 (PR0013) and segment '6002 (PR6002) and the load map M_RING3.

2. Run C_LOAD. C_LOAD uses the Ring 0 source binaries in directory OBJ to create the rest of the memory image files and load map M_RING0. It then calls C_COLD, which calls the program *MAPGEN to create page maps and the cold start memory image *COLDS.

3. Run C_COPY in UFD PRIRUN to copy the completed system from the master disk UFD PRI400 to the system disk UFD PRIRUN.

If it is necessary to recompile (or reassemble) all source modules, run the command file C_ALL. C_ALL performs the compilations and assemblies and then calls C_R3LOAD and C_LOAD to complete loading of the system.

MODIFYING PRIMOS

Almost all of the changes that would normally be made to PRIMOS operations can be done with the commands of the CONFIG configuration data file.

It is also possible to modify the runtime version of PRIMOS, but it is not generally recommended. If patching PRIMOS, be sure to make copies of the files to be modified <u>before</u> attempting any modification.

The running version of PRIMOS can be patched using VPSD (see below). From the supervisor terminal type OPR 1 and VPSD, perform the patches, exit from VPSD, and type OPR 0. Only segments in DTAR0 are accessible. This procedure can also be performed at a user terminal by first allowing read/write access to the segment to be patched using the SHARE command.

PRIMOS can be patched from the disk by attaching to the appropriate UFD and restoring the file PR00xx, where <u>xx</u> is the segment number to be patched. Do a PM to determine the starting and ending addresses and run PSD (or any version of PSD that is loaded above the ending address of PR00xx). After patching the file, save it and run the command file C_COLD to build a new memory image. The patches will become effective on the next cold start of PRIMOS.

An alternative method of patching PRIMOS is to generate a runfile which contains just those locations to be patched (using the PMA ABS and ORG pseudo-operations). This file can be restored over an existing segment by using the SHARE command.

VPSD for Kernel Debugging

The PRIMOS operating system kernel has a built-in version of the VPSD debugger. It is loaded as part of segment number 4, and a toehold to enter it is located at '6000 in segment 14. The toehold serves to enter 64V mode and load DTAR0 before transferring control to VPSD. After an operating system crash, the machine can be master-cleared, '6000 entered in the switches, and the START switch depressed in LOAD mode. VPSD will be entered and will be able to access any kernel segment. Segments not in descriptor table 0, however, cannot be directly accessed by VPSD.

This version of VPSD is usable only if the pages of segment 4 that contain VPSD are wired (made non-pagable, also known as locked). This may be done by uncommenting the WIRE VPSD directive to MAPGEN in the file C_COLD.

VPSD Baud Rate: The VPSD supplied as part of PRIMOS is set to run the supervisor terminal at 300 baud. In some cases, it may be desirable to change this rate. VPSD has three control words assembled into it that affect the baud rate of the supervisor terminal. The values of these three words can be patched if the supervisor terminal cannot run at 300 baud, or if a different baud rate is desired.

VPSD is loaded as part of segment number 4 starting at a word offset of '2000. The three words to patch start at '2004 in segment 4. The table below gives the values of these words for various baud rates:

	BAUD RATE	'2004	'2005	'2006
	110	110	27	74000
default	300	1010	76	34000
	1200	2010	373	34000
	9600	3410	3735	34000

These words can be patched from the control panel, or they can be patched after the system is brought up by using the SHARE command and the VPSD command.

SYSTEM CONFIGURATION

PRIMOS is delivered in a single version which can be configured at cold-start to run between 1 and 64 users. Each user may be configured to have a non-shared (DTAR2) virtual address space of up to 32 MBytes (256 segments). A limit of 64 MBytes (511 segments) of virtual address space is available to all users. CONFIG directives specify the number of configured users, the number of non-shared segments in each user's working area, and the total number of segments available for use in the system. (In the following discussion, the numbers used are decimal; all CONFIG directives require octal arguments).

The total number of users to be configured is specified by three CONFIG directives: NTUSR (number of terminal users), NPUSR (number of phantom users), and NRUSR (number of remote users). The sum of NTUSR, NPUSR, and NRUSR (which is NUSR) must be between 1 and 64. (User 1, the system user, must always be in the system.)

The number of segments available in each user's non-shared (DTAR2) address space is specified by the CONFIG directive NUSEG. It sets the size of each user's descriptor table for DTAR2. NUSEG may not exceed 256. The system has space for a maximum of 4096 DTAR2 SDWs for all users. Thus (NUSR*NUSEG) $\leq$ 4096.

The CONFIG directive NSEG sets the total number of segments available for use throughout the system. A number of segments (currently 17) are used for the operating system; each configured user requires one segment for the ring 0 stack. Thus, NSEG $\geq$ the sum of the segments used by PRIMOS and NUSR-1 (the ring 0 stack for user 1 is included in the segments used by PRIMOS). Since the system has space for 511 page maps, NSEG must not be greater than 511.

Paging Space Requirements

The total number of available segments (total virtual address space) of the system is limited by both the NSEG CONFIG directive and the amount of available paging space on the paging device. It is the minimum of the:

- Number of segments that fit in the specified paging space (as specified by using the records parameter in the PAGDEV and ALTDEV CONFIG directives)
- Number specified in the NSEG CONFIG directive
- Default NSEG setting (192).

Thus, the number of records required for paging is given by the following equation:

RECORDS = [(St-(Sp+(NUSR-1)))*64+(NUSR-1)*8+P]*[records/page]

St total number of segments available

Sp number of segments used by PRIMOS

NUSR number of configured users

P number of pages used by operating system segments

(Sp+(NUSR-1)) is the number of segments used by the operating system and the configured users' ring 0 stacks. Subtracting this number from the total number of segments available (St) gives the number of segments available for users. Although each configured user requires one segment for ring 0 stacks, only eight records are actually used within the segment. The number of records used by all configured users (except user 1) for ring 0 stacks is (NUSR-1)*8.

The number of records/page is 1 for storage modules, 2 for fixed head disks, and 3 for moving head disks.

Supported Devices

PRIMOS supports 1 to 63 users plus the supervisor and a large number of peripheral devices including:

- Eight magnetic tape drives (MT0-MT7)
- Four line printers (PR0-PR3)
- Four card reader/punches (CR0-CR3)
- One paper tape punch/reader (PTP/PTR)
- Four SMLC lines (SMLC00-03)
- Eight disks (or storage modules)
- Eight diskettes

C_PRMO Template

The following is a template that can be used to create the command file C_PRMO. C_PRMO is the command file that, if present in CMDNC0, is used to bring up Revision 17.2 of PRIMOS.

The template below is incomplete; it must be completed by each Administrator to reflect the specific computer configuration at the site. For convenience, a copy of the template, C_PRMO.TEMPLATE is supplied in UFD PRIRUN. Make the appropriate changes to C_PRMO.TEMPLATE and copy it to CMDNC0 on command device 0 as C_PRMO with FUTIL. The information that must be supplied in the file is:

- The name of the CONFIG data file. This file should be named CONFIG (the de facto Prime standard name for this file).
- Local disk(s) to be added when PRIMOS is started up. (Some sites may need to specify more than one ADDISK command in this file.)

- AMLC lines and the speed at which they are to be set to when PRIMOS comes up. (Some sites may need to specify more than one AMLC command in this file.)

In addition, a site should include (at the end of this file) any commands necessary to bring up any separately priced (or other) software when PRIMOS is brought up (for example, DBMS, PRIMENET, etc.).

```
CONFIG -DATA                 /* specify CONFIG file after -DATA
ADDISK                       /* specify local disks to be added
AMLC TTY                     /* specify AMLC lines
OPR 1                        /* SHARE REQUIRES OPR 1
SHARE SYSTEM>ED2000 2000     /* SHARE the editor  -  ED
SHARE SYSTEM>S2050  2050 700/* SHARE FORTRAN LIBRARY
R SYSTEM>S4000
SHARE SYSTEM>K2014A 2014 700/* SHARE MIDAS LIBRARY
SHARE SYSTEM>K2014B 2014 700
R SYSTEM>K4000
SHARE SYSTEM>C2014A 2014 700/* SHARE COBOL LIBRARY
SHARE SYSTEM>C2014B 2014 700
R SYSTEM>C4000
SHARE SYSTEM>F2014A 2014 700/* SHARE FORMS LIBRARY
SHARE SYSTEM>F2014B 2014 700
R SYSTEM>F4000
SHARE 2014
OPR 0
PH BATCHQ>PH_GO              /* STARTUP BATCH MONITOR
PROP -START                  /* STARTUP SPOOLER
A CMDNC0
/* SET THE DATE AND TIME **********
CO TTY
```

PRIMOS SYSTEM INITIALIZATION

PRIMOS is started from PRIMOS II at the supervisor terminal.

The steps to get PRIMOS running are:

1. Turn on processor, disk drive, and all needed peripheral devices, and boot load PRIMOS II as explained in Section 2.

   <u>Note</u>

   Use PRIMOS II supplied with Rev. 17. Earlier versions may cause an error.

2. To initiate PRIMOS, issue the commands:

   A PRIRUN
   R PRIMOS

At this point the preloader (PRIMOS) attaches to UFD CMDNCØ and looks for the command file C_PRMO. If this file exists, it is opened and executable commands read from it, otherwise, the system prompts:

PLEASE ENTER CONFIG

The operator may enter either a single-line CONFIG command or the name of a configuration data file. For example,

CONFIG 21 11ØØ61 1146Ø 2ØØØ 511 6/5 7/2

or

CONFIG -DATA CONFIG

Note

No restrictions are placed on the configuration data filename. Prime recommends that the name CONFIG be used as a standard.

A complete example of system start up using the configuration data file will be found in Section 2.

After system startup, the message LOGIN PLEASE is printed at all user terminals and users may log in.

Caution

User terminals will not respond to commands until the SETIME command is given by the operator.

PRIMOS Memory Parity Tests

On a cold start, PRIMOS performs a simple memory test for all memory except that used by the memory tester. The test is nondestructive, and correct parity is stored in every cell upon completion. Each cell must be capable of reading two patterns Ø52525 and 125252.

System Crash (Memory Parity Halt)

A memory parity error can be recognized in two ways. First, in STOP/STEP mode the address displayed in the panel lights will point to the instruction following the entry point MEMPA_ in segment 4 (source file SEG4). Second, DSWSTAT, register '35in the real register file, can be examined. (The contents of DSWSTAT are described in the Reference Guide, Systems Architecture , PDR3Ø6Ø.) DSWSTATH will have bit 3 set (=1). When it has been determined that a memory parity error has occurred, the following steps should be taken. (See Appendix C for details.)

1. First determine the physical page number (PPN) in which the parity error occurred. When the machine halts, the PPN is in the A-register and the word number is in the B-register and the users whose page got the error is in the X-register. The A-register can be displayed by fetching location 1 (the "300 register set"). The B-register may be displayed by fetching location 2 and the X-register by fetching location 0.

2. It is next necessary to determine the owner of the virtual page in which the parity error occurred. Add the PPN to the location of MMAP, word number '2000 in segment 4, and display the contents of the resulting location. (For example, if the A-reg = '47, display location '2047 in segment 4.)

   If the contents of this location in MMAP are greater than '5200, the virtual page belongs to a user and can be safely mapped out of physical memory (made unavailable for paging). If the contents are less than '5200, the page is a supervisor page and cannot be mapped out.

   Note

   The actual virtual page number is also contained in DSWRMAH, register '34 in the real register file. DSWRMAH may, however, be invalid if the parity error occurred during DMX transfer. It is safer to check MMAP.)

3. If the parity error occurred in a user page, the page can be automatically mapped out by selecting RUN mode and pressing START. PRIMOS will map out the page in error and halt. A warm start can then be performed.

   It is also advisable to inform the user whose page was mapped out that that user cannot continue after the warm start. The user whose page got the error is displayed in the X-register following a memory parity halt.

