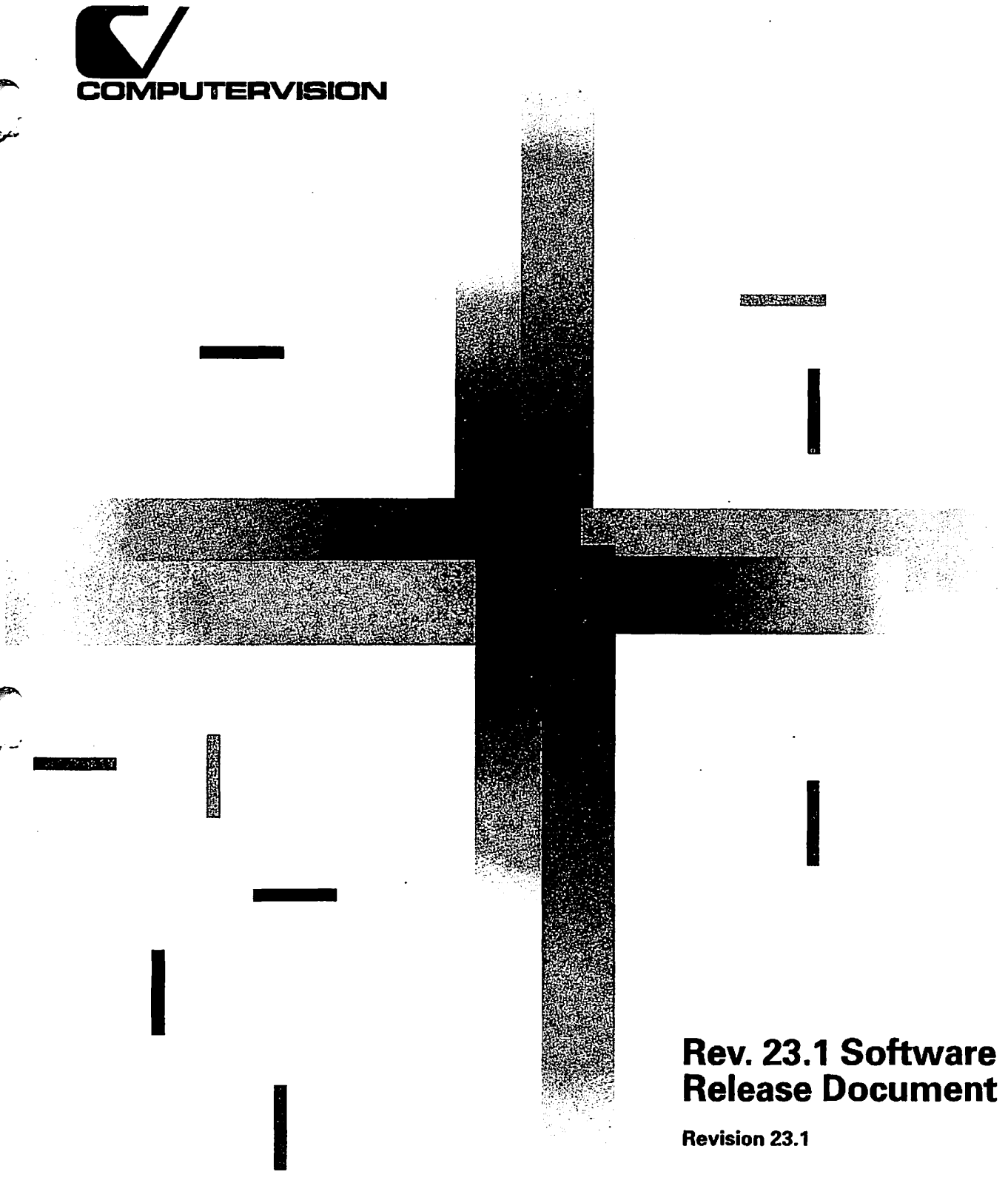


COMPUTERVISION

Rev. 23.1 Software Release Document

Revision 23.1

DOC10001-8PA

Rev. 23.1 Software Release Document

Glenn Morrow

This manual documents the software operation of the PRIMOS® operating system on 50 Series™ computers and their supporting systems and utilities as implemented at Master Disk Revision Level 23.1 (Rev. 23.1).

Prime Computer, Inc., Prime Park, Natick, Massachusetts 01760

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

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

Copyright © 1991 by Prime Computer, Inc. All rights reserved.

PRIME, PR1ME, PRIMOS, and the Prime logo are registered trademarks of Prime Computer, Inc. 50 Series, 400, 750, 850, 2250, 2350, 2450, 2455, 2550, 2655, 2755, 2850, 2950, 4050, 4150, 4450, 5310, 5320, 5330, 5340, 6150, 6350, 6450, 6550, 6650, 9650, 9655, 9750, 9755, 9950, 9955, 9955II, DISCOVER, PRIME EXLNET, Prime INFORMATION CONNECTION, PRIME/SNA, PRIME EXL, PRIME EXL MBX, INFO/BASIC, MIDAS, MIDASPLUS, PERFORM, PERFORMER, PRIFORMA, PRIMAN, Prime INFORMATION, INFORM, PRISAM, Prime INFORMATION PLUS, PRIMELINK, PRIMIX, PRIMENET, PRIMEWAY, PRODUCER, PRIMEWORD, Prime INFORMATION EXL, Prime INFORMATION/pc, PRIME TIMER, RINGNET, SIMPLE, PT25, PT45, PT65, PT200, PT250, and PST 100 are trademarks of Prime Computer, Inc. PrimeService is a service mark of Prime Computer, Inc.

Ethernet is a trademark of Xerox Corp.

IBM is a registered trademark of International Business Machines Corp.

PostScript is a registered trademark of Adobe Systems, Inc.

UNIX is a registered trademark of UNIX System Laboratories, Inc..

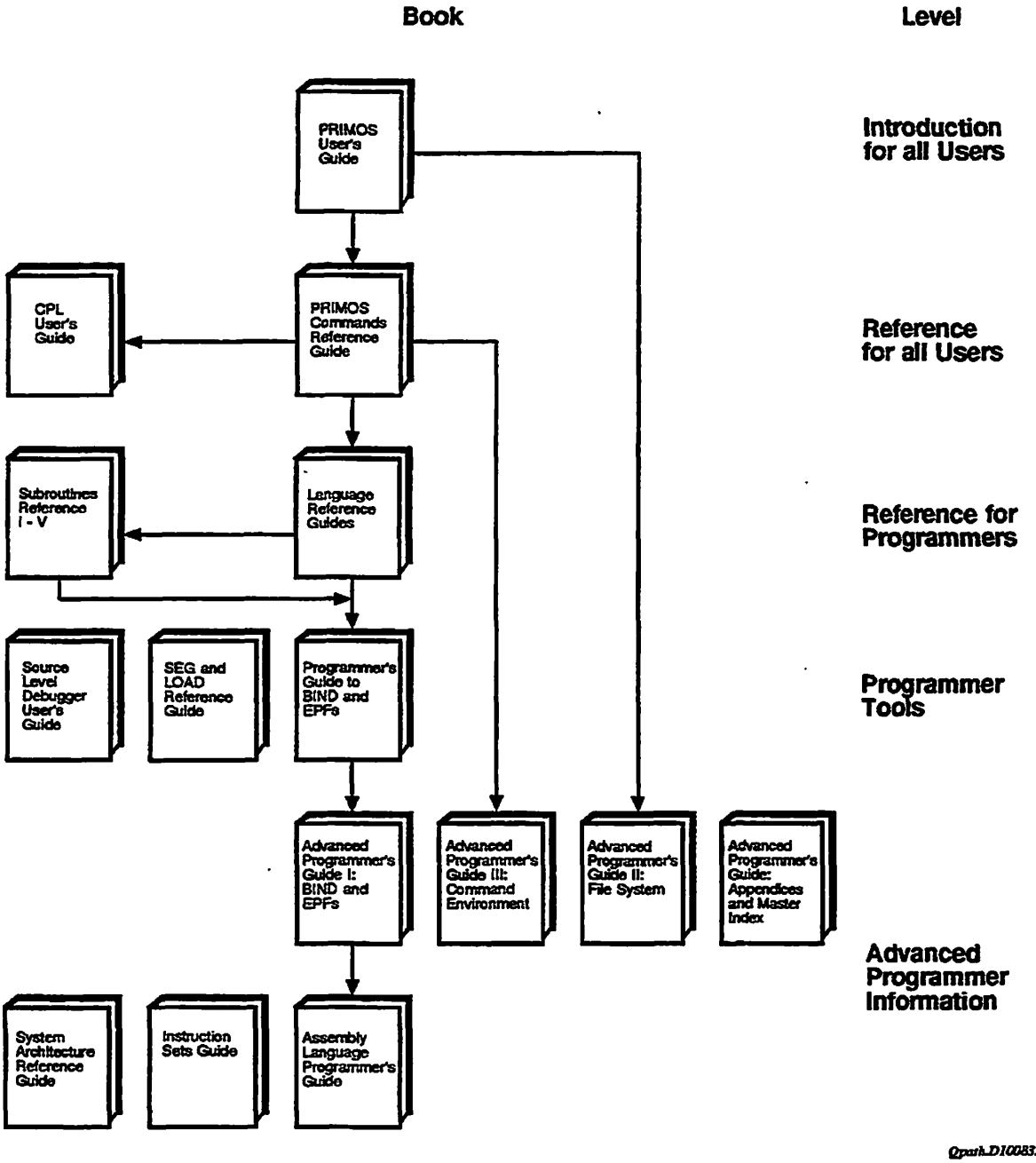
Printing History

First Edition (DOC10001–1PA) April 1985 for Revision 19.4
Second Edition (DOC10001–2PA) January 1986 for Revision 20.0
Third Edition (DOC10001–3PA) August 1986 for Revision 20.2
Fourth Edition (DOC10001–4PA) July 1987 for Revision 21.0
Fifth Edition (DOC10001–5PA) October 1988 for Revision 22.0
Sixth Edition (DOC10001–6PA) July 1989 for Revision 22.1
Seventh Edition (DOC10001–7PA) June 1990 for Revision 23.0
Eighth Edition (DOC10001–8PA) March 1991 for Revision 23.1

Credits

Editorial: Mary Skousgaard
Project Support: Jeannie Nakano, Scott Gorton, Bill Nicolson, Greg Dawe
Production: Judy Gordon

Reading Path for PRIMOS Documentation



Contents

About This Book

1 Introduction

Special Considerations for Rev. 23.1 . . . 1-1
- Installation . . . 1-1
- Microcode Requirements . . . 1-2

Product Retirements . . . 1-3

Rev. 23.1 New Features . . . 1-3
- User and Programmer Features . . . 1-3
- Operator and Administrator Features . . . 1-4

2 New Features for the User and Programmer at Rev 23.1

CNAME Command Enhancement . . . 2-1

CREATE Command Enhancement . . . 2-1

LD Command Enhancements . . . 2-2
- LD –RWLOCK . . . 2-2
- LD –TOTAL . . . 2-4

Configurable RDY Prompts . . . 2-4

TALK Enhancements . . . 2-8

SHOW and WATCH Commands . . . 2-8
- SHOW Command . . . 2-9
- WATCH Command . . . 2-10
- WATCH Session Considerations . . . 2-12

New Command Functions . . . 2-12
- User and System Information Functions . . . 2-13
- String Manipulation Functions . . . 2-17

Multiple Attach Points . . . 2-18
- Usage . . . 2-18
- Parameters . . . 2-19

ACL-protected Shared Memory . . . 2-20
- Usage . . . 2-21
- Parameters . . . 2-21
- Discussion . . . 2-22

Documentation Corrections . . . 2-23
C Compiler . . . 2-23
CBL Compiler . . . 2-23
CPL Functions . . . 2-23
CF$EXT Subroutine . . . 2-24
EPF$MAP Subroutine . . . 2-24
EPF$REG Subroutine . . . 2-24
SRTF$S, SETU$S, and MRG1$S Subroutines . . . 2-25
T$MT Subroutine . . . 2-25
RDLIN$ and WTLIN$ Subroutines . . . 2-25
PRIMOS Error Codes . . . 2-26

3 New Features for the Operator and Administrator at Rev. 23.1

Configurable Login Prompt . . . 3-1
START_LSR Command Enhancements . . . 3-1
–PROMPT Examples . . . 3-2
–REDISPLAY_PROMPT Examples . . . 3-2
Setting Up LOGIN_SERVER Defaults . . . 3-3
Sample Session 1 . . . 3-4
Sample Session 2 . . . 3-5
BATCH –START Command . . . 3-6
Batch INIT Command . . . 3-6
MAKE Command . . . 3-6
SPOOLER Enhancements . . . 3-8
MPC4000 Command . . . 3-8
ASYNC –MAXIMUM_SHUTDOWN_TIME . . . 3-10
HEADER –NO_INITIAL_FF . . . 3-10
Data File Unit Number . . . 3-10
Double-sided Laser Printing . . . 3-10
RECORD_TO_PATH Command . . . 3-10
USAGE Command . . . 3-12
SET_ASYNC Command . . . 3-13
MAGSAV/RST Enhancements . . . 3-13
INDEX_LIB_MANAGER Command . . . 3-13
SEARCH_INDEX_LIB Command . . . 3-15
NSEG System Configuration Directive Enhancement . . . 3-16
MTRS System Configuration Directive Enhancement . . . 3-17
EDIT_PROFILE Enhancement . . . 3-17
Documentation Corrections . . . 3-18
Network Terminal Service (NTS) . . . 3-18
DSM User's Guide . . . 3-18
Operator's Guide to the Spooler Subsystem . . . 3-21

4 Crash Recovery Facilities

Resident Forced Shutdown (RFS) . . . 4-2
- Running RFS . . . 4-2
- RFS Console Messages . . . 4-3

What Is FS_RECOVER? . . . 4-4
- Effects of a System Crash on Your File System . . . 4-4
- What Does FS_RECOVER Do? . . . 4-5
- FS_RECOVER Using a Crash Dump Tape . . . 4-5
- FS_RECOVER Without a Crash Dump Tape . . . 4-7
- Considerations When Using FS_RECOVER . . . 4-7

Installing FS_RECOVER . . . 4-8
- FS_RECOVER Installation Tape . . . 4-8
- Using FS_RECOVER_INSTALL.CPL . . . 4-8
- Changes to Search Rules . . . 4-9
- ACL Requirements . . . 4-9
- Segment Requirements . . . 4-9
- Changes to PRIMOS.COMI . . . 4-10
- Allocating Disk Records for Tape Dumps . . . 4-10

Using FS_RECOVER . . . 4-12
- Recommended Strategy After a System Crash . . . 4-12
- FS_RECOVER Main Menu . . . 4-13
- Reading Crash Dump Tapes . . . 4-14
- Performing the Recovery Analysis . . . 4-16
- Recommendations for Running FIX_DISK . . . 4-18
- Automated FIX_DISK . . . 4-21
- Disk Manager Subsystem . . . 4-23
- Displaying the State of Currently Mounted Disks . . . 4-23

Appendices

A Generating a Crash Dump

How to Generate a Crash Dump Tape . . . A-1
- Halting the Machine . . . A-2
- Initiating a Partial Crash Dump . . . A-2
- 6250-bpi Crash Dumps . . . A-2
- Crash Dump Reels . . . A-2
- Tape Label Information . . . A-3

B FS_RECOVER Warning and Error Messages

Warning Messages at Invocation . . . B-1

Error Messages at Invocation . . . B-2

Runtime Warning Messages . . . B-3

C ***PRIMOS Disk Mount Warning Messages***

D ***Rev. 23.1 Publications***

Index

About This Book

The *Rev. 23.1 Software Release Document* provides a summary of both new and enhanced functionality to Prime® user software at Rev. 23.1. Most changes to Prime user software at Rev. 23.1 are documented in this software release document. This publication consists of four chapters and four appendices:

- **Chapter 1,** *Introduction*, includes special considerations for Rev. 23.1 and lists product retirements and new features.
- **Chapter 2,** *New Features for the User and Programmer at Rev. 23.1*, describes the new and enhanced functionality at Rev. 23.1 for the user and programmer.
- **Chapter 3,** *New Features for the Operator and Administrator at Rev. 23.1*, describes the new and enhanced functionality at Rev. 23.1 for the operator and administrator.
- **Chapter 4,** *Crash Recovery Facilities* describes two new facilities, Resident Forced Shutdown (RFS) and FS_RECOVER, for recovering disk partitions following a system crash.
- **Appendix A,** *Generating a Crash Dump*, describes procedures for performing a crash dump to tape for FS_RECOVER.
- **Appendix B,** *FS_RECOVER Warning and Error Messages*, lists error messages generated by FS_RECOVER.
- **Appendix C,** *PRIMOS Disk Mount Warning Messages*, lists messages returned when attempting to mount a disk following a system crash. These messages are of interest to users of RFS and FS_RECOVER.
- **Appendix D,** *Rev. 23.1 Publications*, lists the most recent editions of all books integral to Master Disk Revision 23.1.

This book contains the only documentation for most of the changes to PRIMOS at Rev. 23.1. Changes to MAGSAV/RST are documented in the Rev. 23.1 update to the *Operator's Guide to Data Backup and Recovery* (UPD10324–11A). A complete list of documentation for Rev. 23.1 is provided in Appendix D. This book should be used in conjunction with the most recent editions of PRIMOS user documentation and the online INFO files.

Prime Documentation Conventions

The following conventions are used throughout this document. The examples in the table illustrate the uses of these conventions.

Convention	*Explanation*	*Example*
Uppercase	In command formats, words in uppercase bold indicate the names of commands, options, statements, and keywords. Enter them in either uppercase or lowercase.	**SLIST**
Italic	Variables in command formats, text, or messages are indicated by lowercase italic.	**LOGIN** *user-id*
Abbreviations	If a command or option has an abbreviation, the abbreviation is placed immediately below the full form.	**SET_QUOTA** **SQ**
Brackets	Brackets enclose a list of one or more optional items. Choose none, one, or several of these items.	**LD** [**-BRIEF** **-SIZE**]
Braces	Braces enclose a list of items. Choose one and only one of these items.	**CLOSE** {*filename* **-ALL**}
Monospace	Identifies system output, prompts, messages, and examples.	`address connected`
Underscore	In examples, user input is underscored but system prompts and output are not.	`OK, RESUME MY_PROG`
Hyphen	Wherever a hyphen appears as the first character of an option, it is a required part of that option.	**SPOOL -LIST**
Ellipsis	An ellipsis indicates that you have the option of entering several items of the same kind on the command line.	*pdev-1* [. . *pdev-n*]
Subscript	A subscript after a number indicates that the number is not in base 10. For example, the subscript 8 is used for octal numbers.	200_8
Parentheses	Parentheses in command or statement formats are a required part of that format. Enter them as shown.	**DIM** *array* (*row, col*)

Introduction

1

This chapter summarizes the functionality changes to the PRIMOS operating system at Rev. 23.1. Specifically, it lists special considerations for Rev. 23.1, products being retired at this release, and new features at this revision of PRIMOS.

Special Considerations for Rev. 23.1

Installation

No new installation procedures are required for Rev. 23.1. To install Rev. 23.1, use the *Rev. 23.0 Software Installation Guide* (IDR10176–3XA). See the PRIMOS help file HELP*>PRIMOS.TEXT>REV231.HELP for the release numbers of independent products that are needed to support Rev. 23.1 functionality.

The following items should be considered when installing Rev. 23.1.

- The instructions for upgrading an existing system to a new revision of PRIMOS require two boot-from-disk operations. These instructions will successfully upgrade PRIMOS from 23.0 to 23.1, but it is possible to simplify the installation procedure for 23.1 by skipping the first boot from disk.
- When upgrading to Rev. 23.1 without reformatting disks, you will want to use ED or EMACS to modify the configuration file. You can modify the configuration file using the non-shared editor (NSED), but to use ED or EMACS, you may have to first re-share these products from the system console.
- Rev. 23.1 requires the translator family Release T3. Some Rev. 23.0 systems may still be running Release T2. Therefore, you may also need to refer to the installation procedures contained in the *Translator Family Software Release Document* (DOC10217–3PA) for installing compilers, libraries, and environment products.

- To install the nonchargeable translator family runfiles and libraries on a new machine, you should follow the procedures described in the *Rev. 23.0 Software Installation Guide*, then execute the command:

```
SSR -DEFAULT ENTRY$
```

- While running FIX_DISK remains an optional part of a software upgrade, the reliable performance of the RFS and FS_RECOVER crash recovery facilities provided with Rev. 23.1 depends heavily on all disks being in an uncorrupted initial state. Therefore, it is *strongly* suggested that you run FIX_DISK on all disk partitions as part of the upgrade to Rev. 23.1.
- The FS_RECOVER facility is supplied on a separate magnetic tape at Rev. 23.1, and can be installed independently of Rev. 23.1. This tape contains FS_RECOVER Version 2.0-21.0 (which designates the second version of FS_RECOVER, supported on all PRIMOS revisions 21.0 or greater). Only Version 2.0-21.0 (and later versions) of FS_RECOVER can be used with Rev. 23.1; the previous Independent Product Release (IPR) version (Version 1.0-21.0) of FS_RECOVER *will not work* in the Rev. 23.1 environment. The installation procedures for FS_RECOVER are described in the Crash Recovery Facilities chapter of this document. Note that installing FS_RECOVER modifies your system search rules.

Microcode Requirements

Prime 5310 and 5320 systems must be upgraded to at least CPU microcode Rev. G before attempting to boot Rev. 23.1. Prime 5330 and 5340 systems must be upgraded to at least CPU microcode Rev. H before attempting to boot Rev. 23.1. Attempting to boot Rev. 23.1 without performing these microcode upgrades may result in unpredictable halts and possible disk data corruption.

You can display your current microcode revision at the system console as follows:

```
<esc><esc>
DIR
MO ST
;
```

Pressing the escape key twice places the system console in maintenance processor mode (this does not effect executing processes). You type DIR to display the current microcode revision. You then type MO ST to return to PRIMOS mode. It may be necessary to type a semicolon (;) to get a prompt from the PRIMOS command processor.

Product Retirements

No products are being retired at Rev. 23.1; however, numerous products were retired at Rev. 23.0. Refer to the *Rev. 23.0 Software Release Document* for a list of those products.

The following products will be retired after Rev. 23.1; further use of these products is discouraged.

- AML buffer
- AMLC
- EDIT_PROFILE
- R-mode

Rev. 23.1 New Features

The new features of PRIMOS Revision 23.1 are listed below, grouped into two categories: those for the system operator and/or System Administrator, and those for the user and/or programmer. Chapter 2 discusses user and programmer features. Chapters 3 and 4 discuss operator and administrator features.

