CVSI, Inc.
4G Crosby Drive
Bedford, MA 01730

Rev. 23.3 Software Release Document

Revision 23.3

DOC13134-1PA

Rev. 23.3 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 23.3 (Rev. 23.3).

Prime Computer, Inc., Prime Park, Framingham, Massachusetts 01701

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 © 1992 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, 5370, 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.

HP LaserJet is a registered trademark of Hewlett-Packard Company.

IBM is a registered trademark of International Business Machines Corp.

PostScript is a registered trademark of Adobe Systems, Inc.

SyncSort is a registered trademark of Syncsort, Inc.

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

Printing History

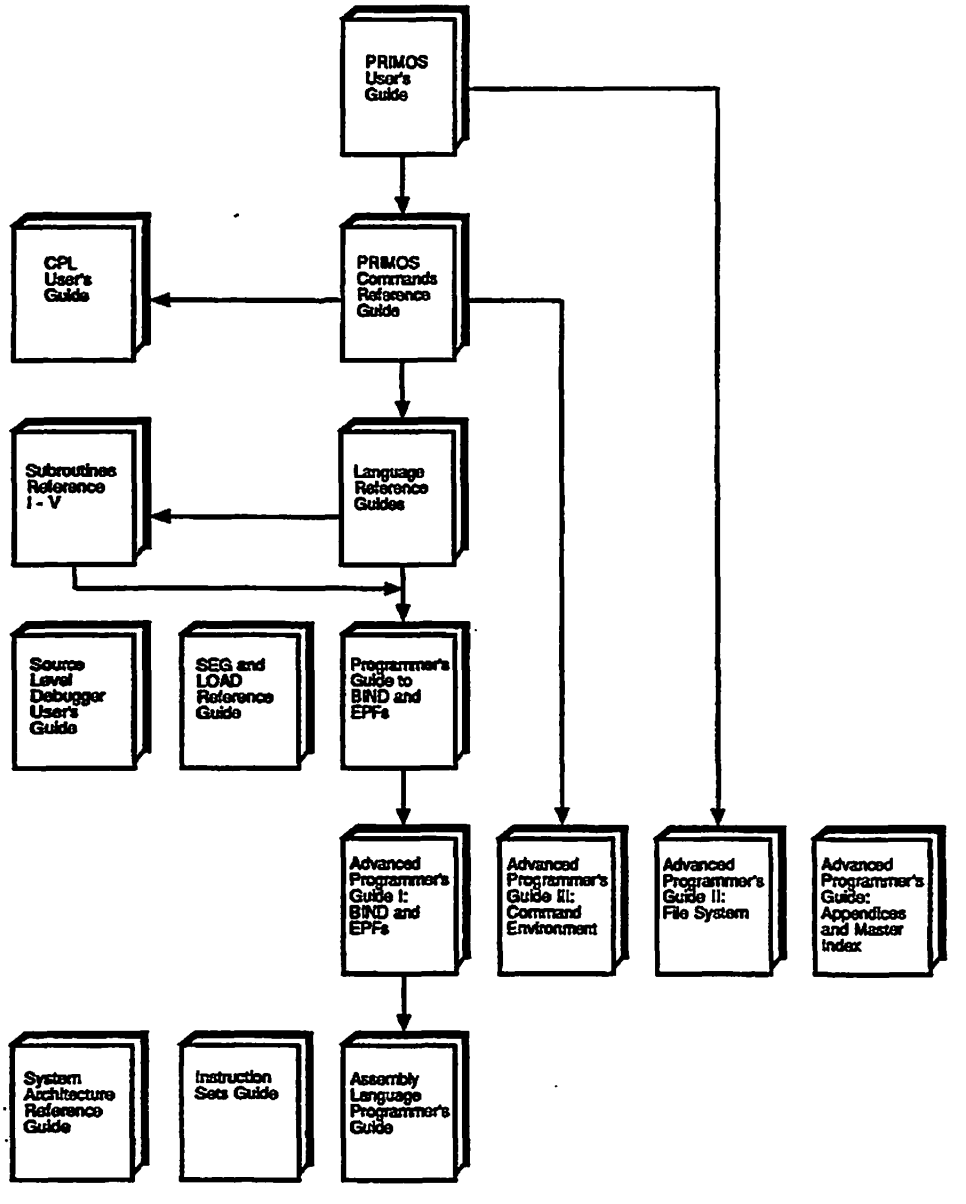
First Edition (DOC10001–1PA) April 1985 for Revision 19.4
Second Edition (DOC10001–2PA) January 1986 for Revision 20.0
Third Edition (DOC10001–3PA) August 1986 for Revision 20.2
Fourth Edition (DOC10001–4PA) July 1987 for Revision 21.0
Fifth Edition (DOC10001–5PA) October 1988 for Revision 22.0
Sixth 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
Ninth Edition (DOC10001–9PA) August 1991 for Revision 23.2