   If the parity error occurred in a supervisor page or if it is suspected that other locations in memory may be bad, it may be desirable to run the 'memory scanner'. This is a program that scans memory for parity errors and displays the location of each bad word found. To run the memory scanner after a parity error, MASTER CLEAR and START at location '777 (load '777 into location 7) with the ADDRESS/DATA switch on DATA and all sense switches down. When the display in the lights changes, the scanner has found a parity error. The panel display with all switches down is the word number of the parity error. The physical page number (PPN) is displayed by raising switch 15. The contents of the location in error are displayed by raising switch 14 (switch 15 down). To continue the memory scan, depress any key on the supervisor terminal. The scanner will continue scanning for the next error. The scanner will halt when the end of memory is reached.

After the memory scanner has been run, bad page(s) cannot be automatically mapped out -- this must be done by hand as follows:

1. As described above, add the PPN (displayed by the scanner when switch 15 is raised) to the location of MMAP, '2000 in segment 4.

2. Note the contents of this location and then set the location to -1 ('177777).

3. The previous contents of this location are a word-number of a pagemap entry, also in segment 4. The contents of this location should be set to 0 (meaning that the copy of the page in memory is no longer valid).

4. A warm start can then be performed.

Note

> If the computer entry address is below '2000, the bad page belongs to the supervisor and PRIMOS cannot be warm started.

Warm Start

After a successful warm start, the message ***** WARM START ***** is sent to all terminals, and all users are automatically restarted (e.g., typing START is not necessary). Because of this procedure, a warm start will not always cure a hung system. In general, do not attempt a warm start until after the system has halted.

Permanent Bad Page

If a page is known to be permanently bad, PRIMOS must be modified and saved with the appropriate pages marked 'unavailable'. For example, if page number 177 is known to contain a bad cell:

MMAP + 177 = 22000 + 177 = 22177

From PRIMOS II:

```
OK: A CMDNC0
OK: REST PRIMOS
OK: PM
SA, EA, P, A, B, X, K:
60 64777 1001 0 0 0 2000
OK: S 56000
$ U 22177 777777
22177 000000 177777
$ R 7000
OK: SAVE PRIMOS 60 64777 1001 0 0 0 4000
OK:
```

PRIMOS HALTS

Locations at which PRIMOS halts are defined in the load map M_PRMOS. This map is in the UFD in which PRIMOS was built and from which it is normally RESUMEd. In general, it is not possible to accurately determine the cause of a halt condition by observing only the address displayed in the panel lights in STOP/STEP mode. This is because the halt may have occurred in one of several different possible segments. To determine the segment in which the halt occurred, display the contents of PBH, the high side of register '14 in the current register set. (See Section 10 and Appendix C.) See Section 10 for a complete list of system halts.

APPENDICES

APPENDIX A

BOOTSTRAPS

INTRODUCTION

When power is first turned on, the semiconductor read/write memory of a Prime CPU is empty. To make the CPU useful, a program (preferably an operating system) must be loaded into memory from an external device such as a disk. This is done by Prime's Automatic Program Load (APL) feature which works when the control panel rotary switch is in the LOAD position. To start this process, a short bootstrap program contained in a Read-Only Memory (physically located on the control panel) takes control temporarily. It loads a more extensive bootstrap from the external device into main memory. Control then passes to this longer bootstrap, which completes the job of loading the memory image of the operating system from the external device.

This appendix describes Prime bootstraps: control panel bootstraps and second level bootstraps stored on paper tape, magnetic tape, disk, or storage module.

Note

> Certain of the devices described here are no longer supplied. The information is included here for reference and archival purposes.

CONTROL PANEL BOOTS

Control Panel Microcode

A control panel can have either 256 or 512 16-bit words of PROM from which bootstrap programs can be loaded into memory. After pressing MASTER CLEAR and dialing the selection switch to the LOAD position, press the START switch to cause the control panel microcode to read PROM locations '0 to '50 into memory locations '6 to '56 and begin execution in 16S mode at the address loaded into Location '7. This initial program, the preboot, can then read succeeding PROM locations into memory.

Prime Pre-Boot

The Prime pre-boot saves the A-register in location '57 and then selects among three classes of bootstraps and stores the appropriate code from the PROM into memory. The three classes of bootstraps are auto-start, paper tape, and mass storage boots. The user selects the desired boot by setting Sense Switches 14, 15, and 16 as follows (set=1, reset=0):

SS =	14	15	16	Code	
	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

Device Specific Boots

Auto-Start (0): enters 64R mode and jumps to the location specified in Sense Switches 1 to 10 ('100 to '177700). If no address is specified, a default of '1000 is used.

Paper Tape (1 and 2): Modifies itself for either ASR or high speed paper tape (by sense switches) and reads a second-level MDL boot into memory.

Mass Storage (3-7): performs further selection for fixed-head disk (FHD), moving head disk (MHD), magnetic tape, diskette, and spare, all of which are loaded by the pre-boot.

FHD (3): Sense Switch 13 is used to select between controller 4001 (SS 13 reset) and controller 4002 (SS 13 set). Sense Switch 12 is used to select between device addresses '21 (SS 12 reset) and '23 (SS 12 set). The boot reads record 0 (448 word PRIMOS record format) of the disk starting at location '770 and begins execution at '1000 (via a JST '777). This boot waits for the drive to come ready and retries on status errors:

SS =	12	13	
	x	0	4001
	0	1	4002, address '21
	1	1	4002, address '23

MHD (4): Moving head disks come in three varieties: two platter drives (6M-obsolete or 12M bytes), 20-surface drives (obsolete) and 40MB, 80MB or 300MB storage modules. Sense Switches 10, 11, 12, and 13 are used as follows (x - don't care):

SS =	10	11	12	13	
	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 the storage module, record 0 (448 words PRIMOS record format) of the selected surface of physical drive 0 is read into memory starting at location '770 and execution is begun at '1000 (via a JST '777). In the storage module, a 1040-word PRIMOS II record is read into memory starting at location '760 (to allow for the 16-word header) and execution is begun at '1000, as for other disk bootstraps. The moving-head disk and storage module bootstraps wait for the disk drive to become ready, and the bootstraps retry on the occurrence of status errors.

MT (5): Sense Switch 12 is used to select between 9-track (SS 12 reset) and 7-track (SS 12 set) magnetic tape drives. The boot starts up the drive, ensures that the tape is set at a loadpoint (space forward, abort, and rewind), and reads one tape record into memory from location '220 through '7777 (4K). Execution begins at '1000 (via a JST '777).

DISKETTE (Floppy) (6): Reads record 0 into memory, starting at location '770. To maintain IBM compatibility, the boot alternately tries to read a 448-word PRIMOS record and a 64-word IBM record. Execution then begins at '1000 (via a JST '777). This boot waits for the drive to come ready and retries on status errors.

SPARE (7): Intended as a user-supplied down-line loader. Currently, halts at location '57.

Key-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.

SECOND-LEVEL DISK BOOTS (BOOT)

The bootstrap, placed on all devices by the MAKE utility, is the same for all devices. The bootstrap is 2 records on all storage devices. Using CPBOOT, the control panel loads the first record of the bootstrap into memory; the first record is a disk input routine that loads the remainder of the bootstrap. BOOT then initializes the supervisor terminal and types PHYSICAL DEVICE=. After the operator enters the physical device, BOOT attempts to find and load the appropriate version of DOS (PRIMOS II) into memory and transfers control to it. The source is BOOT, residing in UFD FILAID on the master disk. The run file on records 0 and 1 is also called BOOT and resides in the MFD.

Note

At Revision 15 and after, only *DOS64 is supplied; all other versions of DOS are obsolete. They are included here for archival purposes.

BOOT OPERATION

BOOT performs the following instructions:

- Cleans up parity, non-destructively, throughout memory
- Sizes available memory
- Requests, from the operator, which device to boot from
- Attaches to the MFD on that device
- Attaches to UFD DOS
- Depending upon memory size and/or sense switches, reads *DOS16, *DOS24, *DOS32, or *DOS64 into memory and starts it

There are three possible outcomes of a boot operation: a successful boot, in which case PRIMOS II takes control; a detected error, in which case the boot returns to request which device to boot from; or an undetected error, such as non-existent device, in which case the boot hangs.

When started, the boot types:

PHYSICAL DEVICE=

The operator response should be the physical device number constructed according to the tables in Appendix D. Which PRIMOS II is read in is determined by either memory size of Sense Switches 1, 2, and 3 in the following manner:

If all sense switches are reset (0), the highest memory PRIMOS II that will fit in available memory is read.

If any of the sense switches are set (1), they are treated as the most significant bits of the high address of memory +1. *DOS64 cannot be forced by setting sense switches.

Sense Switch 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=don't care)	*DOS32 (obsolete)

Once the boot has been successfully brought into memory by the control panel boot, it can be re-executed by MASTER CLEARing and STARTing at '1000. If a status error is detected on the device, BOOT will restart automatically. All drivers will wait for the device to come ready.

Errors

Errors detected while loading BOOT using its own first record cause a halt with an error code in the control panel data lights. The errors checked and put into the lights at this stage are:

Octal Number in Lights	Error
100	Parity
101	Machine check
102	Non-octal 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 and Machine Check Error, 100, 101: If a parity or machine check error occurs while loading the BOOT program itself, then a halt will occur with the code 100 or 101 respectively in the control panel data lights. Parity and machine check errors are caught by the hardware. No further information is available on the P100, P200 or P300. Addition information can be found in the diagnostic status word on the higher CPUs. After the memory test, the error messages, PARITY ERROR, or, MACHINE CHECK, will be printed. If the errors persist, the messages persist.

Non-octal Physical Device Number, 102: The message, OCTAL ONLY, will be printed if the user enters a non-octal character for the physical device number. The PHYSICAL DEVICE= prompt is issued again at the supervisor terminal.

Bad Device Type, 103: The bad device type code will appear 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, loading the BOOT program itself, a halt then occurs with the code 104 in the control panel data lights. If a bad status is detected while loading DOS, the message BAD STATUS is printed followed by the status word.

Bad Record Id, 105: As each record is read, the record address requested is checked against the address of the record read as found in the record itself. If these addresses do not match, then a halt will occur with the code 105 in the control panel data lights. The requested address is in locations '723 and '724 and the address in the record is in locations '760 and '761.

When searching for or loading DOS, a message will be printed BAD RECORD ID, rrrrrr rrrrrr ffffff ffffff, where rrrrrr are two words of requested octal address and the ffffff are two words of found 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 one 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, then the BOOT program will halt with a '106 in the data lights.

'FILE' Not Found: If the required version of DOS or the DOS UFD is not on the requested physical device, the the message, 'FILE' NOT FOUND, will be printed, where 'FILE' is the name of the requested file. The PHYSICAL DEVICE= prompt is isssued again at the supervisor terminal.

Memory Test Failure: While testing the memory, if the test pattern written and that read do not match, then a message will be printed, MEM TEST MISMATCH LOC xxxxxx, where xxxxxx is the location of the word being tested. During memory test, if either a parity error or a machine check is detected, then the address of the word being tested will be printed followed by the message PARITY ERROR or MACHINE CHECK. The PHYSICAL DEVICE= prompt is issued again at the supervisor terminal.

Building BOOT

The BOOT program is stored as a normal PRIMOS II SAVE file on a normal PRIMOS II format record (=0). Consequently, physical record 0 contains first an 8-word PRIMOS II record header, second a 9-word PRIMOS II SAVE file header, and the BOOT program itself. On a non-storage module device the bootstrap occupies 2 records. The 8-word PRIMOS II record header is eliminated by reading the record starting at '770 but starting execution at '1000, the first word of the 9-word SAVE file header. For the storage module, the PRIMOS II record header is 16 words long, but the record is read starting at location '760.