User and Programmer Features

The following enhancements are described in Chapter 2.

- Enhancements to PRIMOS commands:

CNAME command	New –REPORT option
CREATE command	New –REPORT option
LD command	New –RWLOCK and –TOTAL options
RDY command	New options that permit you to configure your prompts to include time, date, user and system information.

- TALK has been enhanced with some new facilities for greater user convenience.
- SHOW and WATCH, two new commands, permit a user to monitor the input and output activities of another user.
- Seven new informational command functions, executable from the command line or within a CPL program.

ATTACH_POINT	Returns current or origin attach point

COMO_INFO	Returns user's COMO file information
GROUP_LIST	Returns a list of the user's groups
SYSTEM_INFO	Returns system name, current PRIMOS revision, configuration information and number of logged-in users
SYSTEM_USAGE	Returns system time, memory and segments usage information
USER_INFO	Returns user's ID, process and project IDs, terminal type, and current task priority
USER_USAGE	Returns user's time, memory and segments usage information

- Two new string manipulation command functions, executable from the command line or within a CPL program.

REVERSE	Reverses its string argument
VALIDATE	Determines whether its string argument is of a valid length and contains valid characters for various types of PRIMOS names, for dates, and for decimal, octal and hexadecimal integers. Also determines if a string contains wildcard characters.

- Multiple attach points. A new subroutine, UT$COPY, permits users to establish multiple file units associated with their attach points or COMO file. This can improve performance of programs that frequently change attach points.
- ACL-protected shared memory. A new subroutine, MM$SHARE_APM, permits users to designate private memory segments for sharing by multiple processes. Access to these shared memory segments can be restricted by using an Access Category (ACAT).
- Corrections to user documentation.

Operator and Administrator Features

The following enhancements are described in Chapter 3.

- Configurable login prompt: a new facility that permits you to specify the system name, date, or other information as the login prompt. Provides new options to the START_LSR command, and a SET_LSR_DEFAULTS command for setting the login server defaults.
- Enhancements to PRIMOS Batch commands:

BATCH –START	New –DISPLAY_ALL option
INIT	New –SEM_INIT option

- MAKE command: new –REPORT option
- SPOOLER enhancements:

MPC4000 command	A new data line configuration command for the MPC4000 high speed print controller.
ASYNC command	New –MAXIMUM_SHUTDOWN_TIME option
HEADER command	New –NO_INITIAL_FF option

- RECORD_TO_PATH command: new command for converting a record address to a pathname.
- USAGE command: new –SYSTEM option
- SET_ASYNC command: new –DCD_LOGIN and –NO_DCD_LOGIN options for detecting the presence of a carrier signal at login.
- MAGSAV/RST enhancements: two new commands for index library management, INDEX_LIB_MANAGER and SEARCH_INDEX_LIB.
- NSEG configuration directive maximum value increased.
- MTRS configuration directive enhancement that permits configuring a different record size for each tape controller.
- EDIT_PROFILE enhancement to simplify duplicating a user profile.
- Corrections to operator and System Administrator documentation.

The following new features are described in Chapter 4.

- Resident Forced Shutdown (RFS): a new facility that shuts down local disk partitions following a system halt or hang. It identifies which partitions had transactions in process, and thus require FIX_DISK processing.
- FS_RECOVER: a new facility that analyzes a crash dump tape to determine what type of FIX_DISK recovery is required for each disk partition.

2 New Features for the User and Programmer at Rev 23.1

CNAME Command Enhancement

The CNAME command supports a new option, –REPORT (or –RPT), which reports on the success of the file change operation. This option should be placed after the *old_name* and *new_name* parameters, as follows:

```
CNAME old_name new_name –REPORT
```

Upon successful completion, CNAME displays the message:

```
"old_name" name changed to "new_name".
OK,
```

For further details on CNAME, refer to the *PRIMOS Commands Reference Guide*.

CREATE Command Enhancement

The CREATE command supports the new –REPORT (or –RPT) option, which reports the name of the directory created. This option should be placed after the *pathname* parameter that specifies the directory to be created, as follows:

```
CREATE pathname –REPORT
```

You can specify multiple CREATE options in any sequence. Upon successful completion, CREATE displays the message:

```
Directory "pathname" created.
OK,
```

For further details on CREATE, refer to the *PRIMOS Commands Reference Guide.*

LD Command Enhancements

The LD command supports two new options, –RWLOCK and –TOTAL. The –RWLOCK option displays the read and write locks on files and segdirs in the current directory. The –TOTAL option displays the total number of records used.

LD –RWLOCK

The –RWLOCK option displays the read and write locks on files and segdirs in the current directory. The –RWLOCK option may be specified with other LD options, as demonstrated in the second example.

Note that directories and ACATs do not have any read and write locks associated with them.

```
OK, LD -RWLOCK

<USERS>MORROW (ALL access)
133 records in this directory, 236 total records out of
quota of 0.

10 Files.
name              rwlock  type  rbf
------------------------------------------------------------
COMO                 sys  dam
X.CAM               none  cam
X.DAM               none  dam
X.SAM               none  dam
Y.CAM               updt  cam
Y.DAM               updt  dam
Y.SAM               updt  dam
Z.CAM                sys  cam
Z.DAM                sys  dam
Z.SAM                sys  dam

3 Segment Directories.
name              rwlock  type  rbf
------------------------------------------------------------
X.SEG               none  sseg
Y.SEG               updt  sseg
Z.SEG                sys  sseg
```

```
3 Directories.
name                type  rbf
--------------------------------------------------------------------------------
X.DIR               dir
Y.DIR               dir
Z.DIR               dir

3 Access categories.
name                type  rbf
--------------------------------------------------------------------------------
X.ACAT              acat
Y.ACAT              acat
Z.ACAT              acat


OK, LD -SIZE -RWLOCK

<USERS>MORROW (ALL access)
133 records in this directory, 236 total records out of
quota of 0.

10 Files.
name                    size    rwlock  type  rbf
--------------------------------------------------------------------------------
COMO                      -1       sys  dam
X.CAM                      1      none  cam
X.DAM                      3      none  dam
X.SAM                      1      none  dam
Y.CAM                      1      updt  cam
Y.DAM                      4      updt  dam
Y.SAM                      2      updt  dam
Z.CAM                     12       sys  cam
Z.DAM                      1       sys  dam
Z.SAM                      7       sys  dam

3 Segment Directories.
name                    size    rwlock  type  rbf
--------------------------------------------------------------------------------
X.SEG                      2      none  sseg
Y.SEG                      2      updt  sseg
Z.SEG                      2       sys  sseg

3 Directories.
name                    size     quota  type  rbf
--------------------------------------------------------------------------------
X.DIR                      5         0  dir
Y.DIR                    128         0  dir
Z.DIR                     77         0  dir

3 Access categories.
name                type  rbf
```

```
X.ACAT              acat
Y.ACAT              acat
Z.ACAT              acat
```

LD –TOTAL

The –TOTAL option displays the total number of records used by the listed items. If you specify –TOTAL, LD lists the names of the files and other objects in the directory, then displays the message:

```
Total records used: nn
```

where *nn* is the number of records. Note that –TOTAL shows the number of records used by the items you select for listing, and not necessarily all of the records used in that directory. For example, LD –DIR –TOTAL displays the total number of records used in subdirectories, and LD MYFILE –TOTAL lists only the total number of records used in MYFILE.

The total records number shown in the LD header may be larger than the number of records shown by the –TOTAL option. This is because the header total gives total storage, which includes record overhead, while the –TOTAL option gives total user data, which includes only user-visible record data.

For further details on the LD command, refer to the *PRIMOS Commands Reference Guide* and the *Rev. 22.1 Software Release Document.*

Configurable RDY Prompts

The RDY command has been enhanced to support dynamic and expandable prompts. You can now include variables in the text for any RDY prompt. These variables are expanded each time the prompt is displayed.

RDY prompt variables are expanded if you have enabled variable expansion. The –EXPAND_PROMPT (or –EP) option enables variable expansion. If you specify –EXPAND_PROMPT, any prompt variables present in the prompt text will be expanded at each subsequent command line, overriding the default value. The –NO_EXPAND_PROMPT (or –NEP) option disables variable expansion. If you specify –NO_EXPAND_PROMPT, prompt variables present in the prompt text are not expanded, and the prompt text appends to, rather than replaces the default value. The following example shows the use of the variable `%dn` (which displays the day of the week) and the –EXPAND_PROMPT option:

```
OK, RDY -LONG
OK 15:30:48  118.130  55.912  level 2
RDY -READY_LONG 'Today is %dn' -EXPAND_PROMPT
Today is Tuesday
RDY -LONG
Today is Tuesday
RDY -NO_EXPAND_PROMPT
Today is %dn 15:31:17  0.060  0.000  level 2
```

Note If you are using ECL (Edit Command Line) and specify a brief prompt in an ECL subcommand, that ECL command overrides the corresponding RDY command. Therefore, variable expansion of a brief prompt is only functional if you have not specified a value for that brief prompt in ECL. ECL has no effect on long prompts.

The OK prompt displayed with brief prompts is not normally displayed with long prompts, unless you have used ECL to specify a prompt value.

Variable prompt expansion works with both brief and long forms of all RDY prompts. Currently supported variables are:

%a	Current attach point
%c	CPU time since last prompt.
%dt	Quick date. Defaults to MM/DD/YY
%da	Abbreviated day name (e.g., Tue)
%dn	Full day name (e.g., Tuesday)
%ds	Day number with suppressed zero (e.g., 3)
%dz	Day number with leading zero (e.g., 03)
%en	Entryname portion of current attach point pathname
%ec	Error code of last command (useful for Error and Warning prompts)
%f	AM or PM flag: if the time is 00:00:00 to 11:59:59 then the string am is returned, else pm is returned
%h1	24 hour value with leading zero (e.g., 08)
%h2	24 hour value with suppressed zero (e.g., 8)
%h3	12 hour value with leading zero (e.g., 08)
%h4	12 hour value with suppressed zero (e.g., 8)
%i	IO time since last prompt

%l	Current command level. If the command level is 1 (the lowest command level) then a null string ("") is returned. If the command level is greater than 1 then the string "level *n*" is returned. If a static mode program was just executed and the command level is greater than 1, then the string "level *n*+" is returned.
%mi	Number of minutes with leading zero (e.g., 09)
%ma	Abbreviated month name (e.g., Jan)
%mn	Full month name (e.g., January)
%ms	Numeric month number with suppressed zero (e.g., 1)
%mz	Numeric month number with leading zero (e.g., 01)
%n	Number of users on the system
%p	ACL protection on the current directory (e.g., LURW)
%rv	PRIMOS revision
%rn	Remote name (i.e., the system name that current attach point exists on)
%se	Seconds with leading zero (e.g., 07)
%sn	System name
%t	Quick time (24 hour format HH:MM:SS)
%ui	User ID
%un	User number
%y2	Two-digit year number (e.g., 90)
%y4	Four-digit year number (e.g., 1990)
%/	CRLF (carriage return and linefeed)
%.	Global variable expansion. Syntax is *%.global–var%* (e.g., %.terminal_type$%) Returns the current value of a global variable.
%%	The percent sign (%). This allows the user to override the meaning of a prompt variable.
%–	Reserved for future use.
%[	PRIMOS command function call. Call must be in the format *%[command–function%]* where *command–function* is the EPF to be called as a function. This allows users to write and call their own command functions each time they are prompted for input. For example, a user could write an active function to check if he/she had mail. This would allow the user to be notified (at each prompt) if he/she had mail.

Note that command function calls may be nested and may also contain prompt variables.

For example: %[BEFORE %[AFTER %a '<'%] '>'%] would pass the user's current attach point to the AFTER function which would then pass the modified string to the BEFORE function.

%] PRIMOS active function call terminator token (see example above).

If the RDY command does not recognize a variable (for example, %xx) it returns the variable string as a literal. If the RDY command cannot expand a variable, it returns an error message in the variable that caused the error. Errors returned from RDY prompt variables are:

$BUFFER_TOO_SMALL$

The current attach point or entryname is longer than the maximum length supported by PRIMOS or a global variable expansion is larger than what is internally representable.

$CURRENT_ATTACH_POINT_UNAVAILABLE$

An undetermined error has occurred while trying to obtain the current attach point for a user.

$ENTRYNAME_UNAVAILABLE$

An undetermined error has occurred while trying to obtain an entryname for a user.

$ACL_UNAVAILABLE$

The ACL on the current directory was unable to be obtained.

$SYSTEM_NAME_UNAVAILABLE$

An undetermined error has occurred trying to obtain the current system name.

$USER_ID_UNAVAILABLE$

An undetermined error has occurred trying to obtain the current user name.

$ERROR_IN_VARIABLE_REFERENCE$

A global variable was referenced that does not exist.

$GLOBAL_VARIABLE_STORAGE_UNDEFINED$

Global variable storage has not been allocated for the current process (via DEFINE_GVAR).

$NESTING_TOO_DEEP$

The nesting of command functions has exceeded the limit of 32 levels.

$UNBALANCED_BRACKETS$

A prompt variable string contains a unequal number of %[and %] variables.

$IMPROPER_USAGE_OF_BRACKETS$

An improper ordering of the command function delimiter variables (%[and %]) was found (e.g., %]%[).

TALK Enhancements

Previously, when one user terminated a TALK session, both users' screens were immediately cleared. At Rev 23.1, the user that did not terminate the session retains the TALK screen display until they wish to clear it. To clear the screen, press any key.

TALK now provides a "busy signal" — a message that is returned when you try to talk to a user that is already engaged in a TALK session. Because of the way PRIMOS handles the message facility, this functionality is only available when calling a specific user number.

When you attempt to establish a talk session with a terminal that has set its message receive state to defer or reject (using the MESSAGE -DEFER or MESSAGE -REJECT command), the talk session fails, returning the message:

```
user is not logged in or is not accepting messages.
```

For further details on the MESSAGE command, refer to the *PRIMOS Commands Reference Guide*.

The talk server is now automatically assigned to the project ID DEFAULT, rather than the project ID TALK.

For further information on TALK, refer to the *PRIMOS Commands Reference Guide* and the *Rev. 22.1 Software Release Document*.

SHOW and WATCH Commands

The SHOW and WATCH commands enable a user to monitor another user's input and output stream. With this ability, the monitoring user can assist the watched user entering a sequence of commands or stepping through an application. Using SHOW and WATCH, the monitoring user can assist others from their own terminals without having to sit next to the user's terminal.

SHOW and WATCH permit a user to monitor the input and output activities of a single other user. The monitoring user process may be a local, NTS, or remote user. The monitored user process may be local, NTS, remote or phantom. The

user to be monitored issues a SHOW command to permit monitoring. The monitoring user then begins monitoring that user by issuing a WATCH command.

SHOW Command

The SHOW command allows local terminal users, NTS users, phantom users, and remote users to grant privileges to other users to monitor their own input and output. This command grants the specified user the privilege to monitor for either the duration of the invoker's login session or until the invoker disables access with one of the disable options. A user may grant privilege to only one user at a time. Subsequent SHOW commands replace the access granted by a previous SHOW command.

```
SHOW {option}
```

You must specify one and only one option for each SHOW command. The SHOW command takes the following options:

username

Permits any user with this *username* to watch your input and output stream.

–ALL

Permits all users to watch your input and output stream. However, only one user can watch at a time.

–DISABLE_ADMIN
–DA

Privileged option for system console. Prevents System Administrators from using the WATCH command. Any System Administrator watch session in progress continues unaffected.

Caution Once disabled, a coldstart of the system is required to re–enable System Administrator use of the WATCH command.

–DISABLE_GROUP
–DG

Privileged option for members of the .WATCH$ ACL group. Prevents users from watching your input and output stream via the .WATCH$ ACL group privilege. If a member user is watching you via .WATCH$ access when you issue this option, that watch session is immediately terminated. Once disabled, ACL-group access to your terminal remains disabled until you log out.

–DISABLE_USER
–DU

Prevents any user from watching your input and output stream, unless that user has access via ACL-group access or System Administrator access. If a privileged user is watching you when you issue this option, that watch session is immediately terminated. Once disabled, user access to your terminal remains disabled until you issue a SHOW –ALL command, or until you log out.

–LIST
–L

Displays the System Administrator, ACL-group, and user access privileges to watch your input and output stream. You enable user access using the –ALL option. You disable access using the –DISABLE_ADMIN, –DISABLE_GROUP, and –DISABLE_USER options.

–HELP
–H

Displays command options.

WATCH Command

The WATCH command allows a user process to monitor another user's input and output streams. The watched user's session is unaffected and uninterrupted while being watched. WATCH monitors the user's input and output stream, including both cominput and comoutput and terminal line input and output. It is possible to watch any local, NTS, remote user, or phantom user. The WATCH command enables a user to monitor phantom processes even though a phantom process does not have terminal input and output buffers.

The other user's input and output stream information are displayed on the monitoring user's terminal. The monitoring user cannot redirect this stream to a COMO file.

To use the WATCH command, access privilege must be granted either by the user to be watched or by the System Administrator. The monitored user grants access permission by issuing a SHOW command. The second way to receive WATCH access is by being a member of the .WATCH$ ACL group. Users in this ACL group can watch any user on the system. This type of access is helpful when assistance is needed and a SHOW command can not be executed (e.g., user needs help while in the middle of an application). The System Administrator can add or remove a user from this ACL group by using the EDIT_PROFILE command. System Administrators should limit this access and strictly control which users are members of this group.

```
WATCH [option]
```

You must specify one and only one option for each WATCH command. The WATCH command takes the following options:

username [–RAW]

Starts to watch the input and output of *username*. *username* must be the only user logged in with this user ID. If there are multiple users logged in with the same user ID, the WATCH command displays an error message. Without the –RAW option, non-printable characters are converted to at signs (@) before they are sent to your terminal; with the –RAW option non-printable characters are sent unchanged to your terminal, which may interpret them.

–USER_NUMBER *n* [–RAW]
–USER

Starts to watch input and output of the user with user number *n*. The user must be currently logged in. Use the STATUS USER command to obtain the user number of a user. Without the –RAW option, non-printable characters are converted to at signs (@) before they are sent to your terminal; with the –RAW option non-printable characters are sent unchanged to your terminal, which may interpret them.

–OFF

Stops the current watch session of the user issuing the command.

–STATUS
–STAT

Displays the user name and user number you are currently watching (if any) and the user name and user number of the user (if any) that is watching you.

–HELP
–H

Displays command options.

When a watch session begins, the WATCH command displays the following message at the watcher's terminal.

```
Watch session started on user username (user n)
```

A watch session terminates either when the watching user issues a WATCH –OFF command, or either user logs out. When the watch session is terminated with the –OFF option, the following message is displayed.

```
Watch session terminated on user username (user n).
```

WATCH Session Considerations

A user may only watch one user at a time. A watched user can watch another user, or two users can simultaneously watch each other. The system prevents circular looping of data; however, avoid WATCH loops that involve remote logins. Two users can not watch a third user simultaneously.

During a watch session, all data displayed at the watched user's terminal is also displayed at the watcher's terminal. Watchers can execute commands and receive output from their own processes during a watch session; however, this data is interspersed with the data received from the watched user's terminal.

The –RAW option sends non-printable characters, such as control characters and escape sequences, to your terminal unchanged. You must use the –RAW option to monitor use of EMACS, ECL, and some other subsystems. However, because the receiving terminal interprets these non-printable characters, unexpected results may occur, especially if the WATCH session involves two different terminal types.

The WATCH command monitors all characters placed in the output buffer. Passwords are not placed in the output buffer; therefore, WATCH can never see passwords. Similarly, WATCH cannot see any activity involving half-duplex. If the monitored user has specified COMO –NTTY, characters are not displayed on that user's screen, but *are* placed in the output buffer, and can therefore be monitored.

New Command Functions

Rev. 23.1 provides nine new command functions. The first seven of these functions return various types of user and system information. The last two functions are used for string manipulation. These command functions can be executed from the command line, as follows:

```
OK, TYPE [ATTACH_POINT -ORIGIN]
<TPUBS>MORROW
OK,
```

or they can be executed as functions within a CPL program. Here is a brief summary of the commands and their options. For a more detailed list consult the FUNCTIONS online help text.

User and System Information Functions

ATTACH_POINT [–ORIGIN]

The ATTACH_POINT function returns either the user's current attach point or origin attach point. By default it returns the current attach point. To return the origin attach point (the initial attach point at login), specify the –ORIGIN option.

COMO_INFO

The COMO_INFO function returns the current state of a user's como file setting. This is useful for opening up a como file without disturbing the current como file. If there is no open como file, COMO_INFO returns –TTY or –NTTY (depending on the current output state).

If there is an open como file, COMO_INFO returns a quoted string that contains the pathname of the current como file, –TTY or –NTTY (depending on the current output state), and –CONTINUE indicating that it is an active como file.

Standard use of this function would be:

```
&SET_VAR como_state := [COMO_INFO]
            como NEW_COMO_FILE
               .
               .
               .
            como [UNQUOTE %como_state%]
```

GROUP_LIST [*wildcard–pattern*]

The GROUP_LIST function returns a list of a user's groups, delimited by spaces. Optionally, the GROUP_LIST function may take a wildcard pattern. If such a pattern is supplied, GROUP_LIST will return only those groups which match the wildcard pattern. (Note that the leading period character (.) is not needed for a wildcard pattern.)

GROUP_LIST is functionally identical to the LIST_GROUPS command, except that it is designed to return the information as a quoted list.

```
OK, TYPE [GROUP_LIST]
            .ADMINISTRATION .TOOLS .PROJECTS
```

If a wildcard pattern (32 character max) is provided with the function, only those groups that match will be returned. You need not include the leading period character (.) when specifying a pattern.

```
OK, TYPE [GROUP_LIST a@]
                .ADMINISTRATION
OK, TYPE [GROUP_LIST ^tools]
                .ADMINISTRATION .PROJECTS
```

SYSTEM_INFO [*option*]

The SYSTEM_INFO function returns information about your system, information normally only provided by USAGE, PRIMAN, or the STATUS command. There are seven options available with this function (abbreviations for these options are indicated by capital letters):

–NAME Returns the current system name, as specified when it was booted. This is the default option.

–REVision Returns the current PRIMOS revision.

–USERS Returns the number of users currently logged in.

–PROCEssor Returns the Prime standard name of the system's CPU (e.g., 9955–II, 6355, 4150, etc.) A $UNKNOWN_CPU$ result indicates that the processor type is outside of the function's knowledge (this only happens with very old processors, or if older revisions of this command function are run on very new systems).