First Edition (DOC13134–1PA) May 1992 for Revision 23.3

Credits

Editorial: Mary Skousgaard
Project Support: Douglas Gilbert, Barbara Bailey, George Gove
Technical Support: Joe Coelho, Scott Gorton, Andy Griffith, Robert Love, Leo Sager
Production: Judy Gordon

Reading Path for PRIMOS Documentation



Contents

About This Book

1 Introduction

Special Considerations for Rev. 23.3 . . . 1-1
- Hardware Requirements . . . 1-1
- Microcode Requirements . . . 1-1
- Installation . . . 1-2

Product Retirements . . . 1-4

Rev. 23.3 New Features . . . 1-5
- User and Programmer Features . . . 1-5
- Operator and Administrator Features . . . 1-6

2 New Features for the User and Programmer at Rev. 23.3

Logging In . . . 2-1
- Implicit Login . . . 2-1
- No Logins Permitted . . . 2-1

Online HELP Enhancements . . . 2-2

LD Command Enhancements . . . 2-2

LIST_USERS Command . . . 2-4
- Selection Options . . . 2-4
- Display Options . . . 2-5
- Examples . . . 2-6

MESSAGE Command Enhancements . . . 2-7
- Maximum Message Line Length . . . 2-8
- Multiline Messages . . . 2-8
- MESSAGE –DISPLAY Command Option . . . 2-9
- MSG$ST and SMSG$ Subroutine Support . . . 2-10
- Notification of Unsent Messages . . . 2-10

TERM Command Enhancement . . . 2-10

IF Command Function . . . 2-11
- Examples . . . 2-12

Other Function Enhancements . . . 2-13
- [ATTRIB –TRUNC] . . . 2-13

[WILD –TRUNC] . . . 2-14
DIRLS, DIRRD, and DIR$SE Subroutines . . . 2-14
PRIMENET X.3 Support . . . 2-15
Per–user Profiles . . . 2-16
X.3 Parameters . . . 2-16
Documentation Corrections . . . 2-18
Instruction Sets Guide . . . 2-18
SEG and LOAD Reference Guide . . . 2-19
Source Level Debugger User's Guide . . . 2-19
Programmer's Guide to Prime Networks . . . 2-19
MAGNET User's Guide . . . 2-19
Subroutines Reference IV: Libraries and I/O . . . 2-20
Subroutines Reference V: Event Synchronization . . . 2-21
PRIMOS Error Codes . . . 2-22

3 New Features for the Operator and Administrator at Rev. 23.3

Model 5520 and 5540 Computers . . . 3-1
CPUID$ Subroutine . . . 3-1
USAGE –MULTI Command Option . . . 3-2
5500 Family System Limits . . . 3-2
G$METR Subroutine . . . 3-3
System Limits . . . 3-3
Redundant Power Supply . . . 3-4
Model 4601 Cartridge Tape Drive . . . 3-4
ASSIGN Command for the Model 4601 Tape Drive . . . 3-4
T$MT Subroutine Support for the Model 4601 Tape Drive . . . 3-5
Error Messages . . . 3-6
System Boot Procedure . . . 3-7
Login Enhancements . . . 3-7
START_LSR Command . . . 3-7
SET_LSR_DEFAULTS Command . . . 3-8
MAXUSR Command . . . 3-8
Logging LOGINs: a CONFIG_USERS Enhancement . . . 3-9
User ID Security Enhancement . . . 3-12
Logout Enhancements . . . 3-12
LOGOUT Command Enhancements . . . 3-12
Forced Logout . . . 3-15
LOUTQM Configuration Directive Enhancement . . . 3-15
User Support Enhancements . . . 3-16
HELP Facility Terminal Types Support . . . 3-16
Defining a Terminal Type for HELP . . . 3-16
INITIALIZE_SEARCH_RULES Command . . . 3-17
MESSAGE Command . . . 3-18
File System Support Enhancements . . . 3-20