The SAVE file header is as follows:

```
word 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 the boot is guaranteed to be executed in 16S mode either by MASTER CLEAR or the control panel boot). The 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 a 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

> The bootstrap can be changed to settings for other terminal characteristics either permanently (by SAVEing it with new B, X, and Keys parameters) or temporarily (by patching locations '1004, '1005, and '1006 of the copy in memory, followed by a MASTER CLEAR and RUN - see Appendix F). When changing any of these values, care must be taken not to change any of the others.

The run-file of the boot will be properly placed on a disk by the MAKE program. Also, the file *BOOT will be 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, and hence can be restored and saved, just as the file BOOT in an MFD.

Bringing Up a Pack that Will Not Boot - Control Panel

If the system pack will not boot but the remainder of the pack is intact, use the following procedure to boot from magtape:

- Load PSD into memory (necessary to load PRIMOS II):

    1. Mount the magsaved tape on drive 0.

    2. Put the tape drive on line.

    3. Set the sense switches to '505 (device + magnetic tape)

    4. Set the rotary switch to LOAD, to load PSD into memory.

    5. Press START. When the system types "TREENAME:", enter the treename exactly as it appears on the index of the magtape. Typically:

        TREENAME: CMDNC0>PSD

    6. The system will either halt at '624 or '174564 with a one in the data lights.

- Load PRIMOS II into memory:

    7. Set the rotary switch to STOP.

    8. Press master clear.

    9. Set the rotary switch to LOAD.

10. Press START. When the system types "TREENAME:", enter the treename exactly as it appears on the index of the magtape. Typically:

    TREENAME: DOS>*DOS64

11. The system halts at either '624 or '174564 with a one in the data lights.

- Use PSD to relocate PRIMOS II:

12. Set the rotary switch to STOP.

13. Press master clear.

14. Reset the sense switches.

15. Set the sense switches to '6000, the first executable instruction of PSD.

16. Set the rotary switch to LOAD.

17. Press START. PSD will respond with its prompt character, $. Use the PSD copy command to move PRIMOS II:

    $ C 10000 57541 130000
    $

- Start PRIMOS II

18. Set rotary switch to STOP.

19. Press master clear.

20. Set the sense switches to '170000, the first executable instruction of PRIMOS II.

21. Set the rotary switch to LOAD.

22. Press START

- Attach to the system pack:

23. When PRIMOS II types "OK": enter the physical disk pack name and attach to its MFD:

    OK: STARTUP nnn

    OK: A MFD xxxxxx

Bringing Up a Pack that Has Crashed - Control Panel

If the system pack has crashed, use the following procedure to reload and boot from magtape:

- Load PSD into memory (necessary to load PRIMOS II):

  1. Mount the magsaved tape on drive 0.

  2. Put the tape drive on line.

  3. Set the sense switches to '505 (device + magnetic tape)

  4. Set the rotary switch to LOAD, to load PSD into memory.

  5. Press START.. When the system types "TREENAME:", enter the treename exactly as it appears on the index of the magtape. Typically:

     TREENAME: CMDNC0>PSD

  6. The system will either halt at '624 or '174564 with a one in the data lights.

- Load PRIMOS II into memory:

  7. Set the rotary switch to STOP.

  8. Press master clear.

  9. Set the rotary switch to LOAD.

  10. Press START. When the system types "TREENAME:", enter the treename exactly as it appears on index of the magtape. Typically:

      TREENAME: DOS>*DOS64

  11. The system halts at either '624 or '174564 with a one in the data lights.

- Use PSD to relocate PRIMOS II:

  12. Set the rotary switch to STOP.

  13. Press master clear.

  14. Reset the sense switches.

  15. Set the sense switches to '6000, the first executable instruction of PSD.

16. Set the rotary switch to LOAD.

17. Press START. PSD will respond with its prompt character, $. Use the PSD copy command to move PRIMOS II:

```
$ C 10000 57541 130000
$
```

- Bring MAKE into memory:

18. Set rotary switch to STOP.

19. Press master clear.

20. Set the sense switches to '505 (device + magnetic tape).

21. Set the rotary switch to LOAD, to load MAKE into memory.

22. Press START. When the system types "TREENAME:", enter the treename exactly as it appears on the index of the magtape. Typically:

```
TREENAME: CMDNC0>MAKE
```

23. The system will either halt at '624 or '174564 with a one in the data lights.

- Start PRIMOS II and then MAKE to format the disk:

24. Set the rotary switch to STOP.

25. Press master clear.

26. Set the sense switches to '170000, the first executable instruction of PRIMOS II.

27. Set the rotary switch to LOAD.

28. Press START. PRIMOS II will respond with the prompt "OK":. Enter "S 1000" to start MAKE and then, when MAKE requests it, enter the physical disk number:

```
OK: S 10000
MAKE XX.X
PHYSICAL DISK: nnn
OK:
```

- LOAD MAGRST to restore the master disk:

29. Set the rotary switch to STOP.

30. Press master clear.

31. Set the sense switches to '505 (device = magtape).

32. Set the rotary switch to LOAD.

33. Press START. Then the system types "TREENAME=", enter the treename exactly as it appears on the index of the magtape. Typically:

    TREENAME= CMDNCO>MAGRST

34. The system with either halt at '624 or '174564 with a one in the data lights.

- Restore the master disk:

35. Set the rotary switch to STOP.

36. Press master clear.

37. Set the sense switches to '170000, the first executable instruction of PRIMOS II.

38. Set the rotary switch to LOAD.

39. Press START. PRIMOS II will respond with the prompt OK:. Enter the disk device number and then attach to the MFD on that disk and start at location '1000:

```
OK: STARTUP nnn
OK: A MFD XXXXXX
OK: S 1000
GO
REV XX
   .
   .
   .
```

Note

Booting from tape fails if any read errors are encountered. To minimize the change of tape errors, save the tape with the minimum number of required utilities:

```
CMDNC0>MAKE
CMDNC0>PSD
CMDNC0>MAGRST
DOS>*DOS64
```

Bringing Up a Pack that Will Not Boot--Virtual Control Panel

If the system pack will not boot but the remainder of the pack is intact, use the following procedure to boot from magtape:

1. Press master clear.

2. Set the addressing mode to absolute:

   ```
   CP> MO ABS
   ```

3. Boot PRIMOS II from magtape:

   ```
   CP> BOOT 505
   TREENAME:  DOS>*DOS64
   HALTED AT 000624: 000001
   ```

4. Relocate PRIMOS II:

   ```
   CP> C10000 57541 130000
   ```

5. Start PRIMOS II:

   ```
   CP> BOOT 170000
   OK: STARTUP physical-device-no.
   OK: A CMDNC0
   ```

Bringing Up a Pack that Has Crashed - Virtual Control Panel

If the system pack has crashed, use the following procedure to reload and boot from magtape:

1. Press master clear.

2. Set the addressing mode to absolute:

   ```
   CP> MO ABS
   ```

3. Boot PRIMOS II from magtape:

```
CP> BOOT 505
TREENAME: DOS>*DOS64
HALTED AT 000624: 000001
```

4. Relocate PRIMOS II:

```
CP> C 10000 57541 130000
```

5. Load and run MAKE:

```
CP> BOOT 505
TREENAME: CMDNC0>MAKE
HALTED AT 175164: 000001
CP> BOOT 170000
OK  S 1000
MAKE XX.X
    :
    :
OK:  (ESC ESC)
CP> STOP
```

6. Restore the disk from tape:

```
CP> BOOT 505
TREENAME  CMDNC0>MAGRST
HALTED AT 000624: 000001
CP> BOOT 170000
OK  A MFD XXXXXX
OK: S 1000
MAGRST XX.X
TAPE UNIT
    :
    :
```

APPENDIX B

VIRTUAL CONTROL PANEL

This appendix describes the Virtual Control Panel as of Software Revision level 17.2. 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, similar to 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, interogating and setting sense switches, 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.

It replaces the System Option Controller (SOC) and also 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	Initialize system	Stop (processor halted)
Remote enable	Permit remote access to system via VCP	Enabled
Remote privilege	Select 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 old control panel and adds flexibility and ease of use by displaying and accepting data from a terminal. Many control panel functions are made to look like a software debugger. Data may be accepted and displayed in hexadecimal, ASCII, binary, decimal, and octal. Commands are entered with key words instead of 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 failed 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 (2 escapes). 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

Anywhere 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 Ø.

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.

▶ 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: 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).

▶ 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: Allow accessing 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 in given in Table B-2

▶ A n

Access address n. The address may be followed by data display specifiers.

▶ 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 repsonses 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 a 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.

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: SDW2/Address of page map
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

Exiting Control Panel Mode: is accomplished by 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).

EXAMPLES

VCP use in a few common situations is shown below

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 II type:

CP> SYSCLR

CP> BOOT 114 for a storage module

The message:

PHYSICAL DEV=

will appear. Proceed as described in Sections 2 and 19.

Warm Start

If the CPU halts, the message:

```
HALTED AT:    halt-address

CP>
```

will appear. 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 Section 10.

Dump Register File

```
CP> MO RFABS
CP> MO RFH
CP>  O  O
CP> D 0 177

 (RFH is dumped)

CP> MO RFL
CP> D 0 177

 (RFL is dumped)
```

Tape Dump

Mount a scratch tape on unit 1, and type the following:

```
CP> SYSCLR
CP> RUN 776
```

APPENDIX C

CONTROL PANEL

This appendix describes operation of the Prime central processors (CPU) 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 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.

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.

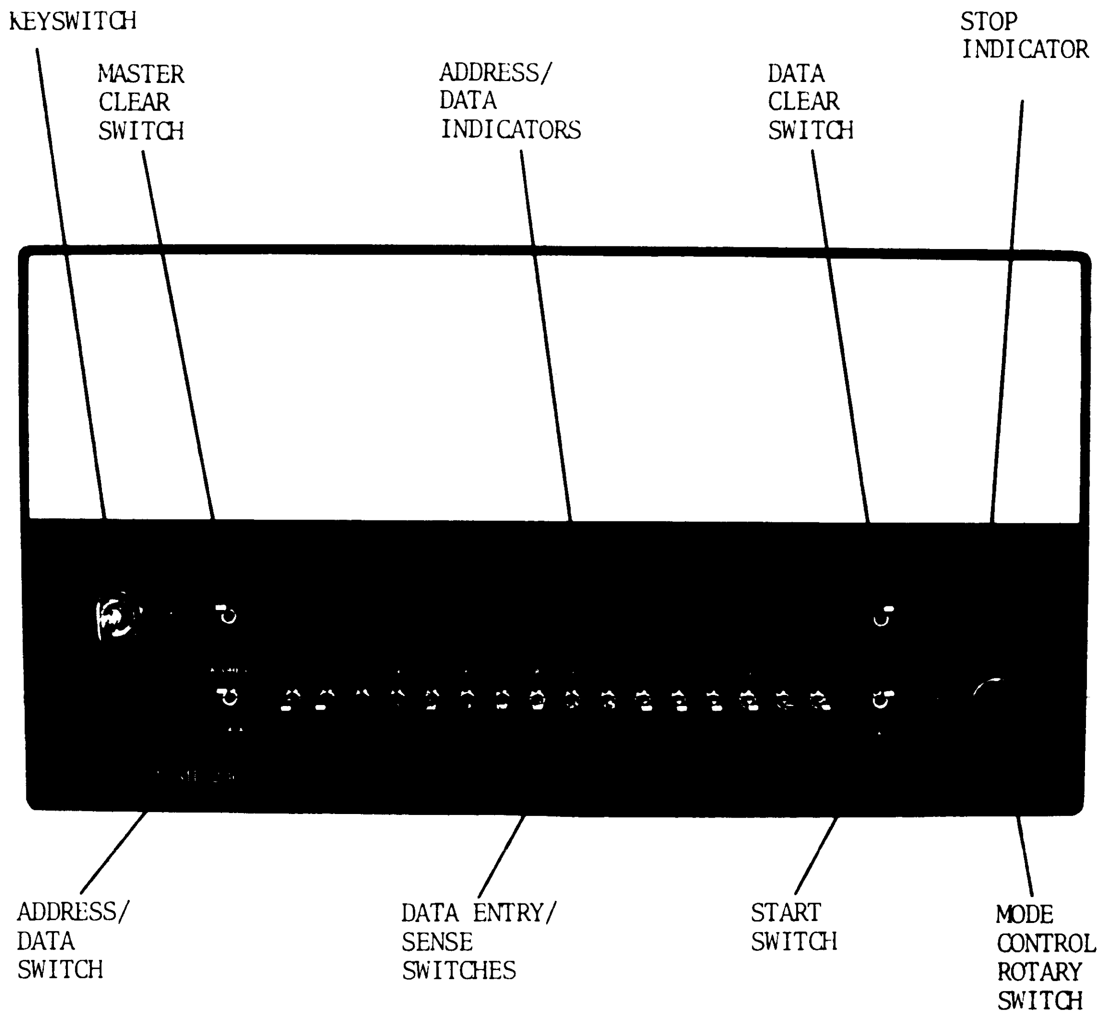


Figure C-1. Prime CPU Controls and Indicators

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 a 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.

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 ADDRESS/DATA switch functions normally).	
MASTER CLEAR Switch	Momentary-down switch sets CPU registers and status to known starting condition (see text entitled Master Clear).	
Address/Data Indicators	What these indicators display depends on the 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 '172Ø)
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 or STORE	Address of next cell to be accessed	Data last set up by data switches, or FETCH or STORE
FETCH Y+1 STORE Y+1	One less than address of next cell to be accessed (address is incre mented 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	
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.
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
START Switch	Initiates operation set up by rotary switch (RUN, LOAD, FETCH, STORE).
Sense/Data Switches	Bank of 16 latching-up/momentary-down toggle switches
	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).*
	Center and latching UP position provide sense switch data to INA '162Ø and sense switch skip instructions (SNS, etc.). Also provide device identification for bootstrap LOAD opration. UP is a 1, center is a Ø for each switch. Switches 1 to 16 correspond to A register bits 1 to 16.
	For P35Ø and up, 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).
A DDRESS/DATA Switch	Selects information to be displayed on data/ address indicators. (See description of indicators.)

*In RUN mode, momentaries can be program tested by INA '172Ø

Table C-2. Register and Memory Access in Prime 350 and up

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

SUMMARY OF OPERATING PROCEDURES

Note

The following procedures apply to all CPUs unless otherwise specified.

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 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 0's).

- The P register is set '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-verificaton 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 (P1ØØ-2ØØ-3ØØ)

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 100 200-300)

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 onthe 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.

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.

Table C-4. Prime CPU Register Correspondence

P35Ø-4ØØ-5ØØ Register Location	P1ØØ 2ØØ 3ØØ 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	Ø	X (index)
1Ø H	13	-
1Ø,11		FARØ,FLRØ (field address and length register Ø)
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)
2Ø H	1Ø	(high half of DTAR3)
2Ø H,L	-	DTAR3 (descriptor table address segments 3072 4Ø95)
21 H,L	-	DTAR2 (segments 2Ø48 3Ø71)
22 H,L		DTAR1 (segments 1Ø24-2Ø47)
23 H,L	-	DTARØ (segments Ø 1Ø23)
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)
3Ø H	-	process 1Ø24-microsecond c.p.u timer

Note

1. In mapped access for P35Ø-4ØØ-5ØØ, the P1ØØ-2ØØ-3ØØ 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.

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 oprating systems from external storage devices. Operating procedures are described below. See Appendix A for the details of internal operations of Prime's APL bootstraps.

1. Install the medium from which a program is to be loaded on the appropriate device. Make sure power is on and both the device and the supervisor terminal are ready to operate.

2. Select the autoload device (and disk surface, if applicable) in SS11-16 as shown in Figure C-2.

3. Set SS1-4 to select the desired version of PRIMOS II (*DOS16, etc.). If SS1-4 are left at zero, the system automatically selects a version that will fit the available memory.

4. Place SS10 down unless you want to prevent automatic startup when bootstrapping is complete.

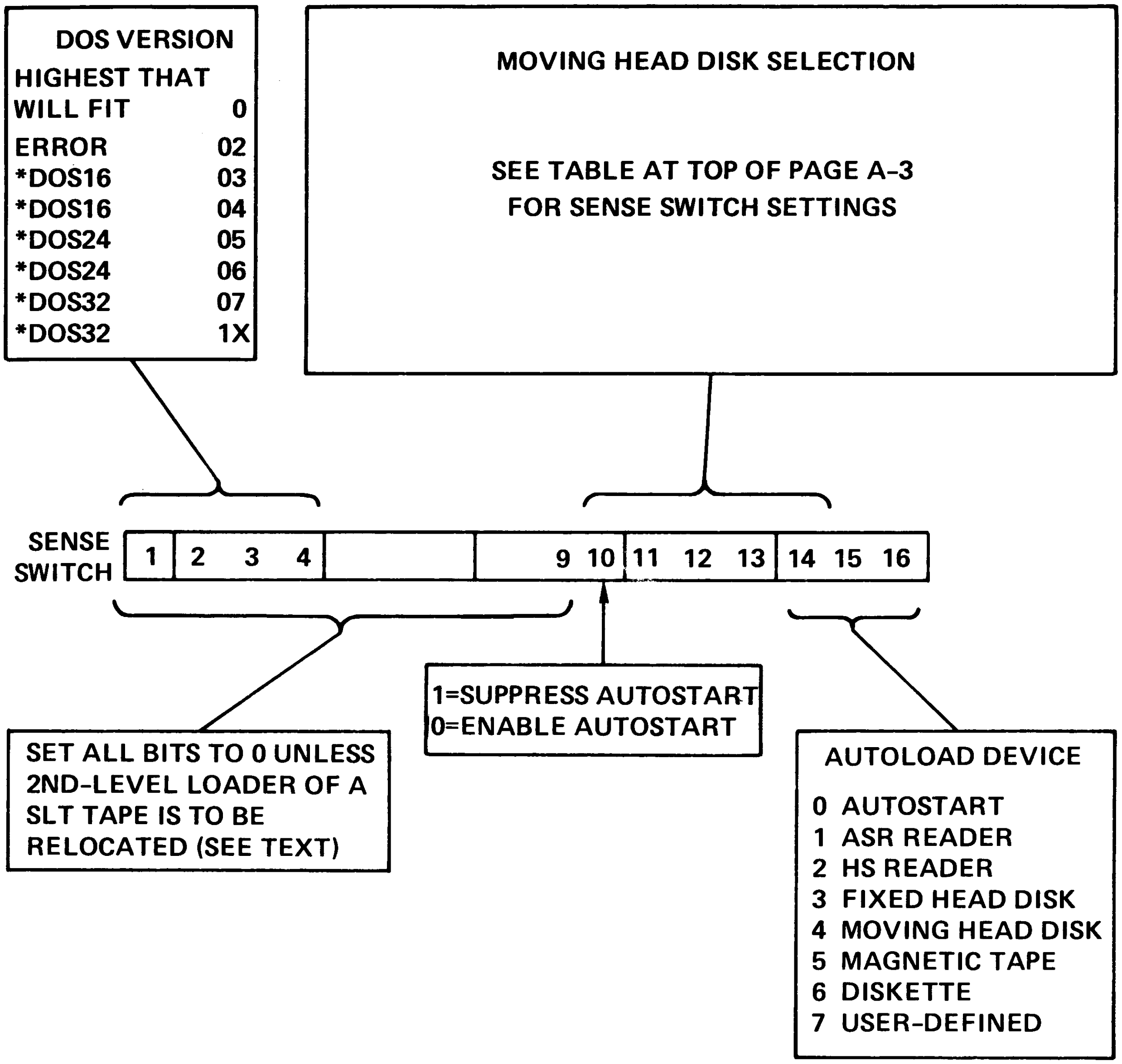


Figure C-2. APL Sense Switch Settings.

5. Press MASTER CLEAR

6. 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-3 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.)

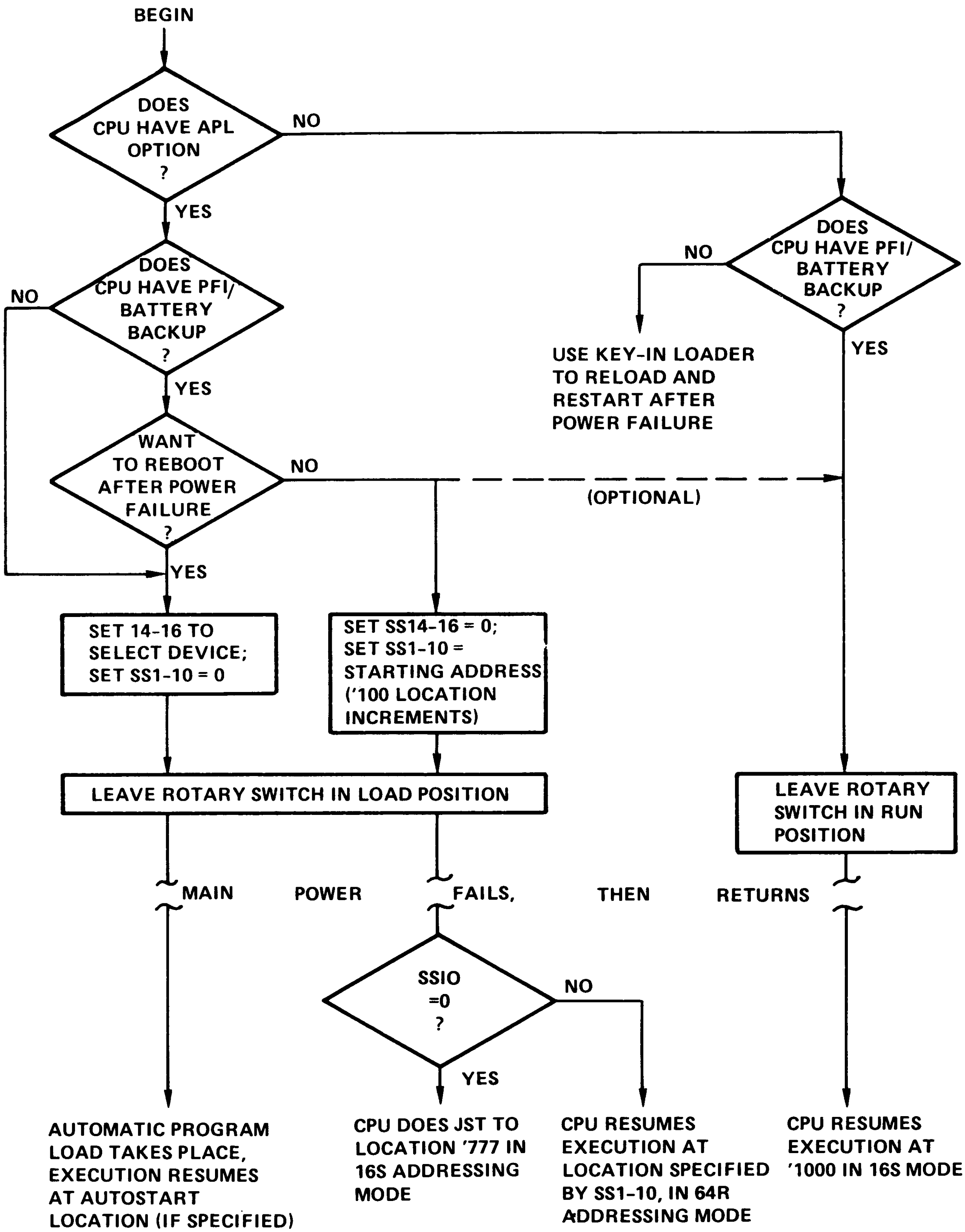


Figure C-3. Panel Setup for Automatic Restart or Reload after Power Failure.

APPENDIX 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 FIXRAT, MAKE, SHUTDN STARTUP, 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. 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 labelled 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. It is important to keep a record in the system log book of drive unit numbers and of the physical device number(s) (including partitioning) for disks mounted on these drives.

CONTROLLER

The controller is an interface between the CPU and the drive unit. The system installer will indicate which controller address has been used. In almost all cases the first four storage modules (or 60 MByte disks) will be connected to the default address; drives in excess of 4 will be connected to the explicit address.

DISKS

Fixed Head Disks

Physical device numbers are:

Sectors/Track	Physical Device Number
8	10
64	40

Diskettes (Floppy Disk)

All diskettes have 4 sectors/track and a total of 304 records (448 words/record). The physical device number is: 2u

where *u* is one less than the number of the diskette drive, (1 8) i.e., 24 is the fifth (unit 5) diskette drive.

Moving Head Disks - Cartridges

The physical device numbers are: 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. (These cartridges have 448 words/record.)

Sectors/Track	Type (t)	Records	Comments
8	0	3248	6MB cartridge (obsolete controller)
8	3	6496	6MB (obsolete 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.

Examples: 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)

This disk has 32 sectors/track and 64960 records (448 words/record).

The physical device number is: 5c5u

where 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) which the disk is mounted.

Example: A 60 MByte disk mounted on drive unit 2 connected to address '23 of the controller has physical device number 5254.