–MEMory Returns the amount of memory configured for the system. If the memory is less than 1megabyte, it will be returned in kilobytes (e.g., 512KB). If it is at least 1megabyte, it is returned in megabytes (e.g., 32MB).

–Num_CONTrollers Returns the number of disk controllers in use on the system.

–Num_DISKs Returns the number of disks in use on the system, from all controllers.

The –NCONT and –NDISK options rely on the G$METR information used by USAGE (i.e., the number of accesses since coldstart). If a disk (or controller) has *never* been used, it will not be counted.

If no option is given, the function defaults to returning the system name.

SYSTEM_USAGE [*option*]

The SYSTEM_USAGE function returns information about the system normally available only through USAGE or PRIMAN. There are seven options available with this function (abbreviations for these options are indicated by capital letters):

–UP_since Returns the boot time of the system in ISO format (e.g., 91–3–31.17:31:44.Sun).

–TIME Returns the total time of operation since boot, in HMS format (e.g., 4h7m24s). This is the default option.

–CPTime Returns the number of seconds of CPU time used since boot.

–IO_Time Returns the number of seconds of I/O time used since boot.

–MEMory Returns the amount of memory in use, in pages.

–WIREd_pages Returns the number of pages that are wired (cannot be flushed or re–allocated).

–SEGmentS Returns the number of segments currently being used.

If no option is specified, the default is to return the amount of time since boot (–TIME). No more than 1 option may be specified at a time.

USER_INFO [*user–number*] [*option*]

The USER_INFO function returns information about a user's process. This function normally defaults to your own process but may be used to examine other processes as well. There are six options available with this function (abbreviations for these options are indicated by capital letters):

–TYPE Returns the user's type, which is assumed to be one of the following: TERMINAL, REMOTE, CONSOLE, PHANTOM, FAM, SLAVE, SERVER, PROCESS, CHILD, NETMAN, OTHER. PROCESS is a label for the various system processes, such as the logout server or kernel processes. SERVER is used for such tasks as the name server, login server, etc.

–NUMber Returns the user's number. This is the default option.

–PROJect Returns the user's project ID.

–NAME Returns the user's login ID.

–Process_ID Returns the unique process ID of the user. This option cannot be specified when requesting information about other users — it is restricted to information on one's own process only.

–PRIority Returns your current task priority, either a number from 4 (highest) to 0 (lowest), IDLE (process on idle queue) or SUSPEND (process is suspended).

If no options are specified, the default is to return the user's number (–NUMBER). If a user number is specified, but no command option is given, the default is to return the user's name (–NAME). No more than 1 option may be specified at a time.

If a decimal user number is specified with the function call, the information on that user is returned instead. If no such user exists, the string $UNKNOWN$ is returned.

```
OK, TYPE [USER_INFO -NAME]
MORROW
OK, TYPE [USER_INFO 423 -TYPE]
$UNKNOWN$
OK, TYPE [USER_INFO 1 -TYPE]
CONSOLE
```

USER_USAGE [*user–number*] [*option*]

USER_USAGE is similar to SYSTEM_USAGE, but returns usage information for your own (or a specified) process. There are seven options that apply to this function (abbreviations for these options are indicated by capital letters):

–LOGIN Returns the date and time the process logged in, in ISO format. (e.g., 91–03–31.17:31:44.Sun).

–CoNneCTed Returns the time logged in for the process in HMS format (e.g., 5h12m4s). This is the default option.

–CPTime Returns the amount of CPU time (in seconds) used by the process since login.

–IO_Time Returns the amount of I/O time (in seconds) used by the process since login.

–MEMory Returns the number of non–shared pages currently in use by the process.

–WIREd_pages Returns the number of pages wired by the process.

–SEGmentS Returns the number of segments in use by the process.

If no option is specified, the default is to return the process's connect time (–CONNECTED). No more than 1 option may be specified at a time.

If a decimal user number is specified with the function call, the information on that user is returned instead. If no such user exists, the string $UNKNOWN$ is returned.

```
OK, TYPE [USER_USAGE -WIRED_PAGES]
340
OK, TYPE [USER_USAGE 90 -SEG]
$UNKNOWN$
```

Using the –CONNECTED option on User 1 is the same as using SYSTEM_USAGE with the –TIME option. Likewise, TYPE [USER_USAGE 1 –LOGIN] will produce the same date as TYPE [SYSTEM_USAGE –UP_SINCE].

```
OK, TYPE [USER USAGE 91 -SEG]
12
```

String Manipulation Functions

REVERSE *string–to–reverse*

REVERSE reverses its string argument. That is, [REVERSE 'ABC'] returns 'CBA'. One use for the REVERSE function is to determine a file suffix.

```
OK, TYPE [REVERSE [BEFORE [REVERSE 'LOGIN.CPL.OLD'] '.']]
```

which returns the string `'OLD'`. Taken step by step:

```
[REVERSE 'LOGIN.CPL.OLD'] returns 'DLO.LPC.NIGOL'
[BEFORE 'DLO.LPC.NIGOL' '.'] returns 'DLO'
[REVERSE 'DLO'] returns 'OLD'
```

VALIDATE [*option*] *string–to–validate*

The VALIDATE function allows the user to validate input from a CPL RESPONSE function. The following options are supported (abbreviations for these options are indicated by capital letters):

–TREEname	Validates that the argument is a valid PRIMOS treename.
–ENtryname	Validates that the argument is a valid entryname.
–USER	Validates that the argument is a valid user ID.
–PROJect	Validates that the argument is a valid project ID.
–GROUP_id	Validates that the argument is a valid ACL group ID.
–PassWord	Validates that the argument is a valid password. Wildcards not accepted.
–DATE	Validates that the argument is a valid date. Date can be in USA form (MM/DD/YY), ISO form (YY–MM–DD), or Visual form (DD Mmm YY); a Visual form date must be a quoted string. Checks for valid character types and numeric values, including leap-year check. Validates complete date strings as supported by CL$PIX, including time of day and day of week; day of week is only validated to be a legal value, not necessarily the correct value for that date. The year field can be omitted or given as one, two, or four digits. Leading zeros are not required. Wildcards not accepted.
–DECimal	Validates that the argument is a valid decimal integer. Wildcards not accepted.

–HEXadecimal Validates that the argument is a valid hexadecimal integer. Letters can be uppercase or lowercase. Wildcards not accepted.

–OCTAL Validates that the argument is a valid octal integer. Wildcards not accepted.

–No_WildCard Disallows wildcard characters in any arguments.

These options only indicate that the input argument is valid (that is, that it contains valid characters and is the proper length). The VALIDATE function does not verify the existence of an object. For instance, [VALIDATE –USER FOOBARHEAD] returns TRUE indicating that FOOBARHEAD is a valid user ID. It does NOT verify that user FOOBARHEAD actually has a user profile on the system. For further details on acceptable values, refer to the CL$PIX appendix to the *Subroutines Reference II:File System*.

The –NO_WILDCARD (or –NWC) option may be used to disallow wildcard characters in arguments. By default wildcards are accepted, except where indicated. For example, [VALIDATE –TREE LOGIN.@] returns TRUE, whereas [VALIDATE –TREE LOGIN.@ –NO_WILDCARD] returns FALSE.

Multiple Attach Points

An application program that regularly attaches back and forth between two different directories must pay the overhead associated with frequently changing its current and home attach points. At Rev 23.1, PRIMOS provides a new subroutine which allows applications to duplicate their attach points for later reference, in order to reduce overhead associated with repeated attach operations.

The new subroutine, UT$COPY, permits you to copy file unit information for an attach point, como file, or open file to another file unit. This duplicate file unit can be either user-specified or selected by PRIMOS.

Usage

```
DCL UT$COPY ENTRY (FIXED BIN, FIXED BIN, FIXED BIN)
RETURNS (FIXED BIN)

real_target_unit = UT$COPY (source_unit, target_unit,
code);
```

Parameters

real_target_unit

RETURNED VALUE. UT$COPY returns the real target unit number. UT$COPY returns either the *target_unit* value, or, if you specified UT$NEW as the *target_unit* value, UT$COPY returns a PRIMOS-chosen unit number.

source_unit

INPUT. The unit number of the source Unit Table Entry (UTE). You can specify the unit number of an open file or one of the following values:

UT$COM	Como file unit.
UT$INI	Initial attach point unit.
UT$HOM	Home attach point unit.
UT$CUR	Current attach point unit.

target_unit

INPUT. The unit number of the target UTE. The *target_unit* can be either a user-specified unit number or a PRIMOS-chosen unit number. You can specify a unit number or one of the following values:

UT$NEW	Let PRIMOS select a free unit.
UT$COM	Como file unit.
UT$INI	Initial attach point unit.
UT$HOM	Home attach point unit.
UT$CUR	Current attach point unit.

If *target_unit* is UT$INI, UT$HOM or UT$CUR, then the value of *source_unit* must be an attach point. In addition, if *target_unit* is UT$INI, then *source_unit* must be a local attach point. This is because the initial attach point must be local. If *target_unit* is a file unit and it is in use, then an error message will be returned. If *target_unit* is an attach point unit (i.e., UT$INI, UT$HOM, or UT$CUR), then this unit will be overwritten. This is because the attach point units are always in use.

code

OUTPUT. Standard error code. Possible values are

E$OK	Successful completion.
E$BUNT	*source_unit* or *target_unit* is out of range.
E$ICUT	Incompatible unit table entry types (e.g., *target_unit* is an attach point, but *source_unit* is not., or *target_unit* is UT$COM, but *source_unit* is not an open file).
E$IANL	Initial attach point is not local (e.g., *target_unit* is the initial attach point (UT$INI), but *source_unit* is remote).

E$NOWR	Not open for write access (e.g., *target_unit* is UT$COM, but *source_unit* is open for read access).
E$UNOP	Source unit is not open.
E$FUIU	Unit table is full.
E$UIUS	Target unit is in use.
E$FIUS	Source unit can not be duplicated due to read/write concurrence lock.
E$PNTF	Remote machine does not support this subroutine.
E$RSHD	Remote disk has been shut down.
E$SHDN	Local disk has been shut down.

ACL-protected Shared Memory

Previously, the only way for two processes to share data was to use the SHARE command; this shared DTAR 1 memory among all user processes. Once the System Administrator had shared a DTAR 1 segment, all users had equivalent access.

With Rev. 23.1, another method is provided to share data. This method shares DTAR 2 memory segments among a list of users. Unlike the SHARE command, you can restrict user access to DTAR 2 memory by using ACL protection. In addition, you can easily specify which memory segments you wish to share by assigning a name to a group of segments.

Each user that wishes to access ACL-protected memory segments issues a call to the MM$SHARE_APM subroutine. Before calling this subroutine, you must establish an Access Category (ACAT) for the group of segments that you wish to share. When calling MM$SHARE_APM, you refer to the desired group of shared segments by the name of this ACAT. All users refer to the same group of shared segments by this ACAT name; access to the shared segments is controlled by the access rights defined for this ACAT.

The initial user of a group of shared segments calls MM$SHARE_APM and assigns a specific starting segment and total number of segments to the ACAT. Subsequent users of these shared segments call MM$SHARE_APM and specify the same starting segment number and total number of segments.

Usage

```
DCL MM$SHARE_APM ENTRY (FIXED BIN(15), CHAR(128) VAR,
        FIXED BIN(15), FIXED BIN(15), FIXED BIN(15),
        FIXED BIN(15), FIXED BIN(15));

CALL MM$SHARE_APM (key, pathname, access, num_to_share,
        rqst_strt_seg, found_strt_seg, code);
```

Parameters

key

INPUT. Must be set to K$INTO.

pathname

INPUT. Pathname of the ACAT that controls access to the shared segment(s). This must be an existing local Access Category on which the user has access rights. It cannot be password-protected. The initial user assigns this ACAT to a group of one or more segments; subsequent users refer to the group of segments by means of this ACAT.

access

INPUT. Type of access requested. The following values are permitted:
K$READ Read-only access
K$WRIT Write-only access
K$RDWR Read and write access

num_to_share

INPUT. The number of segments to share. Permitted values are 1 through 32.

rqst_strt_seg

INPUT. The requested starting segment number. You can specify a DTAR 2 static segment number, or you can specify 0. If you specify 0, PRIMOS assigns a dynamic segment as the starting segment. Shared segments are always contiguous and consecutive.

found_strt_seg

OUTPUT. The actual starting segment number.

code

OUTPUT. Standard error code. Possible values are:

E$OK	Successful completion.
E$APSM	Tried to change the name of or delete an ACAT that is currently protecting shared memory.
E$BKEY	*key* value is not K$INTO.
E$BPAR	Bad parameter. Illegal value for *pathname, access, num_to_share,* or *rqst_strt_seg*. This error code is also returned if you specify a number of segments that differs from the number of segments specified by the initial caller.
E$IREM	The ACAT is remote.
E$NMVS	Not enough segments available.
E$SGIU	Requested a starting segment that is already shared.
E$NTA	Requested a starting segment that is not active.
E$NCAT	*pathname* is not an ACAT.
E$ACNF	*pathname* does not exist.
E$NRIT	Insufficient access rights to ACAT.
E$NPDA	ACAT is a password directory. Password directories are not supported.

Discussion

When the initial user specifies the number of segments to share, MM$SHARE_APM assigns a contiguous, consecutive block of segments of exactly that length. If it is unable to assign all of the segments, it returns an error message and assigns no segments. Subsequent users must specify exactly the same number of segments and the starting segment number returned to the initial user in *found_strt_seg*. Allocated segments are initialized to all zeros.

Assigned segments are assigned from the users' private DTAR 2 space and paged to the paging disk. No file system space is used. Because ACL-protected segments are in private address space, more private segments and more VMFA segments are used. Each process that successfully calls MM$SHARE_APM uses a private DTAR 2 segment, either a static or dynamic segment, depending on the value of *rqst_strt_seg*. One VMFA segment is charged for each unique ACL-protected segment used, regardless of the number of users of that segment. The System Administrator must take care that the system limit, via the CONFIG directive NVMFS, and the number of private static and dynamic segments, via CONFIG_USERS, reflect this new usage.

Segments remain shared until the last user process releases the shared memory by logging out or using the ICE command to initialize its command environment. The paging space and physical memory are released, and the ACAT is disassociated from the segments. Data in shared segments is *not* saved to disk. It is the responsibility of the user program to save data in shared memory. Data in shared memory is also *not* protected during system crashes, shutdowns, halts, and hangs. It is the responsibility of the user program to provide appropriate condition handlers.

Changes to the access privileges to shared memory do not affect users currently sharing those segments. However, once the ACL protection is changed, no other users are allowed to share the segments until all current users have released them.

Documentation Corrections

The following are changes to existing PRIMOS user and programmer documentation that reflect corrections to PRIMOS software and documentation. These changes do not provide new functionality.

C Compiler

If you specify the **open( )** statement with an openMode = 1, it opens the file for both reading and writing. You cannot open a file for write-only using the C open statement. This is a correction to the *C User's Guide* which incorrectly states that you can use **open** to open a file for write-only operations. To open a file for write-only operations, use the OPEN$A subroutine.

When passing character or integer array pointers between modules in different languages, use 64V mode, rather than 32IX mode, for the C language. An array declared as **short array[ ]** is not the same as an array declared as **short *array** in 32IX mode. Pointers of the type **short array[ ]** cannot be passed in 32IX mode.

For information about C data types and their equivalents in other languages, refer to *Subroutines Reference I: Using Subroutines.*

CBL Compiler

The CBL compiler is available in 64V mode only. It does not support 32IX mode. This is a correction to Table 10-3 in the *PRIMOS User's Guide.*

CPL Functions

The ATTRIB function in CPL supports several options not listed in the *CPL User's Guide.* These options are listed in the Command Functions chapter of the *PRIMOS Commands Reference Guide.*

The WILD function in CPL supports several options not listed in the *CPL User's Guide.* These options are listed in the Command Functions chapter of the *PRIMOS Commands Reference Guide.* One of the WILD function's options, –ACAT, is incorrectly shown as –ACL in Chapter 7 of the *CPL User's Guide.* The WILD –FILE option can be abbreviated as –FL when run from either CPL

or the Command Processor; this abbreviation is not shown in the *PRIMOS Commands Reference Guide.*

CF$EXT Subroutine

To truncate a CAM file using CF$EXT, you must have exclusive use of the file. If some other user has the file open, the operation fails and CF$EXT returns an E$FIUS error code. For further details, refer to *Subroutines Reference II: File System.*

EPF$MAP Subroutine

The K$COPY key of the EPF$MAP subroutine copies the segment-image of the file into *temporary* segments. At the conclusion of EPF$MAP processing, the linkage for the EPF is deallocated. Therefore, repeated invocations of EPF$MAP do not map to the same segments and may result in an illegal segno$ error. EPF$MAP can be invoked directly, or indirectly by calling EPF$INVK. For further details, refer to *Subroutines Reference II: File System,* and the *Advanced Programmer's Guide III: Command Environment.*

EPF$REG Subroutine

The correct structure for EPF$REG is as follows:

```
DCL 1 register_info,
        2  pathname CHAR(128) VAR,
        2  init_comline CHAR(1024) VAR,
        2  epf_database FIXED BIN(15),
        2  dependency_list (1:8) CHAR(32) VAR,
        2  priority_search_list (1:8) CHAR(32) VAR,
        2  idl (1:8) CHAR(32) VAR;
```

The `dependency_list`, `priority_search_list`, and `idl` fields are arrays, not pointers to arrays. For further details, refer to *Subroutines Reference II: File System.*

SRTF$S, SETU$S, and MRG1$S Subroutines

The *spcls*(2) collating sequence parameters of these subroutines are not correct in the *Subroutines Reference IV* manual. The correct, complete list of *spcls*(2) parameters is:

0 Default (Prime ECS)

1 Prime ECS

2 EBCDIC

T$MT Subroutine

The second parameter, *buffno*, is either a user-supplied pointer to an input or output buffer, or a null pointer. The value you specify for *instr* determines whether to specify a pointer to a buffer or a null pointer.

The *code* parameter returns the error code value E$BNWD if you specified an invalid number of halfwords to *nhw*. An invalid number of halfwords is defined as *nhw* <=0 or *nhw* > the MTRS config directive value.

In Table 7-4, Controller ID, some of the drive type model numbers are incorrect. The correct drive numbers are as follows:

Version 9	Model 4594 quad density tape Model 4597 150MB 1/4" -SCSI tape Model 4598 8mm cartridge tape
Version 10	Model 4598 8mm cartridge tape
Version 12	Model 4594 quad density tape

T$MT is described in *Subroutines Reference IV: Libraries and I/O*; the MTRS config directive is discussed in the *New Features for the Operator and Administrator* chapter of this document.

RDLIN$ and WTLIN$ Subroutines

When accessing a remote file using these subroutines, the maximum length of a line of characters is 16,000 characters. Exceeding this maximum length results in an E$NBUF error code. This limit does not apply to files on local disk partitions. For further details, refer to the *Subroutines Reference II: File System.*

PRIMOS Error Codes

The following are additional PRIMOS standard error codes. For a list of the other PRIMOS error codes, refer to the *Advanced Programmer's Guide: Appendices and Master Index.*

`E$NDlS (363)` `No Dtar 1 Segments Available.`
Insufficient DTAR 1 segments are available to register this EPF. Decrease the size of the EPF or the number of registered EPFs.

`E$UMLE (364)` `Unexpected GetMutexLock error.`
An unexpected fatal error occurred. Coldstart the system and contact your PrimeService representative.

`E$ERNF (365)` `EPF registration level not found.`
This error occurred when attempting to unregister a registered EPF. When multiple versions of an EPF are registered, each is assigned a level; when attempting to unregister an EPF, you specified an unassigned level.

`E$APSM (366)` `Acat currently protecting shared memory.`
Used by the MM$SHARE_APM subroutine, as described in this document.

`E$UNLI (367)` `Specified user not logged in.`
Used by the WATCH command, as described in this document.

`E$MULI (368)` `Multiple users logged in with this user name.`
Used by the WATCH command, as described in this document.

`E$CWAT (369)` `Can not watch.`
Used by the WATCH command, as described in this document.

`E$UBW (370)` `User being watched by another user.`
Used by the WATCH command, as described in this document.

`E$ICUT (371)` `Incompatible open unit types.`
Used by the UT$COPY subroutine, as described in this document.

`E$IANL (372)` `Initial attach point is not local.`
Used by the UT$COPY subroutine, as described in this document.

`E$NOWL (373)` `Not open for write access.`
Used by the UT$COPY subroutine, as described in this document.

E$NDSM (374)

Not currently returned by PRIMOS.

E$DNTA (375) Invalid physical device number (off line?).

Used by the RECORD_TO_PATH command, as described in this document.

3 New Features for the Operator and Administrator at Rev. 23.1

Configurable Login Prompt

The LOGIN_SERVER has been enhanced at PRIMOS Revision 23.1 to support configurable and dynamically expandable LOGIN_SERVER prompts. Additionally the System Administrator may also control the way LOGIN_SERVER prompts are echoed if a null LOGIN line is entered. New options have been added to the START_LSR command to support these new features.

START_LSR Command Enhancements

The START_LSR command now supports the following command line syntax:

```
START_LSR [-PROMPT 'login_server_prompt']
          [-REDISPLAY_PROMPT]
          [-HELP]
```

-PROMPT

The -PROMPT option allows the System Administrator to specify the LOGIN_SERVER prompt when the LOGIN_SERVER is started up. The prompt may be up to 160 characters in length and can optionally contain RDY expandable prompt variables (for more information on RDY expandable prompt variables consult the New Features for Users and Programmers chapter earlier in this document). If the prompt contains blanks, enclose the prompt with single quotation marks.

-REDISPLAY_PROMPT

The -REDISPLAY_PROMPT option specifies whether or not the LOGIN_SERVER redisplays the LOGIN_SERVER prompt if a NULL command line is entered.

–HELP

The –HELP option displays the command-line syntax for the START_LSR command.

–PROMPT Examples

```
OK, START_LSR -PROMPT 'Please login.'
```

generates the following login server prompt:

```
Please login.
```

```
OK, START_LSR -PROMPT 'Please login to %sn.'
```

generates the following login server prompt (here PLATO is the system name):

```
Please login to PLATO.
```

```
OK, START_LSR -PROMPT 'Please login to %sn  (%dt %t)'
```

generates the following login server prompt:

```
Please login to PLATO  (4/15/91 12:34:56)
```

If you do not specify the –PROMPT option, and the LOGIN_SERVER defaults have not been specified (see the section Setting Up LOGIN_SERVER Defaults below), then the default prompt of `Login Please.` is used and prompt echoing is disabled (see below).