CNAME –FORCE Command . . . 3-20
Batch Subsystem Enhancements . . . 3-21
JOB –STATUS and JOB –DISPLAY . . . 3-21
Batch Queue Full . . . 3-22
Spooler Subsystem Enhancements . . . 3-22
Support for HP LaserJet Printers . . . 3-22
PostScript Formatting Defaults . . . 3-24
Aliases for Mandatory Printer Environment Attributes . . . 3-24
Printer Error Notification . . . 3-25
User Access to Printer Control . . . 3-25
Size of Print Job . . . 3-25
Printing Over TCP/IP . . . 3-26
Restarting the Spooler Subsystem . . . 3-26
Listing Spooler Processes . . . 3-27
User-written External Login Programs . . . 3-27
CDD –INFO Command Options . . . 3-28
CDD –INFO . . . 3-29
CDD –INFO –MEMORY_SIZE . . . 3-32
CDD –INFO –DUMP_SIZE . . . 3-33
CDD –INFO –DUMP_SIZE_TABLE . . . 3-34
MAKE and FIX_DISK Command Enhancements . . . 3-37
MAKE Command . . . 3-37
FIX_DISK Command . . . 3-40
FS_RECOVER Enhancements . . . 3-41
Other Disk Support Enhancements . . . 3-42
PDEV Command . . . 3-42
SPIN_UP Command . . . 3-45
STATUS DEVICES Command Option . . . 3-46
SHUTDN Command . . . 3-47
ICS3 Communications Controller Enhancements . . . 3-48
Documentation Corrections . . . 3-48
DSM User's Guide . . . 3-48
Prime INFORMATION RECOVERY Reference Guide . . . 3-48

4 CONFIG_USERS Terminal Interface Definition

Terminal Administration . . . 4-1
The SMVARS File . . . 4-1
Location of the SMVARS File . . . 4-2
Location of the Keys and Video Files . . . 4-2
Runtime Access to the Keys and Video Files . . . 4-3
Terminal Administration Summary . . . 4-3
MODIFY_FORMKEY and the Keys File . . . 4-4
Keys File Sequences . . . 4-4
Invoking MODIFY_FORMKEY . . . 4-4

Control Keys and Data Keys . . . 4-5
The Entry Screen . . . 4-6
Main Menu . . . 4-7
Exiting the Utility . . . 4-8
Help Screen . . . 4-10
Defining Cursor Control and Editing Keys . . . 4-12
Assigning a Key to a Function . . . 4-12
Defining Function Keys . . . 4-13
Defining Miscellaneous Keys . . . 4-14
Test Keys File . . . 4-17
Format of the Keys File . . . 4-19
Use of Key Descriptions . . . 4-21
Compiling the Keys File . . . 4-21
The Video File . . . 4-22
Format of the Video File . . . 4-22
Video File Commands . . . 4-23
Keywords Taking Parameters . . . 4-23
Summary of Percent Commands . . . 4-24
Output commands . . . 4-24
Stack manipulation commands . . . 4-24
Parameter sequencing commands . . . 4-25
Parameter changing commands . . . 4-25
Control commands . . . 4-25
The Stack – Internal Processing of Parameters . . . 4-25
Examples of Stack Manipulation and Arithmetic Commands . . . 4-26
Examples of Parameter Sequencing Commands . . . 4-27
Examples of Output Commands . . . 4-27
Examples of Parameter Changing Commands . . . 4-27
Examples of Control Commands . . . 4-28
Video File Keywords . . . 4-30
Keyword Summary . . . 4-30
Basic Capabilities . . . 4-30
Erasure Commands . . . 4-30
Cursor Control . . . 4-30
Cursor Position . . . 4-31
Latch Attributes . . . 4-31
Area Attributes . . . 4-31
Message Line . . . 4-31
Borders . . . 4-31
Shifting Field Indicators . . . 4-32
Block Mode Routine Entry Points . . . 4-32
Basic Capabilities . . . 4-32
Attribute Keywords . . . 4-35
LATCHATT . . . 4-36
AREAATT . . . 4-36
SGR . . . 4-37

ASGR . . . 4-38
ARGR . . . 4-38
Use of Color . . . 4-39
Message Line Keywords . . . 4-40
Form Borders . . . 4-40
Shifting Field Indicators . . . 4-42
DRAW_FORM Status Text . . . 4-42
Cursor Position Display . . . 4-43
Compiling a Video File . . . 4-44

Appendices

A ***ASCII Control Codes***

B ***Rev. 23.3 Publications***

Master Index to Software Release Documents

About This Book

The *Rev. 23.3 Software Release Document* provides a summary of both new and enhanced functionality to Prime® user software at Rev. 23.3. All changes to Prime user software at Rev. 23.3 are documented in this software release document. This publication consists of four chapters and two appendices:

- **Chapter 1,** Introduction, provides a summary list of new features at Rev. 23.3. It also includes special considerations and installation requirements for Rev. 23.3 and lists product retirements and restrictions.
- **Chapter 2,** New Features for the User and Programmer at Rev. 23.3, describes the new and enhanced functionality at Rev. 23.3 for the user and programmer.
- **Chapter 3,** New Features for the Operator and Administrator at Rev. 23.3, describes the new and enhanced functionality at Rev. 23.3 for the operator and System Administrator.
- **Chapter 4,** CONFIG_USERS Terminal Interface Definition, explains how to set up your terminal to support the CONFIG_USERS menus.
- **Appendix A,** ASCII Control Codes, a reference table for Chapter 4
- **Appendix B,** Rev. 23.3 Publications, lists the most recent editions of all books integral to Master Disk Revision 23.3.
- **Master Index** to the Rev. 23.0, Rev. 23.1, Rev. 23.2, and Rev. 23.3 Software Release Documents.

Most of the features documented in this book are also documented elsewhere in the PRIMOS® documentation set. PRIMOS documentation has been extensively updated and corrected at Rev. 23.3. A complete list of documentation for Rev. 23.3 is provided in Appendix B. This book should be used in conjunction with the most recent editions of PRIMOS user documentation and the online INFO and READ_BEFORE_USING 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*
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**}
Braces within brackets	Braces within brackets enclose a list of items. Choose either none or only one of these items; do not choose more than one.	**BIND** [{*pathname* / *options*}]
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**
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. 23.3. Specifically, it lists special considerations for Rev. 23.3, products being retired at this release, and new features at this revision of PRIMOS.

Special Considerations for Rev. 23.3

Hardware Requirements

Rev. 23.3 does not support non-IX mode machines. These systems include all three-digit computer room systems (e.g., the 400™, 650™, 750™, and 850™) and the 2250™.

Users wishing to perform dual-porting on a 7210 disk controller need to install a new PROM chip on the controller board. Contact PrimeService for further details.

Microcode Requirements

5310™, 5320™, 5330™, and 5340™ systems must be upgraded to at least CPU microcode Rev. J before attempting to boot Rev. 23.3. Attempting to boot Rev. 23.3 without performing this microcode upgrade 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 affect executing processes). Type DIR to display

the current microcode revision. 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.

Rev. 23.3 uses the following microcode revision levels:

CPU	*DSK7084*	*Minimum*	*Recommended*
2850	–950	D	E
2950	–953	D	E
4050	–935	E	F
4150	–928	J	K
5310	–958	J	M
5320	–960	J	M
5330	–962	K	N
5340	–956	K	N
5370	–964	C	E
6150	–940	J	K
6350	–924	S	T
6450	–941	E	F
6550	–927	L	M
6650	–943	E	F

Rev. 23.3 does not require a microcode upgrade from Rev. 23.2. The levels listed as *Minimum* are the same as those used at Rev. 23.2; these are the minimum revision levels required to fully support automated system recovery, including its hardware auto_restart component. The *Recommended* column lists the latest available revision levels; their use is recommended, but not required for Rev. 23.3. CPUs that are not listed here do not require a microcode upgrade to support Revisions 23.2 or 23.3; systems not listed here do not support automated system recovery.

If you need to update your microcode, please note that the microcode, decode net, and the maintenance processor (diagnostic processor) code must all be upgraded. Simply installing the diskette is not sufficient. The simplest way to ensure all of these items are properly installed is to power down the system, then bring it up with the new diskette installed.

Installation

No new installation procedures are required for Rev. 23.3. To install Rev. 23.3, use the *Rev. 23.0 Software Installation Guide* (IDR10176–3XA). See the PRIMOS help file HELP*>PRIMOS.TEXT>REV233.HELP for the release numbers of independent products that are needed to support Rev. 23.3 functionality.

The following items should be considered when installing Rev. 23.3.

- The instructions for upgrading an existing system to a new revision of PRIMOS require two boot-from-disk operations. These instructions show how to successfully upgrade PRIMOS to 23.3, but it is possible to simplify the installation procedure for 23.3 by skipping the first boot from disk.
- When upgrading to Rev. 23.3 without reformatting disks, you may 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. 23.3 requires translator family Release T2.2, T2.3, T3.0, or T3.1. (the T3 releases, which support registered EPFs, are recommended.) If you are modifying PRIMOS source code, you must be using Release T3.1.