Partitioning: This 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

Examples

1. 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

   Physical device number is: 021450

2. 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 non-zero offset)

   y=4

   Physical device number is 005254, the same as constructed above for non-partitioned disk.

Moving Head Disks - Storage Modules

Storage modules exist in three sizes as below: (They have 1040 words/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 (sub-divided) 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.)

Bits	Meaning
1,2,3,4	One-half the head offset
5,6,7,8,16	Number of heads in partition
9	Controller address indicator. 0 if '26 (default), 1 if address '27 (explicit)
10,11,12,13	Device type. This is 6 (0110) for present storage modules
14,15	Drive unit number (0 - 3)

A complete list of valid physical device numbers is given in Table D-1.

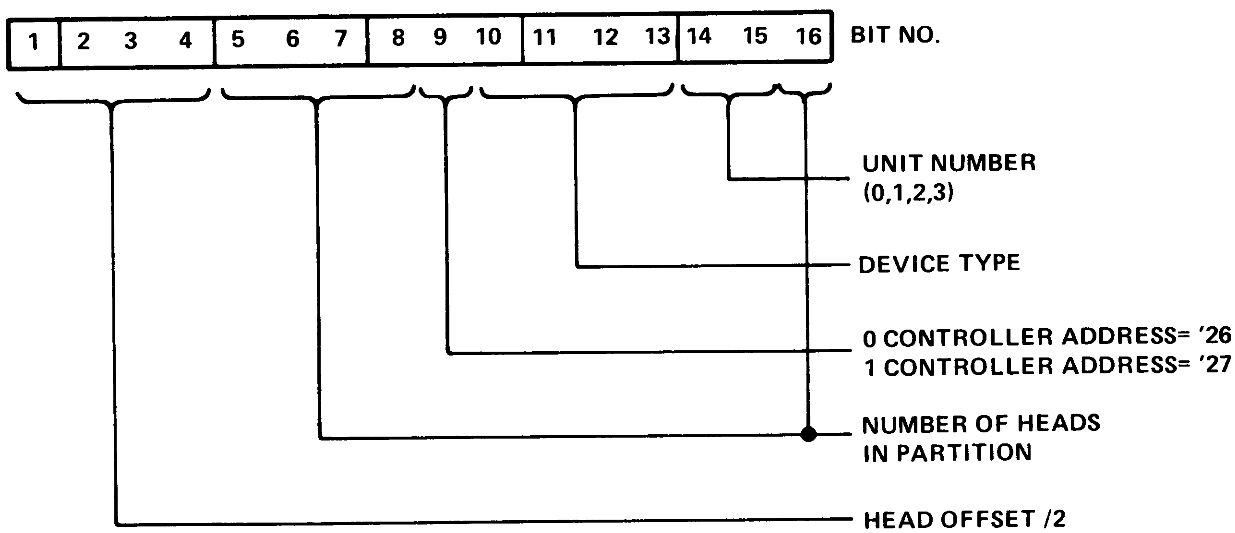


Figure D-1. Construction of Physical Disk/Disk Partition Number

Table D-1. Physical Device Numbers for Storage Modules

Number of Surfaces	Beginning Head Number Ø	2	4	6	8	1Ø	12	14	16	18
1	------	------	Ø2ØØ6z	------	------	------	------	------	------	11ØØ6z
2	ØØØ46y	Ø1Ø46y	Ø2Ø46y	Ø3Ø46y	Ø4Ø46y	Ø5Ø46y	Ø6Ø46y	Ø7Ø46y	1ØØ46y	
3	------	Ø1Ø46z	------	------	------	------	------	------	1ØØ46z	
4	ØØ1Ø6y	Ø11Ø6y	Ø21Ø6y	Ø31Ø6y	Ø41Ø6y	Ø51Ø6y	Ø6zØ6y	Ø71Ø6y		
5	ØØ1Ø6z	------	------	------	------	------	------	Ø71Ø6z		
6	ØØ146y	Ø1146y	Ø2146y	Ø3146y	Ø4146y	Ø5146y	Ø6146y			
7	------	------	------	------	------	------	Ø6146z			
8	ØØ2Ø6y	Ø12Ø6y	Ø22Ø6y	Ø32Ø6y	Ø42Ø6y	Ø52Ø6y				
9	------	------	------	------	------	Ø52Ø6z				
1Ø	ØØ246y	Ø1246y	Ø2246y	Ø3246y	Ø4246y					
11	------	------	------	------	Ø4246z					
12	ØØ3Ø6y	Ø13Ø6y	Ø23Ø6y	Ø33Ø6y						
13	------	------	------	Ø33Ø6z						
14	ØØ346y	Ø1346y	Ø2346y							
15	------	------	Ø2346z							
16	ØØ4Ø6y	Ø14Ø6y								
17	------	Ø14Ø6z								
18	ØØ446y									
19	ØØ446z									

y is twice the unit number (Ø – 3) of the drive unit on which the disk is mounted. z is twice the drive unit number plus one.

These are all the valid physical device numbers for the 4Ø, 8Ø, and 3ØØ MB disks To use this table first decide upon the number of surfaces in the partition and the head number of the first head in the partition. Then, look up the physical device number in the table. If the partition defined is not in this table, then it is not a legal partition.

All partitions must begin on an even head number. Only the last partition on the disk pack can have an odd number of surfaces.

The 4Ø MB and 8Ø MB storage modules have 5 heads (Ø - 4); 3ØØ MB modules have 19 heads (Ø - 18). Device numbers Ø2ØØ6z, Ø1Ø46z, and ØØ1Ø6z are valid for 4Ø MB or 8Ø MB modules only.

All physical device numbers in this table are for controller address '26. Add '2ØØ for devices connected to controller address 27.

There are 74Ø7 1Ø4Ø-word records in a storage module surface. This is equivalent to 17238 448-word records.

Example: A system contains three drive units; drives 0 and 1 have 300 MByte storage modules, drive 2 has an 80 MByte storage module. (See Figure D-2.) For program development, operations, testing, etc., the modules are to be partitioned as follows:

drive 0: partitions of 2 2,6,2,2,2, and 3 heads
drive 1: partitions of 14 and 5 heads
drive 2: partitions of 2 and 3 heads

The physical device numbers are:

<u>0</u>	<u>1</u>	<u>2</u>
000460	003462	000464
010460	071063	010465
021460		
050460		
060460		
070460		
100461		

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.

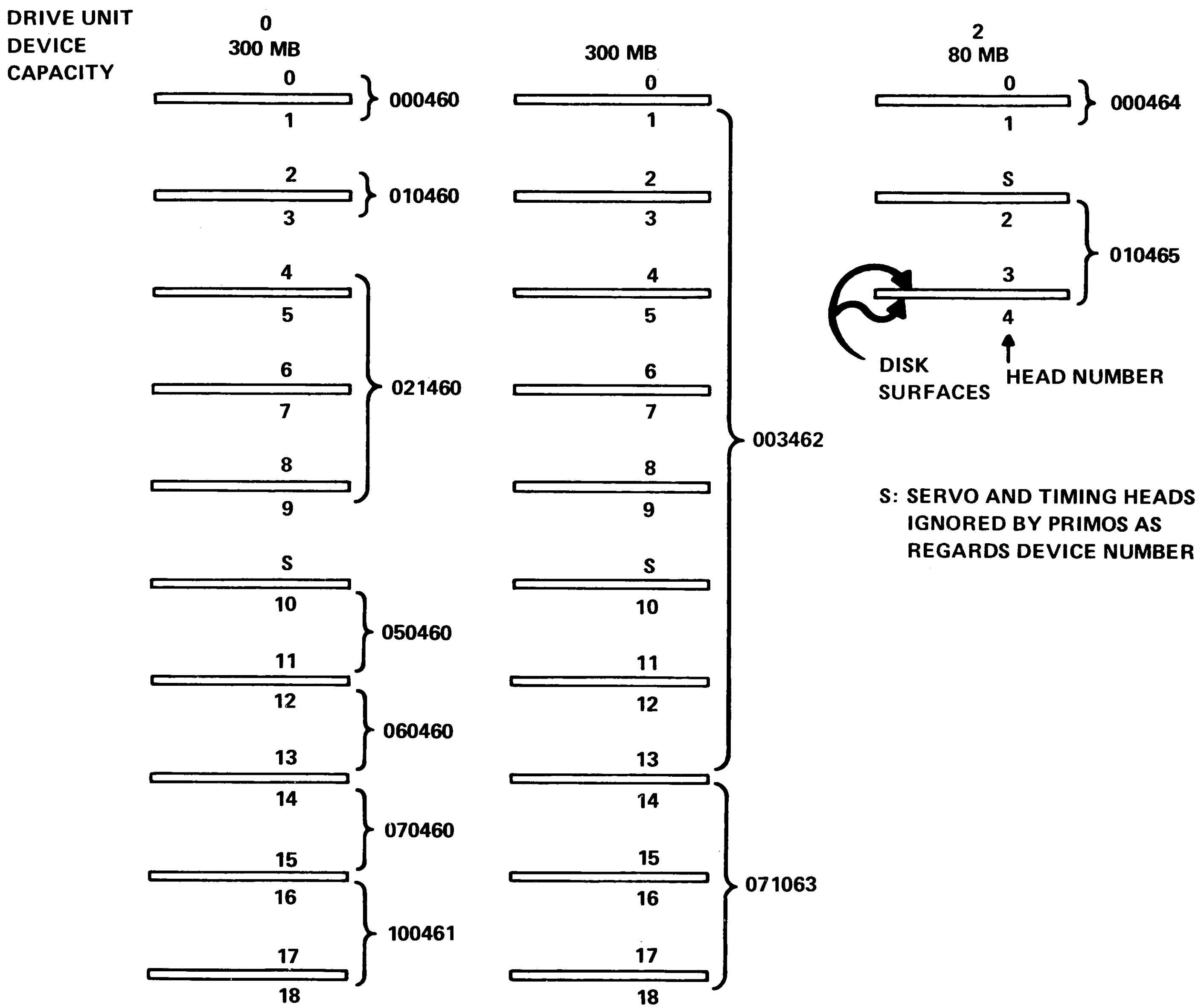


Figure D 2. Example of Storage Module Partitions.

Cartridge Module Devices (CMDs)

Cartridge module devices (CMDs) exist in three sizes - 32 MBytes, 64 MBytes, and 96 MBytes. They may be partitioned according to the table below (Note 1)

CMD Type	Platter(s)	First Controller	Second Controller
32 MB	Removable	6z (16 MB)	26z (16 MB)
	Non-removable	1ØØØ6z (16 MB)	1ØØ26z (16 MB)
64 MB	Removable	6z (16 MB)	26z (16 MB)
	Non-removable	1ØØ46y (32 MB)	1ØØ66y (32 MB)
	(Note 2)	11ØØ6z (16 MB)	11Ø26z (16 MB)
		or	or
		1ØØ46z (48 MB)	1ØØ66z (48 MB)
96 MB	Removable	6z (16 MB)	26z (16 MB)
	Non-removable	1ØØ46y (32 MB)	1ØØ66y (32 MB)
	(Note 3)	11Ø46y (32 MB)	11Ø66y (32 MB)
		12ØØ6z (16 MB)	12Ø26z (16 MB)
		or	or
		1Ø1Ø6y (64 MB)	1Ø126y (64 MB)
		12ØØ6z (16 MB)	12Ø26z (16 MB)
		or	or
		1Ø1Ø6z (8Ø MB)	1Ø126z (8Ø MB)
		or	or
		1ØØ46y (32 MB)	1ØØ66y (32 MB)
		11Ø46z (48 MB)	11Ø66z (48 MB)

Notes

1. y is twice the drive unit number (Ø - 3) on which the disk is mounted.

   z is twice the drive unit number plus one.

2. The non-removable surfaces of the 64MB CMD can be organized as 1 or 2 paritions.

3. The non removable surfaces of the 96MB CMD can be organized as 1 2, or 3 partitions.

APPENDIX E

DISK ERRORS

GENERAL

All record, device, and status numbers are octal.

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(2) act-rec(2) 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(2) — The desired record number within the partition (2 words)

act-rec(2) — The actual record number (in the partition) read (2 words). On reads it should agree with the rec-num(2). On writes it has no validity.

status — indicates reason for error. It may be one of the words in the table below or the sum of two or more.

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

<u>retries</u> the number of times the read or write operation was attempted unsuccessfully before printing the error message. Maximum is 10 ('12).

Under PRIMOS II, the error message format is:

DISK xx ERROR phys-dev rec-num(2) act-rec(2) status

The definitions are the same as for the PRIMOS III, IV, V messages. There is no retry-count; PRIMOS II prints the message and tries ten (12) times.

Error Correction

The storage module controller writes a two-word correcting code checksum on each record. It is generated by the following polynomial:

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 up to 22 bits; (2) a single error burst of up to length 32 bits; 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

<u>wordno</u> The offset relative to the beginning of the record at the beginning of the correction.

<u>error error</u> The 32-bit correction pattern (two 16-bit words) (of which at most, 11 consecutive bits will be non-zero).

Example