–REDISPLAY_PROMPT Examples

Example 1: START_LSR with the –REDISPLAY_PROMPT option causes the login server to redisplay the login prompt after each carriage return, error, or invalid login, as shown in the following example:

```
OK, <cr>
Login Please.
OK, <cr>
Login Please.
OK, junk
Login Please.
ER!
```

Example 2: START_LSR without the -REDISPLAY_PROMPT option causes the login server to only display the login prompt after the initial carriage return or error, as shown in the following example:

```
OK, <cr>
Login Please.
OK, <cr>
OK, <cr>
OK, junk
ER! <cr>
OK, <cr>
```

If you do not specify the -REDISPLAY_PROMPT option, the LOGIN_SERVER will not echo the LOGIN_SERVER prompt on null login command lines unless it has been made to be the default (see section below on Setting Up LOGIN_SERVER Defaults).

Example 3: You can specify both the -PROMPT and the -REDISPLAY_PROMPT options to the START_LSR command, as shown in the following example:

```
OK,START_LSR -PROMPT '%sn login at: %t' -REDISPLAY_PROMPT
PLATO login at: 12:34:56
OK, <cr>
PLATO login at: 12:34:57
OK, <cr>
PLATO login at: 12:34:58
OK,
```

Setting Up LOGIN_SERVER Defaults

The System Administrator may set up LOGIN_SERVER defaults. The defaults are the LOGIN_SERVER prompt and the prompt redisplay characteristics that the LOGIN_SERVER is to use when it is started at coldstart or when it is restarted (via START_LSR) after it has been shut down.

Setting up LOGIN_SERVER defaults also allows the LOGIN_SERVER to inherit the desired prompt and redisplay characteristics when it is started up initially at coldstart.

To set up LOGIN_SERVER defaults simply invoke the SET_LSR_DEFAULTS tool in the SYSTEM directory.

Note The LOGIN_SERVER must be stopped before running SET_LSR_DEFAULTS. If you run the SET_LSR_DEFAULTS tool while the login server is running, PRIMOS returns the following error message:

```
OK, R SYSTEM>SET_LSR_DEFAULTS
File "SYSTEM>LOGIN_SERVER.CPL" could not be opened. (OPEN_FILE)


Error: The LOGIN_SERVER's defaults file can not be opened up for
writing.  The  LOGIN_SERVER must be shut down before running
SET_LSR_DEFAULTS.  If the LOGIN_SERVER is not running then check
the ACL on SYSTEM>LOGIN_SERVER.CPL.


ER, STOP_LSR
Really? yes
```

Sample Session 1

```
OK, R SYSTEM>SET_LSR_DEFAULTS

Retrieving current LOGIN_SERVER defaults...

Prompt redisplay: disabled
Login prompt has not been set and will default to "Login please."

Do you wish to change the defaults (y/n)? y

Prompt redisplay,  when  enabled,  causes the LOGIN_SERVER to redisplay the
LOGIN_SERVER prompt if a  null  or  blank  login  command  line is encountered.

Prompt redisplay is currently disabled.  Would you like to enable it (y/n)?  y

The current LOGIN_SERVER prompt is not set and is "Login please." by default.
Would you like to change it (y/n)? y

New LOGIN_SERVER prompt (<cr> = "Login please."): Please login to %sn.

Prompt redisplay: enabled
Login prompt: "Please login to %sn."

Ok to save new settings (y/n)? y

The defaults for the LOGIN_SERVER have been updated.  The changes  will not take
effect until  the LOGIN_SERVER is restarted or the system is coldstarted.
```

Sample Session 2

```
OK, R SYSTEM>SET_LSR_DEFAULTS

Retrieving current LOGIN_SERVER defaults...

Prompt redisplay: enabled
Login prompt: "Please login to %sn."

Do you wish to change the defaults (y/n)? y

Prompt redisplay, when enabled, causes the LOGIN_SERVER to redisplay the
LOGIN_SERVER prompt if a null or blank login command line is encountered.

Prompt redisplay is currently enabled. Would you like to disable it (y/n)? n

The current LOGIN_SERVER prompt is: "Please login to %sn."
Would you like to change it (y/n)? y

New LOGIN_SERVER prompt (<cr> = "Login please."): <cr>

Prompt redisplay: enabled
Login prompt has not been set and will default to "Login please."

Ok to save new settings (y/n)? n
Do you wish to change the defaults (y/n)? y

Prompt redisplay, when enabled, causes the LOGIN_SERVER to redisplay the
LOGIN_SERVER prompt if a null or blank login command line is encountered.

Prompt redisplay is currently enabled. Would you like to disable it (y/n)? n

The current LOGIN_SERVER prompt is not set and is "Login please." by default.
Would you like to change it (y/n)? y

New LOGIN_SERVER prompt (<cr> = "Login please."): Please login to %sn (%dt %t)

Prompt redisplay: enabled
Login prompt: "Please login to %sn (%dt %t)"

Ok to save new settings (y/n)? y

The defaults for the LOGIN_SERVER have been updated. The changes will not take
effect until the LOGIN_SERVER is restarted or the system is coldstarted.

OK,
```

After the defaults have been set, the System Administrator may simply issue the START_LSR command. The LOGIN_SERVER will then be started with the current defaults. The System Administrator may specify different options to the LOGIN_SERVER during startup but doing so overrides the preset defaults.

BATCH –START Command

The BATCH –START command has a new option, –DISPLAY_ALL (or –DPA). If you start Batch with this option, *all* users will be allowed to see *all* jobs in *all* queues. This option does not allow users to modify other users' jobs. If you start Batch without specifying this option, users can only display information about their own batch jobs (the default).

```
BATCH -START -DISPLAY_ALL
```

Since this is an option to the –START command of BATCH, it is important for System Administrators to decide whether this is a desirable feature to implement on their system. For further details, refer to the *Operator's Guide to System Commands* and the *Operator's Guide to the Batch Subsystem.*

Batch INIT Command

The Batch INIT command has a new option, –SEM_INIT (or –SI). If you specify this option, PRIMOS re-initializes only the semaphores used by the Batch Subsystem and does not remove any jobs from queues.

```
INIT -SEM_INIT
```

This option is provided as a recovery tool to enable the System Administrator to restart Batch without having to resubmit jobs. Use –SEM_INIT if the SEMFIL was left open after BATCH_SERVICE abnormally quit. Make sure that neither FIXBAT nor BATCH_SERVICE is running when using this option. For further details, refer to the *Operator's Guide to System Commands* and the *Operator's Guide to the Batch Subsystem.*

MAKE Command

The MAKE command has a new option, –REPORT (or –RPT). This option provides periodic updates on the progress of the MAKE operation. The screen display from a MAKE command with this option appears as follows:

```
OK, MAKE -DISK 100663 -FMT -RPT -PART PAGING
****  MAKE  <Rev. 23.1>  ****    Copyright (c) 1990,
Prime Computer, Inc.

Disk type must be one of the following:

SMD          80MB or 300MB removable
CMD          Cartridge module device
68MB          68 megabyte fixed media
158MB        158 megabyte fixed media
160MB        160 megabyte fixed media
600MB        600 megabyte fixed media
MODEL_4475   300 megabyte fixed media
MODEL_4714    84 megabyte fixed media
MODEL_4711    60 megabyte fixed media
MODEL_4715   120 megabyte fixed media
MODEL_4735   496 megabyte fixed media
MODEL_4719   258 megabyte fixed media
MODEL_4845   770 megabyte fixed media
MODEL_4721   328 megabyte fixed media
MODEL_4860   817 megabyte fixed media

Disk type? smd
Making     3 head partition PAGING
Tracks:                  823
Sectors per track:         9
Partition is using Reverse Sectoring
Partition is in -All_Controller Mode
Partition size in decimal records:      22221

Beginning format.
Format now 6% complete.
Format now 12% complete.
Format now 18% complete.
Format now 24% complete.
Format now 30% complete.
Format now 36% complete.
Format now 42% complete.
Format now 48% complete.
Format now 54% complete.
Format now 60% complete.
Format now 66% complete.
Format now 72% complete.
Format now 78% complete.
Format now 84% complete.
Format now 90% complete.
Format now 96% complete.
Format completed.

Checking for bad spots.
```

```
Bad spot checking now 6% complete.
Bad spot checking now 12% complete.
Bad spot checking now 18% complete.
Bad spot checking now 24% complete.
Bad spot checking now 30% complete.
Bad spot checking now 36% complete.
Bad spot checking now 42% complete.
Bad spot checking now 48% complete.
Bad spot checking now 54% complete.
Bad spot checking now 60% complete.
Bad spot checking now 66% complete.
Bad spot checking now 72% complete.
Bad spot checking now 78% complete.
Bad spot checking now 84% complete.
Bad spot checking now 90% complete.
Bad spot checking now 96% complete.

Done checking for bad spots.

Partition PAGING     created successfully.
```

SPOOLER Enhancements

MPC4000 Command

The Spooler subsystem has been enhanced to support the MPC4000 high speed print controller. A new data line configuration command, MPC4000, lets you configure the line from this print controller to a Dataproducts, Centronics, or PostScript printer.

The MPC4000 command specifies the high speed MPC4000 print controller, which supports the following printer interfaces:

- Dataproducts standard parallel interface
- Centronics standard parallel interface
- Parallel PostScript interface

The default settings are:

- No serial line support
- The eighth bit stripped from all data
- Odd parity

The MPC4000 command can take the following options:

GP*n*

A required option that identifies the printer. *n* is the port number on the parallel interface; it can take a value from 0 to 7, inclusive.

–BAUD *xxxxx*

Defines the line speed of the asynchronous communications line. This option is required only if the interface is a parallel/serial connect to a PostScript printer. If –BAUD is omitted, the line is not enabled.
The supported speeds (*xxxxx*) are:

75	110	134.5
150	300	600
1200	1800	2000
2400	3600	4800
7200	9600	19200

If this option is specified without the –SERIAL_PARITY option, the despooler will use ODD parity as a default.

–SERIAL_PARITY

Specify either –SERIAL_PARITY ODD or –SERIAL_PARITY EVEN. This option is used with the –BAUD option to inform the despooler that the interface is a parallel/serial connection to a PostScript printer. If this option is omitted, parity defaults to ODD. If the –BAUD option is omitted, the serial interface is not started.

–CENTRONICS

This option informs the despooler phantom to expect the different cable wiring pattern used by a Centronics interface.

–EIGHTHBIT

This option prevents the eighth bit from being removed. This option should be specified for graphic and certain other specialized applications. If this option is not specified, the eighth bit is is stripped from each data byte sent to the printer.

–TIMER *xxxx*

This option allows you to specify a sleep period for the despooler. The parameter *xxxx*, which is specified in milliseconds, allows the despooler to accommodate its transmission rate to the speed of the printer. The following table is a guide to selecting a timer speed to match the speed of the printer.

Printer Speed	*Timer Setting (xxxx)*
1200 lpm	2500
24 pg/m	2000
45 pg/m	1000
90 pg/m	500

If you do not specify this option the default value, 2000, is set. System load

can affect printer performance; if a printer is not running at its rated speed, you can decrease –TIMER in units of 100 milliseconds until the printer reaches its rated speed.

ASYNC –MAXIMUM_SHUTDOWN_TIME

The ASYNC data line configuration command has a new option, –MAXIMUM_SHUTDOWN_TIME *nn* (–MST *nn*). This option allows you to configure the number of seconds the despooler is to wait before automatically shutting down the spooler when the print buffer is full. *nn* is any time from 10 seconds up to, and including, 300 seconds. If you do not specify this option, the despooler waits for 300 seconds (5 minutes) by default.

HEADER –NO_INITIAL_FF

The HEADER printer environment command has a new option, –NO_INITIAL_FF (–NIF). This option suppresses the initial form feed that occurs when the spooler is started. It is of value when a printer is loaded with pre-printed stationery.

For further information, refer to the *Operator's Guide to the Spooler Subsystem.* Corrections to the *Operator's Guide to the Spooler Subsystem* are listed in the Documentation Corrections section of this chapter.

Data File Unit Number

The data file unit number of the file being printed is stored in the dynamic shared data area ENVCOM.DATA_FILE_UNIT_NUMBER. Programs compiled with the ENVCOM insert file can access this data area.

Double-sided Laser Printing

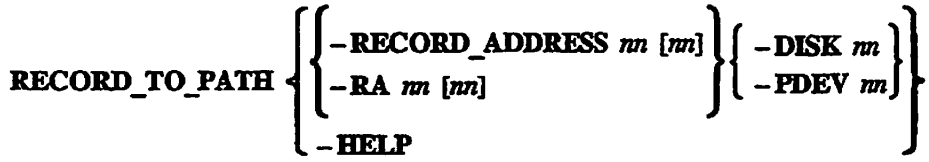
The print handler key S$START_COPY can be sent to the despooler phantom to control laser printers which are capable of double-sided printing.

RECORD_TO_PATH Command

Rev. 23.1 supports a new command, RECORD_TO_PATH, which takes a record address and returns the pathname of the file system object to which that record currently belongs. RECORD_TO_PATH works on local online disk partitions. It can be invoked by a System Administrator or any member of the .RAS$ ACL group, or at the system console.

One common use of RECORD_TO_PATH is to identify the file system object associated with a record cited by a recoverable disk read error. You can then decide whether to run FIX_DISK immediately, or continue running temporarily and avoid using this record. For example, you can copy the file system object to another location, and reroute access to that new file system object.

RECORD_TO_PATH has the following syntax:



The options are defined as follows:

–RECORD_ADDRESS *nn* [*nn*2] ... [*nn*16]
–RA

The address of the record, expressed as an octal number. You can specify up to 16 record addresses, separated by blanks (or commas), following this option. All addresses must be on the same local disk partition. RECORD_TO_PATH returns a separate pathname message for each record address.

–DISK *nn*

The local online disk partition on which the record resides. *nn* is the physical device number (pdev) of the partition, expressed as an octal number. This option is a synonym of the –PDEV option.

–PDEV *nn*

The local online disk partition on which the record resides. *nn* is the physical device number (pdev) of the partition, expressed as an octal number. This option is a synonym of the –DISK option.

–HELP

Displays the online help text for this command.

You must specify both the –RECORD_ADDRESS and the –DISK (or –PDEV) option for each call to RECORD_TO_PATH. RECORD_TO_PATH returns a pathname for each record address, up to sixteen pathnames. If the specified record is the first record of the file system object, RECORD_TO_PATH also returns the message:

```
First record of object
```

If the specified record is not used by any file system object, RECORD_TO_PATH returns the message:

```
Record address nn on device nn is NOT used.
```

If the specified record is inaccessible due to insufficient access rights, RECORD_TO_PATH returns the message:

```
Insufficient access rights.
```

If the disk partition that you specified is invalid or not currently online, RECORD_TO_PATH returns the message:

```
Invalid physical device number (off line?).
```

USAGE Command

The USAGE command has a new option, –SYSTEM (or –SYS). This option displays only the system metering information. The USAGE command with no options or with the –USER option still displays both system metering information and per-user metering information. For further details on USAGE, refer to the *Operator's Guide to System Commands*.

The following system meter display fields have not hitherto been documented:

PIO/S	Number of reads and writes per second caused by paging. This is *not* the same as the number of reads and writes to the paging disk.
%IOK	Percentage of CPU time used to interface with certain communications controllers, such as the LHC300 Ethernet controller.
%NTS	Percentage of CPU time used by the Network Terminal Server.

When reading the system meter display fields, bear in mind the following relationships: Each locate miss (LM/S) results in one disk read operation, which is counted as part of IO/S. Each page fault (PF/S) results in zero or more paging disk operations (PIO/S). The PIO/S are also counted as part of IO/S. The number in PIO/S includes some overhead and thus may be slightly larger than the actual number of paging I/O operations. Almost all remaining I/O operations are due to writes to the locate buffer. Discounting a very small number of assigned disk operations, the number of I/O operations (IO/S) is calculated as follows:

IO/S ~= PIO/S + LM/S + LocateWrites/S

Therefore, the number of locate buffer writes can be calculated as follows:

LocateWrites/S ~= IO/S – (PIO/S + LM/S)

For further details on the system meter display fields, refer to the *Operator's Guide to System Monitoring*.

SET_ASYNC Command

The SET_ASYNC command has a pair of new options, –DCD_LOGIN (or –DCD_LOG) and –NO_DCD_LOGIN (or –NO_DCD_LOG). These options enable the System Administrator to disallow login attempts on lines that have no carrier signal. If this option is set, an attempt to log in on a line that has no carrier signal results in the following error message displayed on the user's terminal:

```
Line must have DCD (Carrier) present for Login.
```

This failed login also displays a message on the system console:

```
Attempted login on line 37. No DCD (carrier) present.
```

If this option is not set (or reset to –DEFAULT), the system does not check for the presence of a carrier signal as part of the login procedure.

This option is set or unset by the System Administrator. This option has no effect if specified by a logged-in user. The SET_ASYNC –DISPLAY option has been modified to display whether or not a DCD (carrier signal) is required for login. For more information on SET_ASYNC refer to the *Operator's Guide to System Commands*.

MAGSAV/RST Enhancements

The MAGSAV/RST facility has been enhanced at Rev. 23.1 to permit index library management functions on the extended indexes. These features are described in more detail in the Rev. 23.1 update package to the *Operator's Guide to Data Backup and Recovery*.

INDEX_LIB_MANAGER Command

INDEX_LIB_MANAGER is the DRB extended index library manager command. It provides administration tools for extended index libraries created by MAGSAV or MAGRST.

```
INDEX_LIB_MANAGER [options]
```

INDEX_LIB_MANAGER options:

–INDEX_LIBRARY *pathname*
–DATE *date*
–DELETE [–NO_QUERY]
–IMPORT *pathname*
–REPORT
–REVERSE
–SORT_DTW
–SORT_VOLID
–VOLUME_ID *volume_id*
–WRITTEN_AFTER *date*
–WRITTEN_BEFORE *date*

–INDEX_LIBRARY *pathname*
–INDLIB

If you specify this option, INDEX_LIB_MANAGER uses the specified pathname as the index library directory. If you do not specify this option, INDEX_LIB_MANAGER uses the system default index library, specified in the DRB configuration file.

–DATE *date*

Specifies that only index files written on this date should be processed.

–DELETE [–NO_QUERY]
[–NQ]

Deletes the index files that match the specified criteria. The user will be queried before each index file is deleted, unless the –NO_QUERY option is specified.

–IMPORT [*pathname*]

Copies any index files from the alternative index libraries, as specified in the config file, into the main index library. Alternatively, you can supply the pathname of an index file to be imported, or the pathname of an index library that contains index files to be imported.

–REPORT
–RPT

Lists the index files that match the specified criteria.

–REVERSE
–REV

This option is only used with the –REPORT option. It specifies that the sort order is to be reversed.

–SORT_DTW
–SORTW

This option is only used with the –REPORT option. If specified, the index files will be sorted by the date they were written, with the most recent first. This is the default if any date selection criteria are also specified.

–SORT_VOLID
–SORTV

This option is only used with the –REPORT option. If specified, the index files will be sorted by volume ID. This is the default if no date selection criteria are specified.

–VOLUME_ID *volume_ID*
–VOLID

Specifies that only index files for the given volume ID should be processed. Wildcards may be used.

–WRITTEN_AFTER *date*
–WRA

Specifies that only index files that were written after this date should be processed.

–WRITTEN_BEFORE *date*
–WRB

Specifies that only index files that were written before this date should be processed.

SEARCH_INDEX_LIB Command

SEARCH_INDEX_LIB searches DRB extended index libraries, created by MAGSAV or MAGRST, for a specified search string. The search string may contain wildcards and can be a fully qualified pathname or a partial pathname. Where more than one saved object matches the specified search string, a list of all matches will be displayed. The Volume ID, Save Number and Checkpoint Number are also displayed for each occurrence.

```
SEARCH_INDEX_LIB  search_string  [options]
```

SEARCH_INDEX_LIB options:

–INDEX_LIBRARY *pathname*
–DATE *date*
–LATEST
–OLDEST
–REVERSE

–WRITTEN_AFTER *date*
–WRITTEN_BEFORE *date*

–INDEX_LIBRARY *pathname*
–INDLIB

If you specify this option, SEARCH_INDEX_LIB uses the specified pathname as the index library directory. If you do not specify this option, SEARCH_INDEX_LIB uses the system default index library, specified in the DRB config file.

–DATE *date*

Specifies that only saves written on this date should be searched.

–LATEST

This option specifies that only the most recent copy of the specified object should be listed.

–OLDEST

This option specifies that only the oldest copy of the specified object should be listed.

–REVERSE
–REV

This option specifies that the sort order is to be reversed, so that the saves of a files are listed with the newest first.

–WRITTEN_AFTER *date*
–WRA

Specifies that only saves that were written after this date should be searched.

–WRITTEN_BEFORE *date*
–WRB

Specifies that only saves that were written before this date should be searched.

NSEG System Configuration Directive Enhancement

NSEG specifies the number of segments available for all users. The maximum value of NSEG has been increased to allow up to 40000_8 (16,385 decimal) segments. The default value of 1776_8 (1022 decimal) remains unchanged. For further details, refer to the *System Administrator's Guide I: System Configuration.*

MTRS System Configuration Directive Enhancement

Prior to Rev. 23.1 you could only specify one argument to the MTRS directive in your system configuration file. At Rev. 23.1, you can specify up to three arguments for the MTRS directive. Each MTRS argument sets the maximum tape record size for one tape controller. Specify the MTRS arguments in the desired order, separated by blanks. The first MTRS argument sets the first tape controller (14 octal), the second MTRS argument sets the second tape controller (13 octal), and the third MTRS argument sets the third tape controller (12 octal). In this way you can modify the maximum tape record size for up to three tape controllers. The format for specifying this max tape record size is described in the *System Administrator's Guide I: System Configuration.*

If you specify no MTRS directive in the configuration file, PRIMOS sets all tape controllers to the system default. If you specify an MTRS directive with one argument, PRIMOS sets all tape controllers to that MTRS value. If you specify an MTRS directive with two arguments, the first and second tape controllers are set to the corresponding argument values, and the third tape controller (if present) is set to the second argument value. If you specify an MTRS directive with three arguments, each tape controller is set to the MTRS value specified in the corresponding argument.

EDIT_PROFILE Enhancement

Hitherto, when you duplicated a user profile by using the command:

```
ADD user2 -LIKE user1
```

the user profile was completely duplicated, including the source user's password. At Rev. 23.1, EDIT_PROFILE has been enhanced to permit you to specify a different password when duplicating a user profile, as follows:

```
ADD user2 -LIKE user1 -PW new_password
```

If you do not specify the -PW option, this command works as it did before.

Documentation Corrections

The following are changes to existing PRIMOS operator and administrator documentation that reflect corrections to PRIMOS software and documentation. These changes do not provide new functionality.

Network Terminal Service (NTS)

The two manuals for NTS have been issued in new editions containing numerous corrections. Make sure that the manuals you are using are the following editions:

- *NTS User's Guide* (DOC10117–3LA)
- *NTS Planning and Configuration Guide* (DOC10159–2LA)

DSM User's Guide

The following are corrections to the *DSM User's Guide* edition for Rev. 23.0 (DOC10061–3LA). Please make the corresponding changes to your documentation.

page xi

change: *Operator's Guide to Prime Networks* (DOC10114–11A)
to: *Operator's Guide to Prime Networks* (DOC10114–1LA) and its update package (UPD10114–11A)

change: *Programmer's Guide to Prime Networks* (DOC10113–11A)
to: *Programmer's Guide to Prime Networks* (DOC10113–1LA)

change: *NTS User's Guide* (DOC10117–2LA)
to: *NTS User's Guide* (DOC10117–3LA)

change: *Rev. 23.0 Networks Release Notes*
to: *Rev. 23.0 Prime Networks Release Notes*

page 2–7

change: EDIT_PROFILE
to: CONFIG_USERS

page 2–9

change: –SYSTEM–LOG
to: –SYSTEM_LOG

page 2–17

change: DSM*>LOGS>UMH>UNDELIVERED.LOG)
to: DSM*>LOGS>UMH>UNDELIVERED.LOG

page 3–5

change: `DSM initialization started`
to: `Primenet database initialized`

change: `DSM Server is now in steady state`
to: `DSM is now in a steady state`

page 4–7

change: Exit from <command name>
to: Exit from CONFIG_DSM

page 4–17

Delete menu options 2 (Nodes in CONFIGURATION GROUP) and 3 (Nodes in ALIEN NODE GROUP) and renumber remaining options

page 4–36

change: `"DSM*>CONFIG_FILES>DSM_DEFAULT.CONFIG`
to: `DSM*>CONFIG_FILES>DSM_DEFAULT.CONFIG`

page 5–7

change: USER
to: –USER

change: DISCARD
to: –DISCARD

page 6–9

change: LIST_LAN_NODES_DETAIL
to: LIST_LAN_NODE_DETAIL

page 6–10

change: `LOG_COLD,generated by SYSTEM_MANAGER on SYS1`
to: `LOG_COLD,generated by user SYSTEM on SYS1`

page 6–11

change: `DSM Server is now in steady state`
to: `DSM is now in a steady state`

change: UNKNOWN
to: DSM

change: `LOG_COLD,generated by SYSTEM_MANAGER on SYS1`
to: `LOG_COLD, generated by user SYSTEM on SYS1`

page 6–12

change: `DSM Server is now in steady state`
to: `DSM is now in a steady state`

page 7–8

change: LOT
to: PLOT

page B–2

change: LOG_TAPE: PRIMOS tape controller errors.
to: LOG_TAPE: PRIMOS tape controller errors and down line load messages.

page B–3

change: LTP
to: PRIMENET/PNX

page D–5

Add TRUST — Trust is the consistency of user names between nodes. It enables users on trusting nodes to access data on another node without issuing a user ID.

Operator's Guide to the Spooler Subsystem

The following are corrections to the *Operator's Guide to the Spooler Subsystem* edition for Rev. 23.0 (DOC9303–4LA). Please make the corresponding changes to your documentation.

page ix

change: *System Administrator's Guide, Volumes I, II,* and *III* (DOC10131–2LA, DOC10132–2LA, DOC10133–2LA)
to: *System Administrator's Guide, Volumes I, II,* and *III* (DOC10131–3LA, DOC10132–2LA with RLN10132–21A, and DOC10133–3LA)

change: *Operator's Guide to File System Maintenance* (DOC9300–5LA)
to: *Operator's Guide to File System Maintenance* (DOC9300–5LA with UPD9300–51A)

change: *Data Backup and Recovery Guide* (DOC10129–2LA)
to: *Operator's Guide to Data Backup and Recovery* (DOC10324–1LA with UPD10324–11A)

change: *Operator's Guide to Batch*
to: *Operator's Guide to the Batch Subsystem*

change: *Programmer's Guide to BIND and EPFs* (DOC8691–1PA)
to: *Programmer's Guide to BIND and EPFs* (DOC8691–1LA with UPD8691–11A and UPD8691–12A)

page x

change: *PRIMOS Commands Reference Guide* (DOC3108–7LA)
to: *PRIMOS Commands Reference Guide* (DOC3108–7LA with RLN3108–71A)

Add the following printer documents:
Prime 3200 Series High Performance Band Printers Operator's Manual (MAN11318–1LA)
Laser Printer Operator's Manual, Model 3410 (MAN10069–1XA)
Laser Printer Quick Reference Guide, Model 3410 (MAN10070–1XA)
Prime 3450 Laser Page Printer User's Guide (MAN11277–1LA)
Prime 3451 PostScript Laser Page Printer User's Guide (MAN11276–1LA)
Diablo 630 and IBM Graphics Printer Personality Reference Manual, Model 3410 (MAN10072–1XA)

page 2–11

Add the following text to the beginning of the paragraph at the top of the page: Attributes named in an environment file can be specified as mandatory, in which case the environment accepts a print request only if the SPOOL command includes all the...

page 3–11

change: –NO_LINEFEED abbreviated –NFL
to: –NO_LINEFEED abbreviated –NLF

page A–1

change: `SPOOL 'MYDIRECTORY PASSWORD>FILE`
to: `SPOOL 'MYDIRECTORY PASSWORD>FILE'`

page A–9

change: –TRUNIT
to: –TUNIT

page B–5

Add to S$START_COPY description: This is used with laser printers which are capable of double-sided printing.

Crash Recovery Facilities

4

Rev. 23.1 provides two new crash recovery facilities, Resident Forced Shutdown (RFS) and FS_RECOVER.

- Resident Forced Shutdown (RFS) attempts to shut down all local disk partitions following a system halt or hang. It performs a normal shutdown on those disk partitions that were not active at the time of the system crash, and thus do not require FIX_DISK processing. It suggests FIX_DISK processing for those local disk partitions that it could not successfully shut down.
- FS_RECOVER analyzes a crash dump to determine what type of FIX_DISK recovery is necessary. FS_RECOVER works in conjunction with AUTOPSY to analyze crash dumps and determine the integrity of the file system. It reduces the mean time to recover by eliminating unnecessary FIX_DISK processing.

These facilities are generally used together. Following a system crash, you would perform the following steps:

1. Generate a crash dump tape (as described in Appendix A)
2. Run RFS
3. Coldstart the system
4. Use FS_RECOVER to analyze the crash dump, generate FIX_DISK CPLs, and run FIX_DISK where recommended

Together, these two facilities can significantly reduce downtime of local disk partitions following an unexpected system event.

Note RFS and FS_RECOVER restore file system integrity. Neither of these facilities guarantees data integrity.

Resident Forced Shutdown (RFS)

Resident Forced Shutdown (RFS) attempts to shut down local disk partitions after a system halt or hang. It successfully shuts down those disk partitions that do not require FIX_DISK processing and identifies those disk partitions that require FIX_DISK processing before adding the disk during restart. It is not necessary to run RFS after a successful forced shutdown.

Only those disk partitions which had file system transactions in progress actually require FIX_DISK processing during restart; other active disk partitions will be shut down and therefore restarted without FIX_DISK processing. (FIX_DISK processing is also required if an uncorrected disk write error occurs, either while the system is running, or while performing RFS processing.) It is estimated that only 20 percent of active local disk partitions have transactions in progress at any given time. Therefore, limiting FIX_DISK operations to only those partitions can significantly speed the time required to restart the system.

Transactions are automatically defined by PRIMOS system software whenever a file system object is extended or truncated. No modification of user programs or procedures is required to use RFS.

Note All disks must be in a stable state for RFS to process them reliably. Therefore, when first installing Rev 23.1 (which contains RFS) on your system, you should make sure that no prior file system damage exists on your disks. You can do this by verifying the messages displayed when each disk is started, or by running FIX_DISK on all local disk partitions.

Running RFS

Following a system halt or hang, the System Administrator runs the RFS procedure from the system console. If you intend to use FS_RECOVER, you must generate a crash dump *before* running RFS. Crash dump procedures are described in Appendix A.

You invoke RFS in the following manner:

```
CP> SYSCLR
CP> RUN 662
```

If the system is hung, you must first stop the main processor. Press the escape key twice (<esc><esc>) to enter the maintenance processor, then issue the maintenance processor command STOP. If the system is already halted, you can omit these steps. Then execute the RFS procedure by issuing the SYSCLR command, then RUN 662. 662 (octal) is the entry vector for starting RFS. If RFS halts while executing, it can be restarted; it continues execution on the next disk partition.

RFS performs the following steps.

1. The RFS routine flushes all modified LOCATE buffers. This ensures that all disk partitions that do not have transactions in progress will be up-to-date when they are shut down. It also increases the chances of maintaining user data integrity on all other partitions even though they will not be able to be shut down properly. RFS restores file system integrity; it may not always restore data integrity.
2. RFS displays a message on the system console as it processes each local disk partition. This message contains the partition's name and pdev. RFS displays a message that describes the status of the each disk partition.
3. When RFS has completed, it displays the following message at the system console:

   ```
   *** From RFS: Shutdown completed, system halting...
   ```

   and then halts the system. The operator then should follow standard procedures for crash dump analysis and/or re-booting the system. A warmstart is not permitted after running RFS; you must coldstart the system.

RFS Console Messages

RFS may display the following messages on the system console.

`*** From RFS: Forced shutdown started!`

The RFS procedure has been successfully initiated. This is always the first message displayed by RFS.

`Disk (`*`nnnnnn`*`) started...OK.`

RFS successfully shut down this disk partition. *nnnnnn* is the pdev for the partition.

`Disk (`*`nnnnnn`*`) started...run FIX_DISK.`

RFS did not successfully shut down this disk partition because of file system corruption. You must run the FIX_DISK utility to restore this partition. *nnnnnn* is the pdev for the partition.

`Disk (`*`nnnnnn`*`) started...unwrite-protect & rerun RFS.`

RFS did not successfully shut down this disk partition because the disk partition is physically write-protected. Press the write protect button, then rerun RFS. *nnnnnn* is the pdev for the partition.

`Disk (nnnnnn) started...unable to mark successful shutdown, unwrite-protect & rerun RFS.`

RFS can successfully shut down this disk partition, but, because the disk partition is physically write-protected, RFS cannot mark it as successfully shut down. Press the write protect button, then rerun RFS. *nnnnnn* is the pdev for the partition.

`*** Locate Buffer links corrupt, run FIX_DISK on all partitions.`

RFS cannot determine the status of any disk partitions because the locate buffer links were corrupted during the halt or hang. You must run the FIX_DISK utility on all active local disk partitions.

`*** From RFS: Shutdown completed, system halting...`

The RFS procedure has completed, and the system is being halted. This is always the last message displayed by RFS.

What Is FS_RECOVER?

FS_RECOVER is a crash recovery tool provided with Rev. 23.1. It is an Independent Product Release (IPR) that is supported on all PRIMOS revisions 21.0 and higher (model number 8503FSR). It is an optional product at Rev. 23.1, functionally independent of Rev. 23.1 and installed separately. Installation instructions are provided in this chapter.

This section describes

- The effects of a system crash on your file system
- What FS_RECOVER does
- How FS_RECOVER works
- Some caveats related to FS_RECOVER

Effects of a System Crash on Your File System

A system crash is an unexpected event. It can happen while PRIMOS® is updating or changing the file system. If it does, it may be impossible to access some or all of the files on the partitions that were active at the time of the crash. The only way to to correct this problem is to run FIX_DISK on the affected partitions.

Note The term *file system*, as used here, refers to the data structures used by PRIMOS to find all the records for files on a partition.

What Does FS_RECOVER Do?

The main goal of FS_RECOVER is to reduce file system recovery time following a system crash. This allows you to make the file system available to users sooner. FS_RECOVER can also assess the general state of your file system and provide an automated interface to FIX_DISK, even if your system has not crashed.

If your system did crash and you took a crash dump, you can use FS_RECOVER to read and analyze the crash dump. FS_RECOVER determines

- Which partitions need to fixed *immediately*
- Which partitions need fixing that can be *deferred* to a more convenient time
- Which partitions were unaffected by the crash

FS_RECOVER also determines the correct FIX_DISK options for those partitions that must be fixed immediately and provides an automated facility for running FIX_DISK.

If your system has not crashed or if your system crashed but you did not take a crash dump, you can use FS_RECOVER to make a generalized assessment of the state of your partitions. FS_RECOVER determines which partitions are damaged, and which partitions are *clean*. (The term *clean partition*, as used here, refers to a partition which does not cause PRIMOS to generate a warning message at the time it is mounted, or added. Refer to Appendix C for a listing of these warning messages.)

FS_RECOVER also determines the correct FIX_DISK options for the damaged partitions and provides an automated facility for running FIX_DISK.

FS_RECOVER Using a Crash Dump Tape

When you reboot your system after a crash, you should allow PRIMOS.COMI to mount all your local disk partitions, but do not start any disk mirrors and do not allow users to log in.

Note If you correctly placed INIT_RECOVER.CPL within PRIMOS.COMI, (when installing FS_RECOVER) this is automatically accomplished, and you can invoke FS_RECOVER by entering Control-P when prompted to do so at coldstart.

When the system is running, use FS_RECOVER to read the crash tapes, and perform the recovery analysis.

When performing a crash dump recovery analysis, FS_RECOVER uses two major sources of data.

- The crash dump itself, which contains detailed information about what was happening on your system at the time of the crash
- The *current* state of the disk partitions

The current state of the disk partitions is available only if each disk is mounted. The current state information is merged with the crash dump information to form a recommendation for each partition that was mounted at the time of crash.

When analyzing the crash dump, FS_RECOVER looks for three types of information, as follows.

Crash type	The type of crash, which affects the types of recommendations FS_RECOVER makes for running FIX_DISK, is determined from the machine state.
Activity	FS_RECOVER identifies file system activity at the time of the crash in order to indicate where damage to the integrity of the file system may be.
Prior Corruption	FS_RECOVER looks for any information that might indicate that file system damage existed *prior* to the crash, such as flag bits set in the DSKRAT indicating that a disk was not cleanly shutdown on some previous occasion. However, it is important to note that all indications of prior damage are not guaranteed to be in the crash dump. Therefore, you are encouraged to keep your partitions clean and healthy.

Generally, FS_RECOVER analyzes all this information in less than ten minutes.

After the analysis is complete, FS_RECOVER prints a recommendation for each partition that was mounted at the time of the crash. Each recommendation includes three pieces of information:

- A list of pathnames for any files on the partition that were active at the time of the crash. The pathnames may or may not be complete, depending on the amount of file system information in the PRIMOS locate buffers at the time of the crash.
- A statement telling you:
  - If FIX_DISK needs to be run on the partition
  - What FIX_DISK options should be used

  - Whether you should run FIX_DISK immediately or if you can defer running FIX_DISK to a more convenient time

  A facility is provided to change the FIX_DISK recommendation, should you decide to do so.

- If a partition was mirrored, the recommendation will tell you which half of the mirrored pair is to be used as the primary when you restart the mirror with the MIRROR_ON command.

When the recommendations are complete, FS_RECOVER builds a CPL program for each partition requiring immediate FIX_DISK. These CPL programs are designed to be run by phantoms. FS_RECOVER then determines how many phantoms will be needed to execute all the CPL programs. This determination will take into account the number of available phantoms, the number of FIX_DISK sessions required, the number of disk drives containing partitions requiring FIX_DISK, and the PRIMOS limit on the number of assignable disks.

FS_RECOVER then tells you how many phantoms are required, and asks you how many phantoms you wish to use. After you have made that decision, FS_RECOVER creates a phantom called the FIX_DISK Monitor that controls the phantoms which perform the FIX_DISK sessions. These phantoms keep separate, date stamped, COMO files for each FIX_DISK session so you can monitor their progress and results. When all of the FIX_DISK sessions have completed, the FIX_DISK Monitor phantom logs out.

FS_RECOVER Without a Crash Dump Tape

You can use FS_RECOVER to make a generalized assessment of the state of your locally mounted partitions. If any one of these partitions are damaged, FS_RECOVER asks if you want to run FIX_DISK on the damaged partitions. If you answer YES, FS_RECOVER sets up for automated FIX_DISK the same way it does for a crash dump recovery analysis.

You can use FS_RECOVER without a crash dump. For example, if you just had a system crash but were unable to get a crash dump, you can take advantage of the automated FIX_DISK facilities of FS_RECOVER. You can also identify and repair partitions that had a defer recommendation from a previous crash dump analysis.

Considerations When Using FS_RECOVER

The crash dump recovery analysis portion of FS_RECOVER works best if you use it immediately after each crash. FS_RECOVER may not work correctly if you attempt to analyze an old crash dump or a crash dump that was taken before other crashes.

The following are other considerations for using FS_RECOVER.

- FS_RECOVER cannot always display the full pathnames of every file affected by a crash. The pathnames are generated using the contents of the locate buffers found in the crash dump. The more pathname information found in the locate buffers,the more complete the pathnames FS_RECOVER can display. Pathnames cannot be generated for CAM files on robust partitions however.
- The automated FIX_DISK facilities of FS_RECOVER cannot be used to repair the command device (COMDEV). File system damage on the command device must be repaired by running FIX_DISK with the –COMDEV option at the supervisor terminal.
- FS_RECOVER cannot be run by phantoms.

This version of FS_RECOVER is applicable to all PRIMOS revisions from Rev. 21.0 through Rev. 23.1.

Installing FS_RECOVER

This section discusses installation of FS_RECOVER on your system, including any changes you may have to make to the system.

FS_RECOVER Installation Tape

Prime distributes FS_RECOVER on a standard 1600-bpi, MAGSAV-format tape. This tape is included in your Rev. 23.1 package. You mount the tape on any tape drive and restore the contents into any convenient partition. Restoring the tape contents creates a directory named FS_RECOVER, which contains about 1500 disk records. You install FS_RECOVER from that directory.

Using FS_RECOVER.INSTALL.CPL

To install FS_RECOVER, attach to the FS_RECOVER directory and execute the FS_RECOVER.INSTALL.CPL file. The installation file copies FS_RECOVER>SYSTEM_DEBUG* to a top-level directory named SYSTEM_DEBUG* on your command device (COMDEV). If you have several command devices, you may want to modify FS_RECOVER.INSTALL.CPL to install FS_RECOVER on all of them. The installation process also copies two new search rules files into SEARCH_RULES*.

Changes to Search Rules

FS_RECOVER uses four search rules files:

AUTOPSY.SR
MAPS.SR
COMMAND$.SR
ENTRY$.SR

The FS_RECOVER.INSTALL.CPL file automatically installs the first two files in SEARCH_RULES*. The last two search rules files are part of standard PRIMOS and already exist. The installation modifies these two files as follows.

- The COMMAND$.SR search rule defines where PRIMOS looks for external commands. The default is the directory CMDNC0 on the COMDEV. The installation adds SYSTEM_DEBUG* to the list so that, as a minimum, COMMAND$.SR contains CMDNC0 and SYSTEM_DEBUG*.
- The ENTRY$.SR search rule defines where PRIMOS looks when it attempts to resolve a dynamic link. The installation adds SYSTEM_DEBUG*>AUTOPSY.RUN.

ACL Requirements

FS_RECOVER contains security checks to ensure that only the supervisor terminal (User 1), the user ID SYSTEM, or the System Administrator use FS_RECOVER. In addition, SYSTEM_DEBUG* and SYSTEM_DEBUG*>CRASH require some specific ACLs. These ACLs are shown in the following example.

When you follow this example, substitute your System Administrator ID for *system_admin*. To set the correct ACLs, enter the following:

```
OK, SAC <0>SYSTEM_DEBUG* SYSTEM:ALL system_admin:ALL $REST:LURX
OK, SAC <0>SYSTEM_DEBUG*>CRASH SYSTEM:ALL system_admin:ALL $REST:NONE
```

Segment Requirements

The user ID SYSTEM and the System Administrator's ID must be configured for at least 128 dynamic segments. Failure to provide this minimum limit may cause unpredictable results. When you invoke FS_RECOVER, it checks the number of dynamic segments configured and prints warning messages if the number is too small.

Changes to PRIMOS.COMI

In order to complete the installation of FS_RECOVER, you must change your PRIMOS.COMI to include running the INIT_RECOVER.CPL program in SYSTEM_DEBUG*. The placement of INIT_RECOVER.CPL within PRIMOS.COMI must occur *after* all local disk partitions are mounted, but *before* user logins are allowed:

```
STI -TZ 0500 -DLST YES                          /* Sets up time-zone information.
START_DSM                                       /* Startup DSM.
ADD_DISKS.CPL                                   /* Mount local disks.
R SYSTEM_DEBUG*>INIT_RECOVER.CPL -PAUSE         /* Invoke FS_RECOVER, if needed.
MAXUSR                                          /* Allow user logins.
```

Note If you omit the -PAUSE option, you will not be able to invoke FS_RECOVER while PRIMOS.COMI is running.

When PRIMOS.COMI invokes INIT_RECOVER.CPL, INIT_RECOVER first displays a header and then saves the PRIMOS maps. The -PAUSE option causes PRIMOS.COMI to display the following message and pause for thirty seconds to allow you to enter Control-P, aborting PRIMOS.COMI and automatically invoking FS_RECOVER.

```
Pausing briefly to allow you to enter CONTROL-P to invoke FS_RECOVER.
Otherwise, PRIMOS.COMI will continue.
```

If you do not enter Control-P at this time to invoke FS_RECOVER, this message displays and PRIMOS.COMI continues.

```
Wait completed, continuing with coldstart.
```

Allocating Disk Records for Tape Dumps

FS_RECOVER cannot work on the raw data contained on the crash dump tape. You must first put the data on the tape into a disk file. FS_RECOVER has special facilities to do this, but sufficient free disk records must exist. Since crash dump files can be rather large, you should set aside some dedicated space on a partition. Ideally, you should set aside this disk space in the <0>SYSTEM_DEBUG*>CRASH directory, but this is not a requirement; you can put the crash data file on any partition.