```
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 errors messages are printed in the same format as storage modules; the meaning of status-word is different.

DISK RD ERROR device-number record address cra status-word

or

DISK WT ERROR device-number record-address cra status-word

On read request errors cra (2 octal words) is the actual record number read and should match the requested record record-address (2 octal words). It has no validity on a write request.

There is no way, 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, depends on the type of controller as follows:

4000 Controller (obsolete)

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

4001 Controller (cartridge disk)

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

Diskette Controller

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

APPENDIX F

BATCH MESSAGES

▶ <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 (zero) 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 is a warning, but it may have differing degrees of fatality depending on the program being run. In general, it causes the command line that was read to be tossed out. In BATGEN command/subcommand mode, the user will be left in command/subcommand mode. In the case of JOB, BATCH, and entering the BATGEN command, this error results in the user being 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 $$ line as the first non-comment line, but the $$ 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, or copy a new BATDEF from a backup copy using FUTIL.

▶ Can't start batch job!

(Message) The Batch monitor was not spawned from the system console, 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.

▶ Cannot run under DOS.

(Fatal) No part of the Batch system can run under PRIMOS II. The system will halt immediately after "DOS." is printed. If it is necessary to change the BATDEF file while under PRIMOS II, the best solution is to delete it. The Batch monitor will not be runnable until it is re-created, but it will not invalidate the database.

▶ (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 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 <u><treename></u> [-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 <u>greater</u> 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>PH_GO runs *FIXBAT 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 command file C_BDIF should be invoked in BATCHQ to reinitialize the entire database. If it 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.

▶ 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. Re-enter 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). <option> will be replaced by the name of the option The information required by that option should be supplied when the command is re-entered.

▶ 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). Re-enter 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. Re-enter 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 typing of the '$' prompt instead of the '>' prompt given when in BATGEN command mode. To re-enter 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, and the environment had been modified. Legal answers to this question are "YES", "NO", and "OK". If "YES" or "OK" is the response, a subsequent START command will re-enter BATGEN command mode with <u>no</u> loss of information about the environment.

▶ 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 of 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.

▶ 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 re-submit 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 or -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 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 0 (zero), or were not legal decimal numbers and not the string "None". Re-enter 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. Re-enter 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 (zero) 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. Re-enter the command line with legal numeric parameters

▶ Illegal option.

(Fatal) One of the Batch programs was expecting an option, i.e., a unquoted token beginning with a dash ('-'). Re-enter 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). Re-enter 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.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 system console. After determining that the BATCHQ ufd exists, re-spawn the Batch monitor.

▶ Incorrect user-name.

(Fatal) A command file was submitted using the JOB command that had a $$JOB line as the first non-comment line, but the user-name specified after the "JOB" specifier did not match the user-name of the submitting user. Edit the command file and change the username in the $$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 re-initializing 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 release a job that is not held.

Job <extnam> for <username><intnam> <status>.

(Message) 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). Re-enter 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 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 when it is run by the C_BDIF or C_RSET files in the BATCHQ ufd. It means that the queue and execution data has 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 restarted another user's job from a user terminal, or if the process 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 (like 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 not be performed. This can be typed out by the BATCH or the JOB commands when they start running.

▶ 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 2 such jobs belonging to the user making the reference that were active. The internal name must be used in this case. Use JOB -STATUS ALL to determine the internal and external names of all 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). Re-enter the command, but specify each option only once.

▶ Multiple processes 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.

▶ No active jobs [named "<jobname>"]

(Response) 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 ('named "<jobname>") 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>"]

(Response) 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 ALL or -STATUS ALL command, and indicates that there are no jobs belonging to that user (or in the entire batch system if the user is SYSTEM).

The text in brackets ('named "<jobname>") is output if a jobname was specified for the -DISPLAY or -STATUS command, otherwise it is omitted.

▶ No longer executing.

(Fatal) A JOB -ABORT or JOB -RESTART was performed on a job that had execution status, but by the time the execution file was read in to determine the user number of the process, 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 that it was to be submitted to (no -QUEUE option), and no suitable queue could be found. Suitability for a queue includes CPU and elapsed time limits being within the confines of the queue, queue being 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 recent jobs [named "<jobname>"]

(Response) 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 TODAY or -STATUS TODAY command, and indicates that there are no jobs belonging to that user (or in the batch system if the user is SYSTEM) that were submitted, initiated, aborted, completed or canceled today.

The text in brackets ('named "<jobname>"') is output if a jobname was specified for the -DISPLAY or -STATUS command; otherwise it is omitted.

▶ No right. Must be logged in as SYSTEM

(Fatal) Either the BATCH command was used to start or stop the monitor, or a -HOLD or -RELEASE operation was attempted using the JOB command, and the user was not logged in as SYSTEM.

▶ 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 a job and its execution. On new partitions, this time is about 20 seconds maximum; on old partitions, it can take up to a minute and a half.

▶ 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 "*>". Re-submit 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 ALL" to obtain a list of all jobs belonging to the user making the request.

▶ Null home ufd.

(Fatal) The home ufd specified with the -HOME option of the JOB command was a null string. Re-submit the job with an absolute pathname after the -HOME option.

▶ Operator start-up.

(Message) The monitor has seen the request to start via a BATCH SYSTEM -START command, and is now running.

▶ Operator stop.

(Message) The monitor received a stop request via a BATCH SYSTEM -STOP command. The monitor will log out after sending this message. Note that the monitor will respond to a stop request even if it hasn't been issued the BATCH SYSTEM -START request yet.

▶ 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. Re-enter 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 cold-start configuration of the number of available units per user.

▶ Please FILE.

(Warning) A QUIT command was issued while in BATGEN command mode, and the environment had been modified, so 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, and because 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 ("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 20 seconds, the "File in use..." message will be output, and after 30 seconds, the message "Timeout of 30 seconds has occurred" will be output and the program will "give up". 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 he 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.

▶ Process not started.

(Fatal) A BATCH SYSTEM -STOP or BATCH SYSTEM -START command was issued, but the Batch monitor was not running See Section BATOP for a description of how to start the Batch monitor up.

▶ 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, and it will disappear when there are no more waiting, held or executing jobs in that queue...then it can 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 MODIFY'ed 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 by users (i.e. read-permitted for non-owners of the BATCHQ UFD).

▶ 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 re-created 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). Re-enter the command line without the register setting.

▶ Removed <queue-name> from BATDEF

(Message) This message is sent to the system console 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.

▶ 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.

▶ Start-up request issued.

(Response) The BATCH SYSTEM -START command has resulted in the Batch monitor being requested to start up. The monitor should respond with "Operator Start-up" within 10 seconds (or, if it has already been started up, a "Start-up request previously processed." message should be sent within a minute and a half).

▶ Start-up request previously processed.

(Message) The monitor received a start-up request via a BATCH SYSTEM -START command, but it had already been started. This message is just a reminder, not a fatal error.

▶ Stop request issued.

(Response) The BATCH SYSTEM -STOP command has resulted in the Batch monitor being requested to stop. Within a minute and a half, the monitor should send an "Operator Stop" message to the system console 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. Re-enter the command without register settings.

▶ System time must be set first.

(Fatal) A BATCH, BATGEN or JOB command was issued from the supervisor terminal before the system date and time has been set. No part of the Batch system can be run until the system time is set using the SETIME command from the supervisor terminal. (The exception is the Batch monitor, which will wait for the system time and date to be set before it does anything else.)

▶ This job cannot be restarted.

(Response) 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) 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) 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 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 six 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 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.

▶ Waiting for BATCH SYSTEM -START.

(Message) This message is output when the Batch monitor is started up, and after it has determined that the system date and time have been set. It serves as a reminder for the operator to issue the BATCH SYSTEM -START command. It is assumed that the monitor has already been CHAPed to the appropriate levels when this command is issued.

▶ 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

▶ Deleted <filename>.

This message means that FIXBAT found either 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 deleted it.

▶ Execute data found (reinitialize). <jobid>

A job that had execution status in the queue file had no corresponding entry in EXECUT. This can be the result of the Batch monitor being logged out focibly in between updating the queue file and updating EXECUT. Use C_RSET to re-initialize.

▶ 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. Use C_RSET to re-initialize the database, and consider changing the BATCHQ password, as it may indicate a security breakage.

▶ 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 a process submitting a job being force-logged out in the middle of creating the new queue entry.

▶ Redundant execute entry (reinitialize). <jobid>

FIXBAT found an executing job that had more than one entry in the EXECUT file, which is impossible. Use C_RSET to re-initialize the database.

▶ Unable to process batch jobs.

Occurs if an attempt is made to run *FIXBAT with -STARTUP specified from the system console, or a user logged in as SYSTEM that is not a phantom that was spawned from the system console.

▶ Unknown -STARTUP argument.

The argument supplied to the -STARTUP option is not SAVE, DELETE, SPOOL or NOLOG.

APPENDIX G

THE EVENT LOGGER

OPERATION OF THE EVENT LOGGING SYSTEM

First Level Event Logger (LOGEV1)

Information about an event is entered into the event buffer, LOGBUF by LOGEV1, an internal PRIMOS subroutine. Each entry in the buffer contains the type and length of the entry and a number of data words passed to LOGEV1 by the routine programmed to record the event. (The exact format of event entries is described below.) When LOGBUF fills up, LOGEV1 discards subsequent entries and increments LOGOVF, a counter of the number of events lost.

LOGEV1 is called from the check handlers in SEG4, DISK10, DOSSUB and TRWRAT.

Second Level Event Logger (LOGEV2)

The internal subroutine, LOGEV2, examines LOGBUF once a minute. If LOGBUF is non-empty LOGEV2 dumps it to a disk file named LOGREC in the current UFD of the user known to PRIMOS as User 1, normally CMDNC0 on the command device. LOGEV2 does not dump LOGREC until the time has been set by the system operator. LOGEV2 is called from two places in PRIMOS: PABORT, when the one-minute process abort occurs, and DOSSUB, when a SHUTDN ALL command is issued. LOGEV2 cannot be called by the user, but the user can monitor the output of LOGEV2 via the LOGPRT command.

LOGEV2 does not dump LOGBUF if the file LOGREC does not exist in CMDNC0, or if the configuration directive LOGREC has been used to set the LOGREC quota to a negative value. This allows operation with a write-protected disk.

Note

> If the command device (disk) is write-protected and a LOGREC file exists in CMDNCO, and a LOGREC 177777 directive has not been used, a disk WRITE-PROTECT error message is printed on the supervisor terminal once each minute.

The LOGREC file can be created with any sequence of PRIMOS commands equivalent to the following:

```
0 'CMDNC0 password>LOGREC' 2 2003
C 2
```

The size of LOGREC is controlled by a parameter in LOGEV2 and is currently '10000 (4096 decimal) words. If LOGREC exceeds this size, or the size specified in the LOGREC directive of the CONFIG command, LOGEV2 prints the message: EXCEEDING QUOTA ON LOGREC at the supervisor terminal. It will, however, continue to log information to LOGREC and print the message once each minute until LOGREC is purged by the LOGPRT routine. (Alternatively, of course, LOGREC can be deleted.) The size of LOGREC can be changed by the LOGREC directive in the CONFIG data file. The quota checking can be disabled or writing to LOGREC can be suppressed.