The amount of disk space required for a crash dump file varies with the system configuration and the type of crash dump. The crash dump procedure outlined in Appendix A is for a *partial* dump, which is all that FS_RECOVER usually needs. *Full* crash dumps are virtually never needed and take up considerably

more disk space. In either case, use the following guidelines for disk space planning purposes.

1. Use the STATUS SYSTEM command at the supervisor terminal to determine the kilobytes (KB) of memory in your system.

```
OK, STATUS SYSTEM

System STAN is currently running PRIMOS rev. 23.1
Copyright (c) Prime Computer, Inc. 1991
32768K bytes memory in use

OK,
```

2. If you are generating *partial* crash dumps (as outlined in Appendix A) go to Step 3.

   For *full* crash dumps, calculate the base number of disk records required, as follows, and go to Step 4. The base number of records for a full tape dump is equal to the KB of memory divided by two:

```
KB of memory
in the system
--------------    =  base number of disk records
      2
```

3. For *partial* crash dumps, calculate the base number of disk records required by using one of the following formulas and then go to Step 4.

   Use this formula if your system has 32768 KB *or less*:

```
(KB of memory) * (0.35) = base number of disk records
```

   Use this formula if your system has *more than* 32768 KB:

```
(KB of memory) / 4  =  base number of disk records
```

4. If your system is a 6150™, 6350™, 6450™, 6550™, or a 6650™ (a 6000 series machine), add 66 to the base number of disk records calculated in either Step 2 or Step 3. This number represents the total number of disk records you should set aside for a crash dump on these machines. (You can use the SYSTEM_INFO command function, described in this document, to determine the model of your system.)

Examples of Calculating Required Disk Records: If your system is a 6350 with 65536 KB of memory and you use partial crash dumps, the number of disk records to set aside is as follows:

```
( 65536 / 4 ) + 66  =  16450 disk records
```

If your system is a 2550™ with 8192 KB of memory and you use partial crash dumps, the number of disk records to set aside is as follows:

```
8192 * 0.35  =  2868 disk records
```

Using FS_RECOVER

You can invoke FS_RECOVER at the supervisor terminal, while logged in as the System Administrator, or under the user ID SYSTEM. After invocation, FS_RECOVER makes several integrity checks to ensure that is was installed correctly. If any of the checks fails, FS_RECOVER displays an error message and returns you to PRIMOS command level. Appendix B contains a list of FS_RECOVER error messages with explanations and corrective actions.

Recommended Strategy After a System Crash

If your system crashes, follow this procedure:

1. Generate a crash dump. (See Appendix A.)
2. If you are running PRIMOS Rev. 23.1 or later, run RFS *after* generating the crash dump tape. (RFS accomplishes a forced shutdown of PRIMOS and shuts down each partition in an orderly manner.)
3. Cold start your system.
4. PRIMOS.COMI executes until it encounters the INIT_RECOVER.CPL command line. It then displays the following message and pauses for 30 seconds.

```
Pausing briefly to allow you to enter CONTROL-P to invoke FS_RECOVER.
Otherwise, PRIMOS.COMI will continue.
```

   Enter a Control-P to abort PRIMOS.COMI and to invoke FS_RECOVER.

5. FS_RECOVER displays its Main Menu. Use Main Menu Option 3 to assess the health of your disk partitions.
   A. If your system crashed because of a forced shutdown or if you successfully ran RFS, all the partitions may be clean. If all the partitions are clean, exit FS_RECOVER and continue PRIMOS.COMI by entering CO CONTINUE 6.
   B. If any of the partitions are damaged, do *not* initiate automated FIX_DISK while you are in Main Menu Option 3. Instead, go back to the Main Menu and select Option 1 to read the crash tape. Then select Main Menu Option 2 to analyze the crash dump file. Execute all

recommended *immediate* FIX_DISK sessions and then continue PRIMOS.COMI by entering CO CONTINUE 6.

C. If the crash dump analysis indicates that there are deferrable FIX_DISK sessions, you can reinvoke FS_RECOVER at a convenient time later and use Main Menu Option 3 to repair the damaged partitions. Continue PRIMOS.COMI by entering CO CONTINUE 6 at this time.

If your command device (COMDEV) is damaged, you must use FIX_DISK at the supervisor terminal.

FS_RECOVER Main Menu

If the installation integrity checks pass when you invoke FS_RECOVER, FS_RECOVER displays its Main Menu and prompts you for a choice:

```
[FS_RECOVER Rev 2.0 Copyright (c) 1991, Prime Computer Inc.]

MAIN MENU:

(1) Read crash tapes
(2) Perform crash recovery analysis
(3) Display state of currently mounted disks

Enter a menu number, or (Q)uit or (M)enu:
```

You have several choices, as follows:

- Use Option 1 when you want to read a crash dump tape into a disk file.
- Use Option 2 to perform a file system recovery analysis on a crash dump file that you created with Option 1. You can then invoke automated FIX_DISK.
- Use Option 3 to assess the state of all currently-mounted local disk partitions. You can then invoke automated FIX_DISK.
- Enter ! *<PRIMOS command line>* to execute a PRIMOS command without leaving FS_RECOVER.
- Enter M to cause FS_RECOVER to redisplay the menu.
- Enter Q to leave FS_RECOVER and exit to PRIMOS command level.

Breaking Out of FS_RECOVER: When you select a Main Menu option, you can stop execution of FS_RECOVER at any time by using Control-P. The *only* exception to this is when you are selecting a choice from the FIX_DISK Menu. While you are in the FIX_DISK Menu, Control-P, ECL support, and PRIMOS command line support are disabled. If you do stop FS_RECOVER by entering a Control-P, you see the following:

```
**** Break! ****
(A)bort, (C)ontinue, or (R)eturn to Main Menu? A
OK,
```

You can abort FS_RECOVER, continue with the interrupted selection, or go back to the Main Menu. You can get back to the Main Menu also by simply entering Q or QUIT in most cases. For example:

```
Enter a menu number, or (Q)uit or (M)enu: 1

Mount the first reel of the crash tape(s) and enter the magtape unit number.
You may also enter:
                  -"! <PRIMOS command>"
                  -"Q" or "QUIT" to return to the main menu.
Tape unit (9 track): Q

MAIN MENU:
```

Executing PRIMOS Commands Within FS_RECOVER: In some places where FS_RECOVER prompts you for input, you can also enter PRIMOS commands. In many instances, as in the previous example, FS_RECOVER explicitly tells you that you may enter PRIMOS commands. To enter a PRIMOS command line from an FS_RECOVER prompt, precede the PRIMOS command line with ! (an exclamation point). Abbreviations, wildcarding, and iteration lists are fully supported. After the PRIMOS command completes, FS_RECOVER prompts for input.

Using ECL Within FS_RECOVER: The ECL environment within FS_RECOVER is totally separate from your PRIMOS ECL environment.

ECL is automatically enabled within FS_RECOVER except in these cases:

- ECL is not installed.
- You invoke FS_RECOVER from the supervisor terminal on a system running a PRIMOS revision *prior* to Rev. 22.1.

Reading Crash Dump Tapes

FS_RECOVER cannot read the raw data on the crash dump tapes. You must use Main Menu Option 1 to read the data from tape into a disk file before FS_RECOVER can analyze the data. The tapes need to be successfully read only once, but individual reels with unrecovered tape errors may be reread as many times as necessary. If you stop reading tapes at the end of a reel, you can leave FS_RECOVER and then come back at some later time and continue reading the tapes, starting with the next reel. Reels must be read in the order that they were written.

To read crash dump tapes, select Option 1 from the Main Menu. Follow the prompts to mount the first reel of the crash dump tapes on a tape drive and enter the tape drive unit number:

```
[FS_RECOVER Rev 2.0 Copyright (c) 1991, Prime Computer Inc.]

MAIN MENU:

(1) Read crash tapes
(2) Perform crash recovery analysis
(3) Display state of currently mounted disks

Enter a menu number, or (Q)uit or (M)enu: 1
Mount the first reel of the crash tape(s) and enter the magtape unit number.
You may also enter:
                  -"! <PRIMOS command>"
                  -"Q" or "QUIT" to return to the main menu.
Tape unit (9 track): 0
```

Checking the Tape Drive: When you enter a magtape unit number, FS_RECOVER attempts to assign the tape drive. If the assign fails, you get an error message followed by another prompt for a magtape unit:

```
Tape unit (9 track): 0
PRIMOS error code 39 while assigning MT0. Device in use.
Mount the first reel of the crash tape(s) and enter the magtape unit number.
You may also enter:
                  -"! <PRIMOS command>"
                  -"Q" or "QUIT" to return to the main menu.
Tape unit (9 track):
```

After assigning the tape drive, FS_RECOVER checks to ensure that a tape is mounted on the tape drive and that the drive is online and ready. If any of these checks fail, you get an error message followed by the magtape unit prompt.

```
Tape unit (9 track): 0
Device offline or not ready.
Mount the first reel of the crash tape(s) and enter the magtape unit number.
You may also enter:
                  -"! <PRIMOS command>"
                  -"Q" or "QUIT" to return to the main menu.
Tape unit (9 track):
```

Crash Dump File: When the magtape drive is online and ready, FS_RECOVER prompts for the pathname of the file you want to put the crash dump data into. Ideally, this should be a file in SYSTEM_DEBUG*>CRASH, but this is not a requirement; you can put the crash dump data file on any partition. Use a unique name for each crash dump file so that the file is easy to

identify. The recommended naming convention includes the system name, followed by a date/time stamp. For example, if your system is named MOLLY and the crash occurred on April 19, 1991 at 1:30 p.m., the recommended name for the crash dump data file is MOLLY.910419.1330 or *filename*.[DATE –FTAG].

Reading the Tape: After you enter the crash dump pathname, FS_RECOVER reads the tape. When the end of the crash dump is detected on tape, FS_RECOVER returns you to the Main Menu. If an end-of-tape occurs before the end of the crash dump, FS_RECOVER prompts for the next reel. At this point, you can mount the next reel and enter the magtape unit number:

```
End of reel 1; 32766 records read; 32766 records dumped; 0 errors.
Are there any more reels? YES
Tape unit number (9 track): 0
```

Performing the Recovery Analysis

After FS_RECOVER reads the tape, select Main Menu Option 2 (Perform Crash Recovery Analysis) after being sure that you meet the following requirements.

- A crash dump file must exist. That is, at some point you must have used Main Menu Option 1.
- When you select Main Menu Option 2, you must know the pathname of the working directory (the directory containing the FS_RECOVER CPL programs and the crash dump file) that you want FS_RECOVER to use.

FS_RECOVER Working Directory: The FS_RECOVER working directory is where FS_RECOVER expects to find the two CPL programs, RUN_FIX_DISK.CPL and FIX_DISK_MONITOR.CPL. FS_RECOVER also uses the working directory to keep COMO files and to build CPL programs for automated FIX_DISK. Prime recommends that you keep all your crash dump files in the working directory also, but this is not a requirement.

The default working directory is SYSTEM_DEBUG*>CRASH. However, you can create and use a different working directory. If you do, copy RUN_FIX_DISK.CPL and FIX_DISK_MONITOR.CPL from SYSTEM_DEBUG*>CRASH into the new working directory.

Here is an example of how to create a new working directory.

```
OK, A MFD 1
OK, CREATE CRASH.NEW
OK, COPY SYSTEM_DEBUG*>CRASH>RUN_FIX_DISK.CPL *>CRASH.NEW>==
OK, COPY SYSTEM_DEBUG*>CRASH>FIX_DISK_MONITOR.CPL *>CRASH.NEW>==
OK,
```

When you select Main Menu Option 2, FS_RECOVER prompts you to enter the pathname of the working directory and displays a default working directory pathname. To select the default working directory, simply press Return.

```
Enter pathname of working directory (default="<0>SYSTEM_DEBUG*>CRASH"): <cr>
```

Pathname of the Crash Dump File: Next, FS_RECOVER prompts you to enter the pathname of the crash dump file you want to analyze. If you just finished using Main Menu Option 1 to read crash dump tapes, FS_RECOVER uses the pathname of the file you read the tapes into as the default pathname. If you want to use the default pathname, simply press Return. Otherwise, enter the pathname of the crash dump file you want to analyze.

FS_RECOVER then attempts to load the crash dump, which takes about one minute.

Example of Doing the Analysis: Following is an example of the display when you select Option 2.

```
MAIN MENU:

(1) Read crash tapes
(2) Perform crash recovery analysis
(3) Display state of currently mounted disks

Enter a menu number, or (Q)uit or (M)enu: 2

--------------------------------------------------------------------------------
                          *** RECOVERY ANALYSIS ***

Enter pathname of working directory (default="<0>SYSTEM_DEBUG*>CRASH"): <cr>
Crashdump pathname: SYSTEM_DEBUG*>CRASH>MILO.121291.0100

(Beginning crashdump load, please wait...)
```

Session COMO File: After FS_RECOVER successfully loads the crash dump, it starts a session COMO file in the working directory. The name of the COMO file is always unique and consists of the crashed system's name and a date/time stamp.

```
(Beginning crashdump load, please wait...)

Your session COMO file is <0>SYSTEM_DEBUG*>CRASH>RES-C4.910405.100048.
```

Messages Indicating the Machine State: After FS_RECOVER starts the session COMO file, FS_RECOVER determines the machine state at the time of the crash. *Record this information in your System Log Book.*

The following messages indicate possible machine states:

```
The machine was stopped by a MASTER CLEAR.

The machine did not halt; it was STOPPED by the
Maintenance Processor.

The machine halted at x(0)/xxxxxx; xxxxxx+'0

PRIMOS executed a Slow Halt at x(0)/xxxxxx; xxxxxx+'0

PRIMOS stopped the machine using a Forced Shutdown.

The machine was stopped using the "SHUT ALL" command at
the System Console.
```

Messages During Analysis of Data: After determining the machine state, FS_RECOVER begins analysis of the data. Analysis can take up to ten minutes. During this time, you see several informational messages:

```
(Building Unit Info table, please wait...)
(Validating Disk Driver data structures, please wait...)
(Validating state of the Locate subsystem, please wait...)
(Validating Unit Table Hash, please wait...)
(Building nllock LOCKLIST database, please wait...)
(Building nllock owners database, please wait...)
(Validating any resident DSKRATs, please wait...)
```

Occasionally you may see other warning or caution messages interspersed with the informational messages. Refer to Appendix B for more information.

Recommendations for Running FIX_DISK

After FS_RECOVER completes the analysis, it presents a summary of each partition with a recommendation to run FIX_DISK. These recommendations fall into one of three categories:

Immediate FIX_DISK

You should run FIX_DISK before using the partition; file system and data integrity are compromised. FS_RECOVER will attempt to use either the –FAST option or the –PARTIAL option to minimize FIX_DISK session time. (The -PARTIAL option is supported but undocumented.) By default, FS_RECOVER builds CPL files to run any immediate FIX_DISK.

Deferred FIX_DISK

You can add the partition but file system integrity may be compromised. If no database recovery is required for the files on the partition, you can make the partition available for use immediately. However, at some convenient time, you *must* run full FIX_DISK on the partition.

Not Required

The partition was clean before the crash and the crash did not damage the partition. You should find all your partitions in this state after a successful forced shutdown or a successful invocation of RFS at Rev. 23.1. If no database recovery is required for the files on the partition, you can make the partition available for use immediately.

No Recommendation

If FS_RECOVER detects that a disk drive containing a partition that was mounted at the time of the crash has been repartitioned, no recommendation will be given.

Example of Immediate FIX_DISK: Here is an example of a partition requiring immediate FIX_DISK:

```
LDEV: '1 PDEV: '6062 NAME: <UNIX01>
Warning: The crashdump indicates 2 serious problems with this partition:
         Prior to the crash, the partition was not clean at ADDISK time.
         A file system transaction was in progress at the time of the crash.
Warning: Currently there is 1 serious problem with this partition:
         The disk was not clean at ADDISK time.

Activity File Type          Pathname
-------- ----------         --------
LT       DAM file           <UNIX01>HOPS>X.COMO

File Activity Codes:
L : file had modified unflushed records in Locate subsystem.
T : file may have had an in-progress transaction.

RECOMMENDATION: run "FIX_DISK -FIX"
```

Example of Deferred FIX_DISK: Here is an example of the summary for one partition requiring a deferred FIX_DISK:

```
LDEV: '2 PDEV: '3462 NAME: <QUALF2>

No file system activity indicated; schedule a FIX_DISK at your convenience.
```

In this example, there was no indication of file system activity or serious problems; an immediate FIX_DISK is not required. If no special database recovery is needed for the files on this partition, you can make it available to users. However, at some convenient time, you must run FIX_DISK to maintain the integrity of the partition's file system.

Changing a FIX_DISK Recommendation: After FS_RECOVER displays the summary and recommendation for a partition it asks you if you agree with the recommendation. If you answer YES, FS_RECOVER continues with the next partition summary. If you answer NO, FS_RECOVER enters the FIX_DISK Menu, which then asks you what you want to do with the partition. While you are in the FIX_DISK Menu, Control-P, ECL support, and PRIMOS command line support are disabled.

```
LDEV: '2 PDEV: '3164 NAME: <UNIX02>

No file system activity indicated; schedule a FIX_DISK at your convenience.
Is this what you want to do? NO

FIX_DISK MENU

(1) Do nothing to this partition.
(2) Run FIX_DISK, without the -FIX option.
(3) Run "FIX_DISK -FIX -FAST" on this partition.
(4) Run "FIX_DISK -FIX" on this partition.

Enter your choice: 4
```

After you enter a valid choice, FS_RECOVER continues with the next partition to be summarized.

FS_RECOVER Summary Display: After all the partitions have been individually summarized, FS_RECOVER displays a general summary of all the FIX_DISK recommendations. FS_RECOVER then asks you if all the recommendations are satisfactory. If you answer NO, FS_RECOVER repeats the individual partition summaries so that you can change recommendations for running FIX_DISK.

```
  *CURRENT*              CURRENTLY    TYPE OF
LDEV   PDEV   NAME       MOUNTED?     FIX_DISK NEEDED      COMMENTS
----   ----   ------     ----------   ------------------   --------
  0    6060   <UNIX00>   yes          none                 COMDEV
  1    6062   <UNIX01>   yes          immediate, full
  2    3164   <UNIX02>   yes          immediate, full

3 partitions analyzed, 2 partitions require FIX_DISK.
2 immediate FIX_DISKs, 0 deferrable FIX_DISKs.

Are these FIX_DISK recommendations satisfactory? YES
```

Automated FIX_DISK

If there are no recommendations for running immediate or deferred FIX_DISK, FS_RECOVER returns to the Main Menu. If there are deferred or immediate FIX_DISK recommendations and you answer YES, indicating that you are satisfied with the FIX_DISK recommendations, FS_RECOVER asks if you want to initiate automated FIX_DISK on all partitions requiring immediate FIX_DISK (except the Command Device (COMDEV)):

```
3 partitions analyzed, 2 partitions require FIX_DISK.
2 immediate FIX_DISKs, 0 deferrable FIX_DISKs.

Are these FIX_DISK recommendations satisfactory? YES
Do you want to initiate the immediate FIX_DISKs? YES
```

If all recommendations were for deferred FIX_DISK or if the only recommendation for immediate FIX_DISK was for the command device, FS_RECOVER returns to the Main Menu.

Administrative Setup for Automated FIX_DISK: If you answer that you want to initiate the immediate FIX_DISK recommendations, FS_RECOVER displays an Administrative Setup screen. If you are running FS_RECOVER from the supervisor terminal, the Administrative Setup screen asks if you want to stop the LOGIN_SERVER. Answering YES prohibits user logins after FS_RECOVER enables MAXUSR for the FIX_DISK phantoms:

```
                *** ADMINISTRATIVE SETUP ***

Do you want to stop the LOGIN_SERVER before starting FIX_DISK? NO
Forcing "MAXUSR ALL" for FIX_DISK sessions.
Attempting to startup the DISK_MANAGER.
```

If you are *not* running FS_RECOVER from the supervisor terminal, FS_RECOVER tells you to go to the supervisor terminal and enter the following commands:

```
                *** ADMINISTRATIVE SETUP ***

The DISK_MANAGER must be started up prior to initiating FIX_DISK phantoms.
Enter the following command at the System Console:

"ECL -OFF"
"DISK_MANAGER -START"

In order to allow FIX_DISK phantoms to login, enter the following command
at the System Console:

"MAXUSR -PUSR 222"
```

```
If you want to prohibit user logins while FIX_DISK is running, enter the
following command at the System Console:

"STOP_LSR"

Press <RETURN> after this is done and/or you are ready to proceed: <cr>
```

Automated FIX_DISK Configuration: After you leave the Administrative Setup display, FS_RECOVER creates a subdirectory within the working directory. FS_RECOVER then builds the CPL programs for automated FIX_DISK in this subdirectory.

Next, FS_RECOVER determines how many phantoms are necessary to execute all the CPL programs. It takes into account the number of available phantoms, the number of FIX_DISK sessions required, the number of disk drives containing partitions requiring FIX_DISK, and the PRIMOS limit on the number of assignable disks.

FS_RECOVER then asks how many phantoms you would like to use:

```
                    *** FIX_DISK SETUP ***

(Building CPL programs for automated FIX_DISK, please wait...)

All the programs which will control the FIX_DISK sessions are located in:
<0>SYSTEM_DEBUG*>CRASH>FIX.RES-C4.910319.164508

The 2 partitions requiring FIX_DISK reside on 2 different disk drives.
Both of these disk drives can be worked on in parallel.  This requires one
phantom per disk drive (each phantom will do ALL the required FIX_DISKs for
a given disk drive), plus one phantom to drive the FIX_DISK_MONITOR program.
If 3 phantoms are too much, fewer (down to a minimum of 2) may be used.

Enter the number of phantoms to use (2-3) or (Q)uit: 3
```

FIX_DISK Manager Phantom: After you tell FS_RECOVER how many phantoms to use, you are prompted to begin automated FIX_DISK. You can also quit or execute PRIMOS commands prior to beginning automated FIX_DISK.

```
FIX_DISK setup is now complete, and we're ready to begin.
Enter <RETURN> to begin, "QUIT", or "! <command>": ! m -all -now -force
The system will be available in about 20 minutes. Please standby...

Enter <RETURN> to begin, "QUIT", or "! <command>": <cr>
```

When you press the Return key, FS_RECOVER initiates the FIX_DISK Monitor phantom. The FIX_DISK Monitor then begins creating phantoms to run the FIX_DISK sessions.