Before dumping LOGREC, LOGEV2 writes an entry to LOGREC noting the current time and date. After LOGREC is dumped, if LOGOVF (the overflow counter) is non-zero, LOGEV2 writes an entry noting the number of LOGBUF overflows.

Note

> To ensure that LOGEV2 gets a chance to dump the contents of LOGBUF, a warm start should be performed whenever possible after a machine check halt. Wait one minute or issue a SHUTDN ALL.

MODIFYING THE EVENT LOGGING MECHANISM

The following paragraphs describe how to make modifications to the event logging mechanism. The relevant modules are found as follows: LOGEV1 and LOGBUF are in PRI400>KS>SEG4.PMA. LOGEV2 is PRI400>KS>LOGEV2.FTN. LOGPRT is supplied in SYSTEM>LOGCRT

Increasing The Size Of LOGBUF

LOGBUF is defined in SEG4.PMA (PRIMOS). The first entry in the buffer (label LOGBUF) is a COLD START entry followed by BSZ which defines the remaining size of LOGBUF (at least 63). It can be redefined as desired.

Adding Event Types

To log a new event type, three actions are necessary:

1. An event message must be built that contains the event type, length of the message, and (optional) data words.

Event Message Format: An event message consists of a header word followed by up to 23 optional data words The header word consists of the event type in bits 1 8 and the total message length in bits 9-16. In PMA, a message could be defined by:

```
MSG   DATA   (5.LS.8)+3,DATA1,DATA2
```

Currently Defined Event Types: Currently, the following event types are defined.

0	COLD START
1	WARM START
2	DATE/TIME STAMP (LOGEV2)
3	CHECKS (MACHINE, MEMORY PARITY, MISSING MEMORY)
4	DISK ERRORS
5	LOGBUF OVERFLOW (LOGEV2)
6	SHUTDN ALL
7	PRIME 300 MACHINE CHECK
8	PRIME 300 MEMORY PARITY
9	PRIME 300 MISSING MEMORY
10-15	Reserved for user defined types
16	DISK MOUNT
17	POWER FAIL CHECK
18	Reserved
19	Reserved
20	REMARK

2. LOGEV1 must be called to enter the message into LOGBUF.

Calling LOGEV1 -- PRIMOS

In PMA, code inside SEG4

```
JSXB LOGEVL (NOTE DIFFERENT NAME)     IP MESSAGE
```

IN PMA, code outside SEG4.

```
CALL LOGEV1      AP MESSAGE,SL
```

In FORTRAN:

```
CALL LOGEV1 (MESSAGE)
```

3 LOGPRT must be modified to recognize the new event type and appropriately format the data associated with the event. LOGEV1 and LOGEV2 do not examine the type field.

Modifying LOGPRT: Currently, LOGPRT recognizes and formats data for event types 0-9, and 16-20. Types 10-15 are accepted, but result in a printout of only:

```
TYPE=type DATA=word-1 word 2 ... word-7
```

Only nine data words are allowed for these types. To add a new type, add a label to the computed GOTO following statement $400. At the new label (between $1950 and $2000) call the STORE routine to perform the required formatting.

The calling sequence for STORE is as follows:

```
CALL STORE (TEXT,TXTLEN,ARRAY,NW,DEC)
```

The meaning of the parameters is as follows:

TEXT	A text string to be printed.
TXTLEN	The length in characters in TEXT. If zero, no text is printed.
ARRAY	An array of words to be translated and entered in the output line. ENTRY(1) is the first data word of the event message. ENTTYP and ENTLEN contain the type and length of the entry.
NW	The number of words in ARRAY. If zero, no words are translated.
DEC	Octal/decimal flag. If zero, translation is to octal with no leading zero suppression. If non-zero, translation is to decimal with leading zeroes suppressed.

The total length of the text to be stored (=TXTLEN+NW*7) should not exceed 67, the maximum length that can be printed at a terminal with an indent in effect. (All lines after the first for an entry are indented 5 spaces.) If the length of TEXT is too long, it will be split at a line boundary subsequent lines being indented.

After formatting the entry, GOTO 99000. Code at that label finishes the formatting and obtains the next entry from LOGREC.

To rebuild LOGPRT, run the command File C_LOGPRT in SYSTEM (or C_LLOGPRT, if a full listing is desired) This creates a run file called *LOGPRT. *LOGPRT can then be moved to the UFD CMDNC0.

Changing The Size Of LOGREC

The size of LOGREC (over which the EXCEEDING... message is printed) is defined by the FIGCOM variable LRQUOT in SEG14 The value of LRQUOT is set by the configuration directive LOGREC.

Changing LOGPRT's Default Input/Output Filenames

The default input name, <0>CMDNC0>LOGREC, is in the array LOGINP The size of LOGINP and length of the name, INNAML should be set to the number of words and characters in LOGINP respectively. The default output name (LOGLST) is in the array LOGOUT, which is always 16 words long, blank padded.

LOGPRT Messages

▶ COLD START [PRIMOS REV RR .CPU TYPE= t MICROCODE REV = mm
ID= iiiiii]

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 PMA Programmer's Guide FDR3059) CPU TYPE is the Central Processing Unit:

Type (t)	Model Number
0	P400
1	Reserved
2	Reserved
3	P350
4	P450, P550
5	P750
6	P500, P650
7	Reserved

<u>mm</u> The revision of the microcode running.

<u>iiiii</u> The full eight-word identification code from the STMP instruction.

<u>rr</u> ASCII version identification string, if present.

▶ WARM START

Indicates a warm start of PRIMOS was performed.

▶ MACHINE CHECK (xxx) DSWSTAT= ssssss ssssss DSWRMA= yyyyy rrrrrr rrrrrr DSWPB= pppppp pppppp [DSWPARITY xxxxxx xxxxxx ..]

A machine check occurred. DSWSTAT, DSWRMA, DSWPB, and DSWPARITY constitute the DSW (descriptor segment word) at the time of the check.

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. If DSPARITY is not present, xxx is an encoding of the machine check code and Not RCM Parity in DSWSTATH 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 the RMA INVALID bit is set (bit 9 of DSWSTATL), yyyyy is (INV), otherwise yyyyy is absent.

If DSWPARITY is present, it is broken down by reporting board (A, C, CS, D) 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 Ø.)

DSWPARITYH

Ø1 - RPARERR1+	CS	DMX input E6: BPD or Burst RØ,R2	
		E5: BPD or Burst - RØ R1 R2 R3	
		DMX output : BMD	
Ø2 - RPARERR2+	CS	DMX input E6: BPD or Burst- R1,R3	
		E5: BPD	
		DMX output : BMA	
Ø3 - FBDMX+	CS	Burst mode DMX transfer	
Ø4 - BURST-INPUT+	CS	1=DMX input, Ø=DMX output	

Ø5,Ø6,Ø7 - Ø	- 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

Ø8 - RCMPE-	A	RCM parity if no board reported error
Ø9 - FMDECCU+	D	Memory reports ECC uncorrectable read error
1Ø - 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
16 -			

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
15 -			
16 -			

▶ MISSING MEMORY DSWSTAT= ..

A missing memory module check occured. Information is the same as for a machine check, except that the machine check code xxx does not appear.

▶ MEMORY PARITY (xxxx) DSWSTAT= . . PPN,WN= pppppp wwwwww

A memory parity error occurred. xxxx is either ECCC (corrected) or ECCU (uncorrected). The DSW is followed by PPN,WN=pppppp wwwwww, which identifies the physical page and word number of the error. For an ECCC error the PPN is followed by Bit = xx. (xx = 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=x, x=Ø or 1, setting of DSWSTATL bit 6 (overall parity).

DSWPARITY is displayed but not decoded.

▶ POWER FAIL CHECK

A power fail check occurred.

▶ DISK xx ERROR DVNO= dddddd (typecode) CRA= rrrrrr rrrrrr CYL= ccc HEAD= hh RECORD= rr RCRA= aaaaaa aaaaaa STATUS(ACCUM)= ssssss STATUS (LAST) = llllll RETRIES= tt mmmmmm

A disk error occurred during an xx operation, where xx is RD for read or WT for write. DVNO is followed by dddddd, the device number. typecode gives the controller number and device type (MHD for moving head disk SM for storage module). CRA rrrrrr rrrrrr gives the record address, which is broken up into CYL (cylinder), HEAD and RECORD address (all decimal). For a read operation, RCRA gives the CRA read on a CRA error STATUS (ACCUM) is the OR of all status bits obtained during retries. STATUS (LAST) is the status of the last operation.

RETRIES gives the number of retries attempted. If RETRIES is less than 1Ø, the operation was completed successfully, the value of string mmmmmm will be (RECOVERED).

If RETRIES = 1Ø, mmmmmm will be (UNCORRECTABLE) if the error could not be corrected. If an ECC error has been successfully corrected by the software, mmmmmm is WORDNO= and CORRECTION=, which gives the word number in the record and the 32-bit correction pattern used.

▶ DISK MOUNT: volume-name ON dvno

An ADDISK or STARTUP command was issued. The indicated volume-name was mounted on the disk identified as dvno (which is a physical disk or partition number).

▶ 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 gives the user's PC at the time of the check.

▶ MEMORY PARITY

A Prime 300 memory parity error occurred (see next entry).

▶ MEMORY PARITY PPN= pppppp WN= wwwwww CONTENTS= cccccc

The format of an entry for a Prime 300 memory parity error encountered during a WARM start memory scan. The physical page number (PPN), word number offset in the page (WN), and the incorrect contents are given.

▶ MISSING MEMORY

A Prime 300 missing memory check entry.

▶ LOGBUF OVERFLOW -- nnnnn ENTRIES LOST

Indicates nnnnn (decimal) event entries were lost due to overflow of the event logging buffet (LOGBUF).

▶ SHUTDOWN BY OPERATOR

The operator issued a SHUTDN ALL command. (This automatically dumps LOGBUF.)

▶ TYPE=tt DATA= dddddd ...

A LOGREC 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

▶ "Text of Operator remark"

Contents of the REMARK EVENT (type 20).

▶ BAD ENTRY xxxxxx

An entry of unrecognized type or length longer than 81 words was encountered. The length is printed.

▶ *** LOGREC EMPTY ***

This message is printed is LOGPRT finds no entries in LOGREC.

▶ *** END OF FILE -- nnnnn ENTRIES, ppppp PROCESSED ***

This message is printed when LOGPRT reaches the end of LOGREC. nnnnn (decimal) gives the number of entries processed <u>not</u> including date/time and LOGBUF overflow entries, ppppp is the number of entries processed.

When all the entries in LOGREC (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), LOGPRT positions to the beginning of the input file before closing, in effect emptying the file.

Finally, if the -SPOOL option is in effect, 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 then deleted.

APPENDIX H

THE NETWORK EVENT LOGGER

OPERATION OF THE NETWORK EVENT LOGGING SYSTEM

First Level Event Logger (NETEV1)

Information about the network event is entered into an event buffer, NETBUF, by NETEV1, an internal PRIMOS subroutine. Each entry in the buffer contains the type and length of the entry and a number of data words passed to NETEV1 by the routine wishing to record the event. (The exact format of event entries is described below.) When NETBUF fills up, NETEV1 discards subsequent entries and increments NETOVF, a counter of the number of events lost

NETEV1 is called from the communications device interface modules (dims), DOSSUB and the X.25 network software.

Second Level Event Logger (NETEV2)

The internal subroutine, NETEV2, examines NETBUF every 30 seconds. If NETBUF is non-empty, NETEV2 writes it to a disk file named NETREC in the current UFD of user 1, normally CMDNC0 on the command device. NETEV2 will not dump NETREC until the time has been set by the system operator. NETEV2 is called from two places in the PRIMOS supervisor: PABORT when the 30 second process abort occurs, and DOSSUB when a SHUTDN ALL command is issued.