Disk Manager Subsystem

When running FS_RECOVER from the supervisor terminal, FS_RECOVER automatically initiates a program called the DISK_MANAGER while you are in the Administrative Setup screen. If you are *not* running from the supervisor terminal, FS_RECOVER instructs you to manually initiate the DISK_MANAGER at the supervisor terminal.

The DISK_MANAGER program services certain commands for the FIX_DISK phantoms. Due to PRIMOS restrictions, commands such as ADDISK, SHUTDN, and DISKS are *privileged* and can only be executed from the supervisor terminal. Whenever a FIX_DISK phantom needs one of these privileged commands executed, it calls the supervisor terminal. The DISK_MANAGER program allows the supervisor terminal to *listen* for these commands and then execute them on behalf of the FIX_DISK phantom.

You can still use the supervisor terminal to execute PRIMOS commands with the exception of DELSEG, ICE, and ECL, but do not enter commands that take longer than a few seconds to execute, because the DISK_MANAGER can *listen* for commands from the FIX_DISK phantoms only when the supervisor terminal is not busy.

When the DISK_MANAGER program receives a command from one of the FIX_DISK phantoms, it displays the command, along with the results, on the supervisor terminal:

```
*** DISK_MANAGER at 12 March 91 15:32
*** Starting "AD 6062" for SYSTEM (user 110).
*** Finished "AD 6062" for SYSTEM (user 110).
```

Displaying the State of Currently Mounted Disks

Main Menu Option 3 is used to make a generalized assessment of the health of all currently mounted local partitions. During this assessment, FS_RECOVER recognizes only two states that a partition can be in, as follows:

Clean A clean partition is one in which the file system structures on the partition are completely intact. This is indicated by bits set in the partition's DSKRAT that tell PRIMOS whether or not the partition had been cleanly shutdown since its last full FIX_DISK session. If the bits are not set, PRIMOS displays a warning message when the partition is mounted. (Refer to Appendix C) However, there are exceptional instances when a clean partition can become damaged *after* it is mounted. As of Rev. 23.1, PRIMOS has specialized support to make information about these exceptions available to FS_RECOVER.

Damaged A damaged partition is one that was either not clean at the time it was mounted, or it was damaged after it was mounted. If the damage occurred after the partition was mounted and you are running PRIMOS Rev. 23.1 or later, FS_RECOVER will tell you the type of problem that damaged the partition.

The following is an example of the use of Main Menu Option 3.

```
MAIN MENU:

(1) Read crash tapes
(2) Perform crash recovery analysis
(3) Display state of currently mounted disks

Enter a menu number, or (Q)uit or (M)enu: 3
```

```
                        *** SHOW LOCAL DISKS ***

                        FIX_DISK
LDEV  PDEV  NAME        NEEDED? COMMENTS
----  ----- ----        ------- --------
0     6060  <UNIX00>    no      COMDEV
1     6062  <UNIX01>    full    *Not Clean*
2     3164  <UNIX02>    full    *Not Clean*

3 partitions displayed, 2 require full FIX_DISK.
```

FS_RECOVER then asks if you want to run FIX_DISK on the damaged partitions. If you answer YES, FS_RECOVER sets up for automated FIX_DISK the same way.

Appendices

Generating a Crash Dump

A

Follow the directions below to generate a crash dump tape. If you are running PRIMOS Revision 23.1 or later, you should then run RFS in order to cleanly shut down the disks. The last step is to coldstart the system. The following example illustrates the entire procedure:

```
CP1> SYSCLR
Really? YES
CP1> RUN 773
Enter P for PARTIAL dump, Enter F for FULL dump: P

Dump completed.
CP1> SYSCLR          /* Note: Skip the following if you don't have Rev 23.1.
Really? YES
CP1> RUN 662
.         .         .
.         .         . /* Output from RFS intentionally removed
.         .         .
*** From RFS: Shutdown completed, system halting...

CP1> SYSCLR          /* Note: We're rebooting now.
Really? YES
CP1> BOOT 14114
```

How to Generate a Crash Dump Tape

You can initiate a crash dump only from the supervisor terminal and then only if the CPU is not running. Therefore, for hangs or other situations in which PRIMOS did not halt, you must stop the CPU before you initiate the crash dump.

See your CPU Handbook for a complete discussion of halts, hangs, and crash tape dumps.

Halting the Machine

Use the following procedure to stop a 50 Series machine:

1. Turn the keyswitch on the Status Panel to the *enable* position.
2. Press the Escape key on the supervisor terminal twice. The maintenance processor should respond with the `CP>` or `CP1>` prompt.
3. Enter the STOP or the HALT command. The maintenance processor should respond with a `CPU halted at` message.

Initiating a Partial Crash Dump

Now that the system is halted, you can initiate a crash dump. Mount a 1600-bpi or 6250-bpi magnetic tape, with a write ring, on MT0 and enter the following command sequence:

```
CP1> SYSCLR
Really? YES          /* Not all PRIME machines will ask you this.
CP1> RUN 777         /* There may be a slight pause before the tape turns.
```

6250-bpi Crash Dumps

If you have a 6250 bpi tape drive on MT0 and if your system is running at PRIMOS Rev. 22.0.3 or later, you can write the crash dump tape at 6250 bpi. This takes less time and tape than a 1600 bpi crash tape dump. The argument to the RUN command is only slightly different from the previous example:

```
CP1> SYSCLR
Really? YES          /* Not all PRIME machines will ask you this.
CP1> RUN 773         /* You only need a partial dump.
Enter P for PARTIAL dump, Enter F for FULL dump: P
```

Crash Dump Reels

The crash dump may use only part of a single reel or it may take several reels. The number of reels required depends upon:

- The amount of memory in the system
- The type of crash dump generated (full or partial)
- The number of users logged in at the time of the crash
- The tape density used (1600 or 6250 bpi)

Tape Label Information

Remember to put a label on each reel. The label should include:

- The name of the system
- The date and time of the crash
- The PRIMOS revision that the system was running
- The tape density used (1600 or 6250 bpi)
- The reel number (*X* of *N* reels)

B FS_RECOVER Warning and Error Messages

Warning Messages at Invocation

The following warning messages may display when you invoke FS_RECOVER.

```
Warning: The system is currently running PRIMOS Rev
"xx.xx.xx", which is a non-standard revision stamp that
FS_RECOVER does not understand. Enter the Prime compatible
revision stamp with a format like "21.0.3".
Compatible revision stamp:
```

If you have a customized PRIMOS, the next time you rebuild you should use a standard format revision stamp: *MajorRev.MinorRev.FixRev.YourRev*.

```
Warning: Can't open text file "xxxxxx>xxxxx>xxxxx".
```

Fix AUTOPSY.SR search rules, which should contain three rules: SYSTEM_DEBUG*, SYSTEM_DEBUG*>HELP, and SYSTEM_DEBUG*>FRU. If that does not work, reinstall FS_RECOVER.

```
Warning: FS_RECOVER may not work because AUTOPSY is not the
same version.
```

Reinstall FS_RECOVER if any problems occur.

```
Warning: Can't access DSKRAT for PDEV 'xxxxxx <xxxxxx>.
```

Edit the ACLs on the DSKRAT for the specified partition to ensure that the System Administrator and SYSTEM have at least LUR access. This message will also appear if you are running PRIMOS Revision 23.0 or later, and have partitions that are not mounted at the root level.

Error Messages at Invocation

The following error messages may display when you invoke FS_RECOVER.

```
*** USAGE ERROR ***
FS_RECOVER cannot be invoked by phantoms.
```

Do not invoke FS_RECOVER from a phantom.

```
*** USAGE ERROR ***
FS_RECOVER can only be run by the System Console or System
Administrator.
```

Log in as SYSTEM or as the System Administrator and reinvoke FS_RECOVER.

```
*** FATAL ERROR ***
FS_RECOVER  does not support PRIMOS revisions prior to Rev
21.0.
```

Upgrade PRIMOS to at least Rev. 21.0.

```
*** FATAL ERROR ***
This version of FS_RECOVER does not support Rev xx.xx.
Upgrade FS_RECOVER.
```

Upgrade to latest version of FS_RECOVER.

```
*** FATAL ERROR ***
Error during link to AUTOPSY.  FS_RECOVER has not been
properly installed.
```

Reinstall FS_RECOVER.

```
*** FATAL ERROR ***
Error initializing AUTOPSY.  Configuration or installation
is incorrect.
```

Reinstall FS_RECOVER.

```
*** FATAL ERROR ***
FS_RECOVER and AUTOPSY versions do not match.  Installation
is incorrect.
```

This error indicates that your FS_RECOVER installation failed or that your ENTRY$.SR search rules are incorrect. Enter the PRIMOS ICE command first and then retry FS_RECOVER. If FS_RECOVER still fails, then reinstall FS_RECOVER. If the failure occurs again, use the LSR ENTRY$ command to examine your ENTRY$ search rules for more than one line containing a link to AUTOPSY.RUN. Problems have been seen with early versions of DTB and FS_RECOVER on the same system, because DTB sometimes inserts its copy of AUTOPSY into ENTRY$.SR.

```
**** FS_RECOVER default on-unit invoked for condition
"xxxxxx".
```

Open a COMO a file and enter the PRIMOS DMSTK command. Open a service call with PrimeService to report the problem.

Runtime Warning Messages

```
Warning: This crashdump is from a different PRIMOS Rev:
XX.XX.XX.XX
```

FS_RECOVER has detected that at the time of the crash, a different PRIMOS revision was running than is running now. This is important if the major revision is different, and can be important for some minor and fix revisions.

```
Warning: The crash dump is more than 24 hours old.
```

FS_RECOVER has detected that you are attempting to analyze a crash dump that is more than 24 hours old. It is recommended that you use FS_RECOVER on a crash dump as soon as possible.

```
Warning: This crashdump is from system "xxxxxx".
```

FS_RECOVER has detected that you are attempting to analyze a crash dump from another system. The analysis will continue, but automated FIX_DISK will not be done.

```
PRIMOS error code x while looking for xxx>xxx.CPL.  <PRIMOS
Message>
Either installation was incomplete, or wrong working
directory.
```

FS_RECOVER was attempting to attach to and verify the working directory. If the PRIMOS error message was `Not found`, then the specified working directory either does not exist or does not contain one or both of the required CPL programs. (Refer to the subsection entitled FS_RECOVER Working Directory in Chapter 3.) If the PRIMOS error message is something other than `Not found`, then the error probably occurred while attempting to attach to the specified working directory. Verify that the specified working directory exists and then check ATTACH$.SR search rules and ACLs. If the specified working directory is the default working directory and the CPL programs are missing, then reinstall FS_RECOVER.

```
Warning: All the file system disks did not start in a
known/clean state. This means that disk corruption may have
existed BEFORE the crash occurred.
```

FS_RECOVER has determined that at least one of the partitions that was mounted at the time of the crash was not clean. This means that there may be file system integrity problems on the partitions that will not show up during the crash analysis.

```
Warning: PDEV 'xxxxxx <xxxxx> has been repartitioned.
```

While merging the state of the partitions at the time of the crash with the state of the currently mounted partitions, FS_RECOVER detected that a disk drive had been remade/repartitioned.

C PRIMOS Disk Mount Warning Messages

The following messages may display when you add a partition to the system.

****** Disk *xxxxxx*: Not ready.**

Disk is not powered up or has not completed spin up. Ensure you are using the correct pdev.

***** Cannot read DSKRAT of disk *xxxxxx***

Disk is accessible, but DSKRAT header cannot be read. Ensure you are using the correct pdev. If your pdev is correct, call PrimeService.

***** Disk *xxxxxx*: Old partitions not supported.**

PRIMOS detects that you are attempting to mount a pre-Rev. 19 partition. If this is not so, ensure that you are using the correct pdev. If the pdev is correct, then ensure that the partition has been properly formatted.

***** Format of disk *xxxxxx* not supported by this revision of PRIMOS.**

PRIMOS detects that you are attempting to mount a partition whose version number is greater than the latest version supported by the version of PRIMOS you are running. This can happen if you upgrade PRIMOS, upgrade the partitions with either MAKE or FIX_DISK, and then attempt to install an earlier version of PRIMOS.

***** Proper shutdown of Robust Partition *xxxxxx* did not take place.**
***** Partition will be write protected.**

PRIMOS prints this message when you force-mount a damaged robust partition. Run FIX_DISK –FAST and remount the partition.

***** Robust Partition *xxxxxx* has not been properly shutdown.**
***** Fast Fix_disk has to be run before it can be added.**

PRIMOS displays this message when you attempt to mount a damaged robust partition. Run FIX_DISK –FAST and remount the partition.

***** Disk "*xxxxxx*" was not shutdown properly, Run FIX_DISK.**

PRIMOS displays this message, as of Revision 22.1, when you mount a standard partition that is not clean.

```
(Quota system may be incorrect; run FIX_DISK.)
```

Pre-Rev. 22.1 PRIMOS displays this message when you mount a standard partition that is not clean.

Rev. 23.1 Publications

D

This appendix lists all books that are integral to Master Disk Revision 23.1. For any books not related to a specific revision, see the *Guide to Prime User Documents* (DOC6138–9PA) and its supplement (RLN6138–91A), or type HELP DOCUMENTS.

This appendix lists Prime publications in tables by function: PRIMOS administration and operation, PRIMOS architecture and assembly, PRIMOS use and programming, communications, data management, languages, and editors.

The column heads in each table organize the information about each book within a table as follows (left to right):

- The *Book Title* column lists the book's title
- The *Most Recent* column contains information on the most recent revision at which the title has been published and the document number of that document.
- The *Also Required* column provides the document numbers for ordering any additional publications, such as release notes, updates, and full editions of the book, that work together with the document listed in the *Most Recent* column. Some of these additional publications may have titles that are different from the book title.

Note If the *Most Recent* publication is a release note (RLNnnnnn–nnA) or an update (UPDnnnnn–nnA), you must have the last full edition of the title (DOCnnnnn–nnA) and any updates to that edition that also were published prior to the most current release. If the *Most Recent* publication is a full edition (DOCnnnnn–nLA), you do not need to order any prior editions, updates, or release notes with it. A new edition contains all changes that have occurred since the last published full edition.

Table C–1. PRIMOS Administration and Operation				
Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
DSM User's Guide	23.0	DOC10061–3LA		
MAGNET User's Guide	22.0	UPD10156–11A	21.0	DOC10156–1LA
Operator's Guide to Data Backup and Recovery	23.1	UPD10324–11A	23.0	DOC10324–1LA
Operator's Guide to File System Maintenance	23.0	UPD9300–51A	22.1	DOC9300–5LA
Operator's Guide to System Commands	23.0	DOC9304–5LA		
Operator's Guide to System Monitoring	21.0	DOC9299–3LA		
Operator's Guide to the Batch Subsystem	21.0	DOC9302–3LA		
Operator's Guide to the Spooler Subsystem	23.0	DOC9303–4LA		
Operator's System Overview	22.0	DOC9298–3LA		
Rev. 23.0 Software Installation Guide	23.0	IDR10176–3XA		
Rev 23.1 Software Release Document	23.1	DOC10001–8PA	23.0 22.1 22.0	DOC10001–7PA DOC10001–6PA DOC10001–5PA
System Administrator's Guide, Volume I: System Configuration	23.0	DOC10131–3LA		
System Administrator's Guide, Volume II: Communication Lines and Controllers	22.1	RLN10132–21A	22.0	DOC10132–2LA
System Administrator's Guide, Volume III: System Access and Security	23.0	DOC10133–3LA		

Table C–1. PRIMOS Administration and Operation (continued)

Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
Prime 400 Handbook (Including the 350 and 500)	21.0	UPD8799–11A	20.0	DOC8799–1LA
Prime 850 Handbook (Including the 750)	22.0	UPD10063–12A	21.0 20.0	UPD10063–11A DOC10063–1LA
Prime 2250 Handbook	22.0	UPD10073–12A	21.0 20.0	UPD10073–11A DOC10073–1LA
2455 Handbook (Including 2350 and 2450)	22.0	UPD10086–21A	21.0	DOC10086–2LA
Prime 2550 Handbook	19.4.3	UPD8638–11A	19.2	DOC8638–192L
2755 Handbook (Includes 2550, 2655)	22.0	UPD8638–31A	21.0	DOC8638–3LA
4150 Handbook (Including the 4050, 2850, and 2950)	22.1	RLN10162–21A	22.0.2	DOC10162–2LA
4450 Handbook	22.0	DOC10227–1LA		
5300 Family Handbook	22.1.3	DOC10356–1LA		
6350 Handbook	21.0	UPD10161–11A	20.2.3	DOC10161–1LA
6550 Handbook (Including the 6350, 6150 6650, and 6450)	22.1.1	RLN10161–22A	22.0 21.0.2 21.0.3	UPD10161–21A RLN10161–21A DOC10161–2LA
Prime 9955 Handbook	22.0	UPD8887–23A	21.0 20.0.2 20.0	UPD8887–22A UPD8887–21A DOC8887–2LA
Using Your Prime 2250	21.0	UPD6516–11A	22.0 19.1	RLN10239–001 DOC6516–191L
Using Your Prime 2450 (Including the 2350)	20.0	DOC10085–1LA	22.0	RLN10239–001
Using Your 2455	21.0.1	UPD10085–21A	22.0 21.0	RLN10239–001 DOC10085–2LA
Using Your Prime 2550	19.4.3	UPD8552–11A	22.0 19.2	RLN10239–001 DOC8552–192L
Using Your Prime 2655	20.0	DOC8552–2LA	22.0	RLN10239–001
Using Your 2755	21.0	DOC8552–3LA	22.0	RLN10239–001

Table C–1. PRIMOS Administration and Operation				
Using Your 4150	21.0.2	DOC10163–1LA	22.0	RLN10239–001
Release Note for 6250–bpi Tape Dump	22.1	RLN12005–1XA		

Table C–2. PRIMOS Architecture and Assembly				
Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
Assembly Language Programmer's Guide	T2.0–22.1	DOC3059–3LA		
50 Series Technical Summary	21.0	DOC6904–2LA		
Instruction Sets Guide	21.0	DOC9474–2LA		
System Architecture Reference Guide	21.0	DOC9473–2LA		

Table C–3. PRIMOS Use and Programming				
Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
Introduction to PRIMOS	20.2	DOC10111–1XA		
PRIMOS User's Guide	22.0	DOC4130–5LA	23.0	DOC10316–1PA
PRIMOS User's Release Document	23.0	DOC10316–1PA		
PRIMOS Commands Reference Guide	22.1	RLN3108–71A	23.0 22.0	DOC10316–1PA DOC3108–7LA
New PRIMOS Help	23.0	DOC20012–1PA		
CPL Programmer's Companion	19.3	FDR7811–193		
CPL User's Guide	21.0	DOC4302–3LA		

Table C–3. PRIMOS Use and Programming (continued)				
Programmer's Guide to BIND and EPFs	23.0	UPD8691–12A	22.0 19.4	UPD8691–11A DOC8691–1LA
Security Features User's Guide	22.0	UPD10130–11A	21.0	DOC10130–1LA
SEG and LOAD Reference Guide	18.1 to 19.2	DOC3524–192L		
Source Level Debugger User's Guide	20.2	UPD4033–22A	19.4 19.2	UPD4033–21A DOC4033–193L
Subroutines Reference I: Using Subroutines	23.0	UPD10080–21A	22..1 21.0	RLN10247–1LA DOC10080–2LA
Subroutines Reference II: File System	23.0	DOC10081–2LA		
Subroutines Reference III: Operating System	23.0	DOC10082–2LA		
Subroutines Reference IV: Libraries and I/O	23.0	DOC10083–2LA		
Subroutines Reference V: Event Synchronization	23.0	UPD10213–11A	22.1 22.0	RLN10247–1LA DOC10213–1LA
Advanced Programmer's Guide I: BIND and EPFs	T3.0–23.0	DOC10055–2LA		
Advanced Programmer's Guide II: File System	23.0	DOC10056–3LA		
Advanced Programmer's Guide III: Command Environment	T3.0–23.0	DOC10057–2LA		
Advanced Programmer's Guide: Appendices and Master Index	T3.0–23.0	DOC10066–4LA		

Table C–4. Communications

Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
Distributed Processing Terminal Executive (DPTX) Guide	22.0	DOC4035–4LA		
ICS User's Guide	21.0	UPD10094–11A	20.1	DOC10094–1LA
LTS300 Installation Guide	21.0	DOC11034–2LA		
NTS Planning and Configuration Guide	23.1	DOC10159–2LA		
NTS Quick Reference Card	22.0	IDR10218–1XA		
NTS User's Guide	23.0	DOC10117–3LA		
PRIME/SNA Application Program Interface Programmer's Guide	21.0.6	DOC10100–2LA		
PRIME/SNA Concepts and Overview	21.0.5	DOC11095–1LA		
PRIME/SNA Installation and Configuration Guide	21.0.5	DOC11097–1LA		
PRIME/SNA Interactive Terminal User's Guide	21.0.5	DOC8910–2LA		
PRIME/SNA LU 6.2 API Programmer's Guide	21.0.5	DOC11101–1LA		
PRIME/SNA Operator's Guide	21.0.5	DOC8909–5LA		
PRIME/SNA Resource Planning Guide	21.0.5	UPD11096–11A	21.0.5	DOC11096–1LA
PT200 Keyboard PRIME/SNA Interactive Reference Card	21.0	IDR8910–2RA		
PRIMOS TCP/IP Guide	22.0	DOC10155–3LA		
Rev. 23.0 Prime Networks Release Notes	23.0	RLN10252–1LA		
Operator's Guide to Prime Networks	22.0	UPD10114–11A	21.0 23.0	DOC10114–1LA RLN10252–1LA

Table C–4. Communications				
Programmer's Guide to Prime Networks	22.0	UPD10113–11A	21.0 23.0	DOC10113–1LA RLN10252–1LA
PRIMENET Planning and Configuration Guide	22.1	UPD7532–41A	22.0 23.0	DOC7532–4LA RLN10252–1LA
Remote Job Entry Phase II Guide	22.0	UPD6053–42A	21.0 21.0	UPD6053–41A DOC6053–4LA
User's Guide to Prime Network Services	22.1	UPD10115–11A	21.0 23.0	DOC10115–1LA RLN10252–1LA
WSI300 User's Guide	21.0	DOC10155–2LA		

Table C–5. Data Management				
Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
DBMS Administrator's Guide	22.0	DOC6292–2LA		
DBMS Data Description Language Reference Guide	19.4	UPD5717–11A	18.1	DOC5717–181L
DBMS Data Manipulation Language Reference Guide	21.0	UPD5308–12A	19.4 19.0	UPD5308–11A DOC5308–190L
DBMS Master Index	22.0	DOC10164–3PA		
DBMS Programmer's Companion	19.4	DOC8645–1XA		
DBMS User's Guide	20.2	UPD6291–11A	19.2	DOC6291–192P
DBMS/QUERY Report Generator Casebook	18.2	IDR5650–182P		
DISCOVER Reference Guide	22.0	DOC7798–2LA		
DISCOVER User's Guide	20.0	UPD7799–11A	19.4	DOC7799–1PA
MIDASPLUS Concepts	20.2	DOC9243–1PA		
MIDASPLUS Companion	19.4	DOC10045–1XA		
MIDASPLUS User's Guide	22.0	DOC9244–2LA		

Table C–5. Data Management				
PRIFORMA Forms Design and Administraton Guide	22.0	DOC10240–1LA		
PRIFORMA Forms Manipulation Language Guide	22.0	DOC10241–1LA		
PRIFORMA Programmer's Guide	22.0	DOC10242–1LA		
Prime ORACLE 5.1 Administrator's Manuals	22.0	DCP10078		
ORACLE VERSION 6.0 Installation and User's Guide for 50 Series Systems	v6.0.27	UPD10134–31A	22.0	DOC10134–3LA
PRISAM Programmer's Companion	21.0	DOC10088–1XA		
PRISAM User's Guide	22.0	DOC7999–4LA		
ROAM Administrator's Guide	22.0	UPD7345–31A	21.0	DOC7345–3LA

Table C–6. Languages				
Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
Translator Family Software Release Document	T3.0–23.0	DOC10217–3PA		
BASIC/VM Programmer's Guide	19.4	UPD3058–33A	19.0 18.1 17.2	COR3058–002 COR3058–001 FDR3058–101B
C User's Guide	T3.0–23.0	DOC7534–4LA		
C++ User's Guide	T2.2–22.1	DOC20004–1LA		
COBOL 74 Reference Guide	T1.0–21.0	UPD5039–23A	21.0 20.2 20.0	UPD5039–22A UPD5039–21A DOC5039–2LA
COBOL85 Reference Guide	1.1–22.0	RLN10166–11A	22.0	DOC10166–1LA

Table C–6. Languages (continued)

CBL to COBOL85 Conversion Program Guide	22.0	DOC10276–1PA		
FORTRAN Reference Guide	21.0	UPD3057–34A	19.4 19.0 18.1 17.2	UPD3057–33A COR3057–002 COR3057–001 FDR3057–101B
FORTRAN 77 Reference Guide	T2.0–22.1	UPD4029–51A	T1.0–21.0	DOC4029–5LA
Pascal Programmer's Companion	21.0	DOC7095–2XA		
Pascal Reference Guide	21.0	DOC4303–4LA		
PL/I Conversion Guide: IBM PL/I to Prime PL/I	19.4	DOC5769–1LA		
PL/I Reference Guide	TI.0–21.0	UPD5041–11A	19.4	DOC5041–1LA
PL/I Subset G Reference Guide	19.4	UPD4031–13A	19.0 18.1 17.2	PTU2600–084 PTU2600–075 IDR4031
PRIME Common LISP Environment Reference Manual	20.2	MAN10120–1LA		
PRIME Common LISP Language Reference Manual	20.2	MAN10119–1LA		
Prime RPG II Debugging Template	20.0	IDR11001–2XA		
RPG II V–mode Compiler Reference Guide	T1.0–21.0	UPD5040–22A	21.0 20.0	UPD5040–21A DOC5040–2LA

Table C–7. Editors				
Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
EMACS Extension Writing Guide	19.4	DOC5025–2LA		
EMACS Primer	18.3	IDR6107		
EMACS Quick Reference Card	21.0	IDR5026–1RA		
EMACS Reference Guide	21.0	DOC5026–2LA		
EMACS Standard User Interface Guide	19.4.5	DOC7446–2LA		
New User's Guide to EDITOR and RUNOFF	19.0	FDR3104–101B		

Index

Symbols

! character, before commands in FS_RECOVER, 4-14
@ character
  WATCH monitoring, 2-11
  wildcard, 2-14, 2-18

A

Access categories, for sharing memory, 2-20
ACL groups
  listing user's, 2-13
  .RAS$, 3-10
  validating IDs, 2-17
  .WATCH$, 2-10
ACL protection
  displaying at prompt, 2-6
  FS_RECOVER, 4-9
  SYSTEM_DEBUG*, 4-9
ACL-protected shared memory, 2-20
ADDISK messages, C-1
ASYNC –MAXIMUM_SHUTDOWN_TIME command, 3-10
Async lines, disallowing login if no carrier, 3-13
Attach points
  displaying at prompt, 2-5
  establishing multiple, 2-18
  returned by function, 2-13
ATTACH_POINT function, 2-13
ATTRIB function, 2-23
AUTOPSY
  run file, 4-9
  search rules file, 4-9

B

Backup and recovery subsystem, 3-13
BATCH –START –DISPLAY_ALL command, 3-6
Batch subsystem, 3-6
  BATCH –START –DISPLAY_ALL, 3-6
  INIT –SEM_INIT, 3-6
  reinitializing semaphores, 3-6
  restarting, 3-6
  viewing job queues, 3-6
BATCH_SERVICE, 3-6

C

C compiler
  opening files, 2-23
  passing array pointers, 2-23
CAM files
  following system crash, 4-8
  truncating, 2-24
Carriage return
  in prompts, 2-6
  redisplaying login prompt, 3-2
  when spooler starts, 3-10
Carrier signal, 3-13
CBL compiler, 32IX mode, 2-23
CF$EXT subroutine, 2-24
Character strings
  collating sequences, 2-25
  maximum line length, 2-25
  reversing, 2-17
  validating numeric, 2-17
CNAME –REPORT command, 2-1
COBOL. *See* CBL compiler
Coldstart
  disks inactive since, 2-14
  during PRIMOS installation, 1-1
  following crash, 4-1
  pausing to invoke FS_RECOVER, 4-10
  re-enabling WATCH access, 2-9
  required after RFS, 4-3
  setting log in prompt, 3-3
  time since, 2-15
COMDEV
  damaged by crash, 4-8
  SYSTEM_DEBUG*, 4-8
Command functions. *See* Functions
Command line
  functions, 2-12
  prompts, 2-4
  within FS_RECOVER, 4-14
Commands
  BATCH –START –DISPLAY_ALL, 3-6
  CNAME –REPORT, 2-1
  CREATE –REPORT, 2-1
  exclamation point character, 4-14
  executing within FS_RECOVER, 4-14
  INDEX_LIB_MANAGER, 3-13
  INIT –SEM_INIT, 3-6
  INIT_RECOVER –PAUSE, 4-10
  LD –RWLOCK, 2-2
  LD –TOTAL, 2-4
  MAKE –REPORT, 3-6
  RDY –EXPAND_PROMPT, 2-4
  RDY –NO_EXPAND_PROMPT, 2-4
  RECORD_TO_PATH, 3-10
  SEARCH_INDEX_LIB, 3-15
  SET_ASYNC –DCD_LOGIN, 3-13
  SET_ASYNC –NO_DCD_LOGIN, 3-13
  SHOW, 2-8
  START_LSR, 3-1
  STATUS SYSTEM, 4-11
  TALK, 2-8
  USAGE –SYSTEM, 3-12
  WATCH, 2-8
COMO –NTTY
  returning current output state, 2-13

WATCH session, 2-12
COMO files
crash recovery session, 4-17
duplicating file unit info, 2-18
FIX_DISK, 4-7
FS_RECOVER, 4-16, 4-17
returning current state, 2-13
WATCH session, 2-10
COMO_INFO function, 2-13
Configuring
DTAR 2 memory sharing, 2-22
log in prompt, 3-1
multiple tape controllers, 3-17
number of user segments, 3-16
segments for FS_RECOVER, 4-9
Control codes
during FS_RECOVER, 4-13
WATCH monitoring, 2-12
CPL
for automated FIX_DISK, 4-16
for running FIX_DISK, 4-7
functions, 2-12, 2-23
CPL files
FIX_DISK_MONITOR.CPL, 4-16
FS_RECOVER.INSTALL.CPL, 4-8
INIT_RECOVER.CPL, 4-10
RUN_FIX_DISK.CPL, 4-16
CPU, returning processor type, 2-14
CPU time
displaying at prompt, 2-5
since boot, 2-15
since login, 2-16
system meters, 3-12
Crash dumps
6250 bpi tape, A-2
allocating records for, 4-10
analyzing, 4-16
calculating record requirements, 4-11
checking tape drive, 4-15
creating files, 4-15
disk space for analysis, 4-11
file pathname, 4-17
full, 4-11
generating, A-1
initiating partial dump, A-2
labeling tapes, A-3
multi-reel, 4-16, A-2
partial, 4-11, A-2
performing, 4-1, 4-12
reading tapes, 4-14, 4-16
size of, A-2
tape reels, A-2
Crash recovery tools
FS_RECOVER, 4-4
RECORD_TO_PATH, 3-10
RFS, 4-2
CREATE –REPORT command, 2-1

D

Data integrity, following system crash, 4-3
Date
displaying at log in, 3-1
displaying at prompt, 2-5
validating, 2-17
DCD carrier signal, 3-13
Despooler
phantom, 3-10
setting shutdown interval, 3-10
setting sleep period, 3-9
Directories
creating, 2-1
displaying ACL protection, 2-6
FS_RECOVER, 4-8
FS_RECOVER working directory, 4-16
listing contents, 2-2
records used, 2-4
SYSTEM_DEBUG*, 4-9, 4-10, 4-16
Disk controllers, returning number of, 2-14
Disk partitions
bad record, 3-11
error messages, C-1
formatting, 3-6
initial state, 4-2
mounting, C-1
recovering, 4-1
requiring FIX_DISK, 4-2
returning number of, 2-14
shutting down, 4-2
space for crash analysis, 4-11
DISK_MANAGER, 4-23
Documentation
list, D-1
NTS, 3-18
online, ix
Rev. 23.1, ix
DRB configuration file, 3-14
DSM subsystem, documentation corrections, 3-18
DTAR 1, segment sharing, 2-20
DTAR 2, segment sharing, 2-20
Dynamic segments, FS_RECOVER requirements, 4-9

E

ECL
effects on prompts, 2-5
using within FS_RECOVER, 4-14
WATCH monitoring, 2-12
ED editor, 1-1
Edit Command Line. *See* ECL
EDIT_PROFILE, duplicating user profile, 3-17
EMACS
during installation, 1-1
WATCH monitoring, 2-12
EPF$INVK subroutine, 2-24
EPF$MAP subroutine, 2-24
EPF$REG subroutine, 2-24
Error codes, 2-26
displaying at prompt, 2-5
Ethernet controller, 3-12
Extended index libraries, 3-13

F

File system integrity, 4-1
File units
duplicating, 2-18
file being printed, 3-10
Filenames
changing, 2-1
determining from record address, 3-10
suffixes, 2-17
validating, 2-17
Files
affected by a crash, 4-8
being printed, 3-10
crash dump, 4-17
current COMO file, 2-13
displaying locks, 2-2
duplicating file unit info, 2-18
for crash dump, 4-15
records used, 2-4
reporting name change, 2-1

search rules, 4-9
FIX_DISK
automated, 4-21
–COMDEV option, 4-8
COMO files, 4-7
deferred, 4-18
determining if required, 4-2
during PRIMOS install, 1-2
FS_RECOVER recommendation examples, 4-19
immediate, 4-18
manager phantom, 4-22
monitor, 4-16
monitor phantom, 4-7
not required, 4-19
recommendations for running, 4-18
FIXBAT, 3-6
Forced shutdown, procedure for, 4-12
Form feed. *See* Carriage return
FS_RECOVER, 4-4
ACL requirements, 4-9
breaking out of, 4-13
changing PRIMOS.COMI, 4-10
checking tape drive, 4-15
COMO file for session, 4-17
COMO files, 4-16
control-P during, 4-13
crash dump file, 4-17
crash recovery analysis, 4-16
creating crash dump files, 4-15
data analysis messages, 4-18
deferred FIX_DISK, 4-18
disk manager, 4-23
ECL environment, 4-14
error messages, 4-12, B-2
example of crash analysis, 4-17
executing PRIMOS commands during, 4-14
FIX_DISK not required, 4-19
FIX_DISK recommendations, 4-18
following forced shutdown, 4-12
immediate FIX_DISK, 4-18
installation CPL file, 4-8
installation errors, 4-12, B-1
installation of, 4-8
installation tape, 4-8
invocation messages, B-1
invoking, 4-12
invoking at coldstart, 4-10
machine state, 4-17
main menu, 4-13
messages, B-1
options, 4-13
phantoms, 4-8
reading crash dump tapes, 4-14, 4-16
runtime warning messages, B-3
security, 4-9, 4-12
stopping, 4-13
using, 4-12
version numbers, 1-2
working directory, 4-16
Functions
ATTACH_POINT, 2-13
COMO_INFO, 2-13
displaying at prompt, 2-6
GROUP_LIST, 2-13
REVERSE, 2-17
SYSTEM_INFO, 2-14
SYSTEM_USAGE, 2-14
USER_INFO, 2-15
USER_USAGE, 2-16
VALIDATE, 2-17

G

Global variables, displaying at prompt, 2-6
GROUP_LIST function, 2-13
Groups. *See* ACL groups

H

Half-duplex, WATCH sessions, 2-12
Halts and hangs. *See* System crashes
Hexadecimal numbers, validating, 2-18

I

ICE command, freeing shared segments, 2-22
Index files, 3-13
INDEX_LIB_MANAGER command, 3-13
INIT –SEM_INIT command, 3-6
INIT_RECOVER.CPL, 4-10
Installation
FS_RECOVER, 1-2, 4-4, 4-8
FS_RECOVER size, 4-8
FS_RECOVER tape format, 4-8
PRIMOS, 1-1
PRIMOS microcode, 1-2
I/O buffers, monitoring, 2-10
I/O operations, metering, 3-12
I/O time
displaying at prompt, 2-5, 2-6
since boot, 2-15
since login, 2-16

L

LD command
–RWLOCK option, 2-2
–TOTAL option, 2-4
header display, 2-4
LHC300 controller, 3-12
Locate buffers
flushing, 4-3
metering, 3-12
Log book, 4-17
Log in
configuring prompt, 3-1
disallowing if no carrier signal, 3-13
time of, 2-16
LOGIN_SERVER, 3-1
setting defaults, 3-3

M

Magnetic tapes. *See* Tapes
MAGSAV/RST subsystem, 3-13
Maintenance processor, entering, 4-2
MAKE –REPORT command, 3-6
Manuals. *See* Documentation
Memory
amount configured, 2-14
determining size of, 4-11
pages in use by process, 2-16
pages in use by system, 2-15
sharing, 2-20
wired pages, 2-15, 2-16
MESSAGE command, effects on TALK, 2-8
Microcode requirements, 1-2
MMSSHARE_APM subroutine, 2-21
error codes, 2-26
Monitoring user I/O, 2-8
MPC4000 command, 3-8

MRG1$S subroutine, 2-25
MTRS configuration directive, 3-17
  effects on T$MT, 2-25
Multiple attach points, 2-18

N

Network Terminal Service. *See* NTS users
Non-printable characters, WATCH monitoring, 2-12
NSED editor, 1-1
NSEG configuration directive, 3-16
NTS users
  CPU time used, 3-12
  documentation, 3-18
  monitoring, 2-8
Numbers, validating, 2-18

O

Octal numbers, validating, 2-18

P

Paging I/O, 3-12
Passwords
  specifying in EDIT_PROFILE, 3-17
  validating, 2-17
  WATCH sessions, 2-12
Pathnames, determining from record address, 3-10
Performance considerations, changing attach points, 2-18
Phantoms
  despooler, 3-10
  FIX_DISK manager, 4-22
  FIX_DISK monitor, 4-7
  for automated FIX_DISK, 4-7, 4-22
  for FS_RECOVER, 4-8
  identifying process as, 2-15
  monitoring, 2-8
PIO/S system meter, 3-12
PRIMOS revision
  displaying at log in, 3-1
  displaying at prompt, 2-6
  documentation, ix
  documentation for, 3-18, D-1
  FS_RECOVER support, 1-2, 4-4
  installing, 1-1
  list of new features, 1-3
  microcode requirements, 1-2
  product retirements, 1-3
  returning to function, 2-14
  translator family, 1-1
PRIMOS.COMI
  changing for FS_RECOVER, 4-10
  INIT_RECOVER –PAUSE, 4-10
  pausing, 4-10
Print buffer, 3-10
Print controllers, MPC4000, 3-8
Printers
  Centronics, 3-8
  Dataproducts, 3-8
  double-sided laser printers, 3-10
  graphic, 3-9
  PostScript, 3-8
  setting speed, 3-9
Process ID, returning, 2-15
Processes
  idle, 2-15
  suspended, 2-15
  user information, 2-15
Project ID
  returning, 2-15
  TALK server, 2-8
  validating, 2-17
Prompts
  carriage return, 2-6
  error prompts, 2-4
  log in, 3-1
  ready prompt, 2-4
  redisplaying log in prompt, 3-2
  variables, 2-5

R

.RAS$ ACL group, 3-10
RDLIN$ subroutine, 2-25
RDY command
  –EP option, 2-4
  –EXPAND_PROMPT option, 2-4
  –NEP option, 2-4
  –NO_EXPAND_PROMPT option, 2-4
  error messages, 2-7
Read and write locks
  C open statement, 2-23
  displaying, 2-2
RECORD_TO_PATH command, 3-10
  error codes, 2-27
Records
  converting addresses, 3-10
  displaying number used, 2-4
  FS_RECOVER requirements, 4-8
  maximum tape record size, 3-17
  required for full crash dump, 4-11
  required for partial crash dump, 4-11
Registered EPFs
  EPF$REG subroutine, 2-24
  insufficient segments, 2-26
  registration levels, 2-26
Remote files, maximum line length, 2-25
Remote system name, displaying at prompt, 2-6
Remote users
  monitoring, 2-8
  user information, 2-15
  WATCH sessions, 2-12
Resident Forced Shutdown. *See* RFS
REVERSE function, 2-17
RFS, 4-2
  console messages, 4-3
  initial disk state, 4-2
  invoking, 4-2

S

S$START_COPY print handler key, 3-10
Scheduler, returning priority, 2-15
Search rules
  AUTOPSY, 4-9
  changes for FS_RECOVER, 4-9
  COMMAND$, 4-9
  ENTRY$, 1-2, 4-9
  MAPS, 4-9
SEARCH_INDEX_LIB command, 3-15
Segment directories, displaying locks, 2-2
Segments
  assigning for sharing, 2-22
  configuring, 3-16
  FS_RECOVER requirements, 4-9
  in use by process, 2-16
  maximum number of, 3-16
  number in use, 2-15
  sharing, 2-20
  VMFA, 2-22
Semaphores, reinitializing for batch, 3-6
SEMFIL, 3-6

Servers, user information, 2-15
SET_ASYNC command
  –DCD_LOGIN option, 3-13
  –NO_DCD_LOGIN option, 3-13
SET_LSR_DEFAULTS, 3-3
SETU$S subroutine, 2-25
SHOW command, 2-8
  syntax, 2-9
Software
  See also PRIMOS revision documentation, D-1
  FS_RECOVER as IPR, 1-2
  independent products, 1-1
  retired, 1-3
  translator family release, 1-1
Spooler subsystem, 3-8
  ASYNC –MAXIMUM_SHUTDOWN_TIME command, 3-10
  data file unit number, 3-10
  documentation corrections, 3-21
  HEADER –NO_INITIAL_FF command, 3-10
  MPC4000 controller, 3-8
SRTF$S subroutine, 2-25
START_LSR command, 3-1
Static mode execution, displaying at prompt, 2-6
Strings. *See* Character strings
Subroutines (new)
  MM$SHARE_APM, 2-21
  UT$COPY, 2-18
System Administrator
  disabling WATCH monitoring, 2-9
  maintaining log book, 4-17
  RECORD_TO_PATH access, 3-10
  segment requirements, 4-9
  SET_ASYNC –DCD_LOGIN access, 3-13
  WATCH access permission, 2-10
System crashes
  analyzing data integrity, 4-18
  determining machine state, 4-17
  effects on shared memory, 2-22
  forced shutdown, 4-12
  halting the machine, A-2
  recovery recommendations, 4-12
  recovery tools, 4-1
System metering information, 3-12
System name
  displaying at log in, 3-1
  displaying at prompt, 2-6, 3-1
  returning to function, 2-14
SYSTEM_DEBUG*
  ACL requirements, 4-9
  CRASH, 4-9, 4-10, 4-15, 4-16
  in COMMANDS search rules, 4-9
  INIT_RECOVER.CPL, 4-10
  installing FS_RECOVER, 4-8
SYSTEM_INFO function, 2-14
SYSTEM_USAGE function, 2-14

T

T$MT subroutine, 2-25
TALK facility, 2-8
Tape controllers, configuring record size, 3-17
Tape drives
  assigning for FS_RECOVER, 4-15
  crash dump to 6250 bpi, A-2
  error messages, 4-15
  model numbers, 2-25
Tapes
  buffers, 2-25
  FS_RECOVER installation, 4-8
  generating crash dump, A-1
  labeling crash dumps, A-3
  maximum record size, 3-17
  multi-reel crash dump, 4-16, A-2
  T$MT subroutine, 2-25
  using crash dump, 4-5, 4-14
Terminal display, during WATCH session, 2-12
Terminal monitoring, 2-8
Text strings. *See* Character strings
Time
  displaying at login, 3-1
  displaying at prompt, 2-5
  log in, 2-16
Translator family release, 1-1

U

USAGE –SYSTEM command, 3-12
User ID
  displaying at prompt, 2-6
  returning, 2-15
  validating, 2-17
User number
  displaying at prompt, 2-6
  returning, 2-15
USER_INFO function, 2-15
USER_USAGE function, 2-16
Users
  disabling WATCH access, 2-10
  displaying at prompt, 2-6
  displaying WATCH monitoring, 2-11
  monitoring, 2-8
  multiple logins and WATCH, 2-11
  returning number logged in, 2-14
  returning type, 2-15
  SYSTEM, 4-9
  viewing batch queues, 3-6
UT$COPY subroutine, 2-18
  error codes, 2-26

V

VALIDATE function, 2-17
VMFA segments, 2-22

W

WATCH command, 2-8
  error codes, 2-26
  multiple sessions, 2-12
  syntax, 2-11
WATCH$ ACL group, 2-10
  disabling, 2-9
WILD function, 2-23
Wildcards
  listing groups, 2-14
  validating strings, 2-18
WTLIN$ subroutine, 2-25