NETEV2 does not dump NETBUF if the file NETREC does not exist in CMDNC0. The NETREC file can be created with any sequence of PRIMOS commands equivalent to the following:

```
0 'CMDNC0 password>NETREC' 2 2003
C 2
```

Note

If possible, a warm start should be performed after a machine check halt. This allows NETEV2 to dump NETBUF, either after 30 seconds or on a SHUTDN ALL command.

MODIFYING THE EVENT LOGGING MECHANISM

The following paragraphs describe how to modify the event logging mechanism. The relevant modules are found as follows: NETEV1 and NETEV2 are found in PRI400>NS. NETBUF is found in PRI400>NS>COMDEF. NETPRT is in SYSTEM.

Increasing the Size of NETBUF

NETBUF is defined in COMDEF. The first entry in the buffer (label NETBUF) is a 1 word COLD START entry followed by a BSZ which defines the remaining size of NETBUF (currently 63). It can be redefined as desired.

Adding Event Types

To log a new event type, three actions are necessary:

1. An event message must be built that contains the event type, length of the message, and (optional) data words.

   Event Message Format: An event message consists of a header word followed by up to 23 optional data words. The header word consists of the event type in bits 1 8 and the total message length in bits 9-16. In PMA, a message could be defined by:

   ```
   MSG    DATA    (5.LS.8)+3,DATA1,DATA2
   ```

   This defines a message for event type 5, length of message (including header word) is 3 words.

   Currently Defined Event Types: Currently, the following event types are defined.

0	COLD START
1	SHUTDOWN
2	DATE/TIME STAMP (NETEV2)
3	VIRTUAL CIRCUIT RESETS
4	PACKETS OUT OF SEQUENCE
5	NETBUF OVERFLOW (NETEV2)
6	X.25 LEVEL III PROTOCOL DOWN IN A HOST
7	CIRCUIT CLEAR CAUSED BY LOCAL PROCEDURAL ERROR
8	RING1 HARDWARE ERROR
9	RING2 HARDWARE ERROR
10	RING3 HARDWARE ERROR
11	NETWORK SOFTWARE ERROR SIGNALLED BY CALL TO NETDMP
12	SMLC 1 STATUS ERROR
13	SMLC 2 STX/ETX
14	SMLC3 PROTOCOL TEMPORARILY OUT OF MEMORY FOR MESSAGES
15	SMLC4 LINE RESET
16	HOST DOWN
17	POWER FAIL CHECK
18	Reserved
19	Reserved
20	REMARK

2. NETEV1 must be called to enter the message into NETBUF.

   Calling NETEV1

   In PMA:

   ```
   CALL NETEV1
   AP   MESSAGE,SL
   ```

   In FORTRAN:

   ```
   CALL NETEV1(MESSAGE)
   ```

3. NETPRT must be modified to recognize the new event type and appropriately format the data associated with the event. (NETEV1 and NETEV2 do not examine the type field.)

   Modifying NETPRT: Currently, NETPRT recognizes and formats data for event types 0-20. To add a new type, add a label to the computed GOTO at statement $600. At the new label (between $2500 and $2900), call the STORE routine to perform the required formatting.

   The calling sequence for STORE is:

   ```
   CALL STORE (TEXT,TXTLEN,ARRAY,NW,DEC)
   ```

   TEXT — A text string to be printed.

   TXTLEN — The length in characters in TEXT If zero, no text is printed.

   ARRAY — An array of words to be translated and entered in the output line. ENTRY(1) is the first data word of the event message. ENTTYP and ENTLEN contain the type and length of the entry.

   NW — The number of words in ARRAY. If zero, no words are translated.

   DEC — Octal/decimal flag. If zero, translation is to octal with no leading zero suppression. If non-zero, translation is to decimal with leading zeroes suppressed.

   The total length of the text to be stored (=TXTLEN+NW*7) should not exceed 67, the maximum length that can be printed on a TTY with an indent in effect (All lines after the first for an entry are indented 5 spaces.) If the length of text is too long, it will be split at a line boundary subsequent lines being indented.

After formatting the entry, GOTO 985ØØ. Code at that label finishes the formatting and obtains the next entry from NETREC.

To rebuild NETPRT, run the command file C_NETPRT in SYSTEM (or C_NETPRT if a full listing is desired). This will create a run file called *NETPRT. *NETPRT can then be moved to CMDNCØ and renamed to NETPRT.

Changing NETPRT's Default Input/Output Filenames

The default input name, <Ø>CMDNCO>NETREC, is in the array INPNAM. The size of INPNAM and length of the name INNAML, should be set to the number of words and characters in INPNAM respectively. The default output name (NETLST) is in the array OUTNAM.

NETPRT Messages

COLD START

A cold start of PRIMOS was performed.

OVER

CIRCUIT RESET REMOTELY/LOCALLY ORIGINATED contoller: physical-line -or-node

A virtual circuit was reset. If bit 1 of FLAG is Ø, the reset was incoming. If bit 1 of FLAG is 1, the reset was outgoing. All other bits are undefined.

PACKET OUT OF SEQUENCE CONTROLLER: physical line number or node

A packet was received with an unexpected sequence number.

LEVEL III PROTOCOL DOWN controller: physical-line-or-node

The Level III protocol for X.25 is down for this host.

LOCAL PROCEDURAL LEVEL CAUSING CLEAR controller: physical-line-or-node

A local procedural error caused a circuit clear in this host.

TOKEN INSERTED INTO THE RING NETWORK

The software controlling the PNC hardware issued a ring network control token.

▶ RING DIM OUT OF RECEIVE BLOCKS (FARDIM)

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 networks 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.

▶ NETDMP CALLED AT: xxxxxx xxxxxx

A network software problem has occurred at this address.

▶ SMLC STATUS ERROR STATUS 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

▶ SMLC - NO STX PRECEEDING ETX PHYSICAL LINE NUMBER IS xxxxxx DEVICE ADDRESS IS xxxxxx

SMLC packets must begin with DLE/STX and end with DLE/ETX.

▶ SYSTEM BLOCKS UNAVAILABLE FOR SMLC PROTOCOL MESSAGE IS xxxxxx LOGICAL LINE IS xxxxxx

The level II synchronous protocol had no buffers in which to send this type of protocol-generated message.

▶ SMLC RESET FOR LOGICAL LINE xxxxxx - cause

Resets can be caused in three ways. The level II synchronous protocol timed out and was forced to 'reset' the line, normal message sequence numbers were incorrect, or this line just received a COMMAND REJECT from the remote system.

▶ *** NETREC EMPTY ***

This message is printed if NETPRT finds no entries in NETREC.

▶ *** END OF FILE - nnnnn ENTRIES, ppppp PROCESSED ***

This message is printed when NETPRT reaches the end of NETREC. nnnnn (decimal) gives the number of entries in NETREC not including date/time and NETBUF overflow entries. ppppp gives the number of entries processed.

When all the entries in NETREC (or other input files) have been processed NETPRT will normally close the file and exit. If, however, the -PURGE option has been specified NETPRT will position to the beginning of the file before closing, in effect emptying the file.

If the SPOOL option is in effect, NETPRT 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 then deleted.

APPENDIX I

SEQUENTIAL JOB EXECUTION (CX)

INTRODUCTION

CX is a utility that allows users to queue jobs to execute sequentially on a first in first out basis.

At Rev. 17, CX has been replaced functionally by the Batch execution environment. The C_PRMO template delivered with the system includes the command to bring up the Batch monitor. It does not include the command to bring up CX. (For security reasons, it is inadvisable to have Batch and CX running simultaneously.)

All existing CX jobs can be run as Batch jobs, using the JOB filename command. Therefore, CX does not have to be employed to use existing CX files. Batch offers considerably more functionality than CX does (see Section 13). For these reasons, we strongly suggest that you implement Batch and not CX. If you do want to implement CX, however, this appendix will tell you how to do it.

OPERATION

CX consists of a phantom monitor program which monitors the progress of another phantom, the CX slave. Normally, both phantoms are running under the login name SYSTEM. Which is the monitor and which the slave can be determined by examining the supervisor terminal output printed when the system was brought up. The CX monitor sends a message to the supervisor terminal announcing its initialization.

The CX monitor is started by the command:

```
PH CX***>PH_GO
```

This command can be placed in the command file C_PRMO so that CX will be brought up automatically each time the system is brought up. When it receives the command, PRIMOS responds:

```
PHANTOM IS USER user-number
```

When the phantom starts up, the message:

```
SYSTEM (user-number) LOGGED IN AT time
```

is printed at the supervisor terminal When the monitor has initialized itself (approximately one minute later), it sends the message:

```
*CX* CX monitor, rev 17.1
```

THE CX MONITOR

Only in rare cases will the CX monitor need operator intervention. The CX monitor wil log itself out if it encounters an unrecovered disk error (Appendix E). To restart properly:

1. Log in to UFD SYSTEM with the owner password at any terminal.

2. Force logout the CX slave phantom.

3. Type the command PH CX***>PH_GO.

If another user starts up a phantom between the time the monitor is started and the time it starts up its slave(s), it may cause the CX monitor to send the message MINIMUM PHANTOMS NOT AVAILABLE to the supervisor terminal This may also prevent the restart of the CX monitor. In this case, the operator must force-logout a phantom user to free enough phantoms for CX operation.

At the user s request or for other reasons (such as a phantom program looping infinitely) the operator may cancel a currently running job in the CX queue. To do this, login to UFD SYSTEM and force logout the CX slave (not the CX monitor).

MONITORING THE CX QUEUE

The contents of the local CX job file are printed at the terminal by:

CX -A

Example

OK, CX A

CX job file listing 16 Mar-79 3:38 pm

File	Id	Owner	State	Date/Time	Cpu limit	Prio
CX##10		HOUDE	Completed	16-Mar-79 12:51 pm	none	3
CX##09		CAD	Completed	14-Mar-79 1:53 pm	none	3
CX##08		CAD	Completed	14-Mar-79 11 36 am	none	3
CX##07	DELETE	STURAE	Completed	13 Mar-79 4:15 pm	none	3
CX##06		CAD	Completed	12-Mar-79 10:26 pm	none	3
CX##05		MEG	Completed	12-Mar-79 5:38 pm	none	3
CX##04		HOUDE	Completed	09-Mar-79 5:33 pm	none	3
CX##03		EMBERS	Completed	08-Mar-79 7:01 pm	none	3
CX##02		HOUDE	Completed	08-Mar-79 6:19 pm	none	3
CX##01		JGENO	Aborted	08-Mar-79 5:13 pm	none	3

[CX rev 17.1]

- The command CX -P prints only CX jobs requested by the user (OWNER column)
- The command CX Q prints only those jobs executing or awaiting execution Completed, aborted, and dropped jobs are not printed.
- Information about a specific job is obtained by

      CX -Sjob#

Example

```
OK, CX S9

Owner=CAD      State=Completed
Cpu limit=none  Prio= 3  14 Mar-79  1:53 pm

[CX, rev 17.1]
```

- Whenever the status of an executing job is printed, its user number is also printed immediately following the job's priority.

SUPERVISOR TERMINAL MESSAGES

CX sends the following messages to the supervisor terminal:

▶ *** cx-phan-no hh'mm
CX Executing CX##job for user ufd-name (cx-phan-user-no)

The CX processor, running as user number cx phan-no has started execution of a job at hh'mm. The job, requested by ufd-name, is number job in the CX queue. It is running as user number cx-phan-user-no:

```
*** 57 13'05
*CX*    Executing CX##05 for user TEKMAN (60)
```

▶ *** CX-phan-no hh'mm
CX Job CX##job user ufd-name (cx-phan-user-no) aborted

The CX job number job requested by ufd-name, running as user cx-phan-user-no has been aborted at time hh'mm. The CX processor is running as user number cx-phan-no:

```
*** 57 17'58
*CX*    Job CX##31 user TEKMAN (60) aborted
```

```
*** CX-phan-no hh'mm
*CX*     Job CX##job user ufd-name (cx phan-user-no) completed
```

The CX job number job requested by ufd-name, running as user cx-phan-user no has been completed at time hh'mm. The CX processor is running as user number cx-phan-no:

```
*** 57 17'58
*CX*    Job CX##31 user TEKMAN (60) completed
```