  Some systems may still be running an earlier T2 release. 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 and subsequent revisions 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. 23.3.
- The FORMS/FED software is loaded slightly differently at Rev. 23.3. Previously, the FED.SAVE file was located in CMDNC0. At Rev. 23.3, CMDNC0 contains a FED.CPL file, which accesses the file SEGRUN*>FED.SEG. Customers wishing to preserve the old FED.SAVE file should rename it prior to installing Rev. 23.3. For further details, refer to the *FED User's Guide.*
- Update your version of FS_RECOVER to Version 4.0. Version 4.0 of FS_RECOVER is supplied on a separate magnetic tape shipped with Rev. 23.3. The FS_RECOVER facility can be installed independently of the revision. The available versions of FS_RECOVER are as follows:
  - Version 4.0 is supplied with Rev. 23.3. This version supports PRIMOS Rev. 21.0 and all subsequent revisions. This is the only version that supports Rev. 23.3.

- Version 3.0 is supplied with Rev. 23.2. This version supports PRIMOS Rev. 21.0 through 23.2. Support for crash dump to disk is provided in this and subsequent versions.
- Version 2.0-21.0 is available as an Independent Product Release for PRIMOS revisions 21.0 through 23.1 and is supplied with Rev. 23.1. This version is not compatible with PRIMOS revisions greater than Rev. 23.1.
- Version 1.0–21.0 is available as an Independent Product Release for PRIMOS revisions 21.0 or greater. This version *does not work* in the Rev. 23.1, 23.2 or 23.3 environments.

The installation procedures for FS_RECOVER are described in the Crash Recovery Facilities chapter of the *Rev. 23.1 Software Release Document* and in the *Using FS_RECOVER* document. Version 3.0 of FS_RECOVER is described in the third edition of *Using FS_RECOVER* (DOC13062–3LA); Version 4.0 changes are documented in an update package for this document (UPD13062–31A).

Product Retirements

The configuration directive TPDUMP is obsolete and no longer used. If a system CONFIG file contains a TPDUMP directive, PRIMOS displays a message on the system console that TPDUMP is no longer supported and is being ignored. The presence of a TPDUMP directive in your CONFIG file does not otherwise affect system configuration.

The SyncSort/PRIME product is no longer marketed, distributed, or supported by Prime Computer.

The following products will continue to be supplied as part of PRIMOS, but will not be supported; further use of these products is discouraged.

- AMLBUF directive
- AMLC command
- EDIT_PROFILE utility

EDIT_PROFILE continues to be provided, but the new CONFIG_USERS features described in this document and in the *Rev. 23.2 Software Release Document* are not supported by EDIT_PROFILE. Using one feature of CONFIG_USERS supplied at Rev. 23.2 (the Password History feature) can render your SAD permanently incompatible with EDIT_PROFILE. If you choose to use this Rev. 23.2 CONFIG_USERS feature, you will not be able to subsequently access your SAD using EDIT_PROFILE. The CONFIG_USERS features described in this document do not modify your SAD in this way. Chapter 4 of this document provides information on using CONFIG_USERS on non-Prime terminals.

Rev. 23.3 New Features

The new features of PRIMOS Rev. 23.3 are listed below, grouped into two categories: those for the user and/or programmer and those for the system Operator and/or System Administrator. 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.

- The user login procedure has been enhanced to better specify when login is permitted and to allow logins without specifying the LOGIN command.
- Online HELP has been enhanced to improve both terminal access to HELP texts and the contents of the online HELP texts.
- The LD command has been enhanced with eight new options, permitting you to select entries by file type, and to control the height and width of the LD display page.
- The new LIST_USERS command provides a wide range of information about processes currently running on a specified system.
- The MESSAGE command has been enhanced to support longer messages and messages with a multiline format.
- The TERM command has been enhanced to display the status of the break character.
- The new IF function provides conditional branching in the execution of a command line.
- The ATTRIB and WILD functions support new options that determine if a file system object has been truncated by FIX_DISK.
- The DIR$SE subroutine uses a new version of the selection_criteria structure.
- PRIMENET permits you to specify your own X.3 parameters.

Operator and Administrator Features

The following enhancements are described in Chapters 3 and 4.

- The new Model 5520 and 5540 computers are supported by enhancements to the NLBUF, NSEG, and NVMFS configuration directives, the USAGE command, and the CPUID$ and G$METR subroutines.
- The maximum number of Locate buffers has increased.
- The maximum number of PRIMENET remote users has increased.
- Partial failure of a redundant power supply is logged to DSM.
- The new Model 4601 high-capacity 8mm cartridge tape drive is supported by new ASSIGN command options and T$MT subroutine options.
- The system boot prompt has been enhanced.
- The login server has been enhanced to permit you to
  - Allow users to log in without typing LOGIN
  - Specify values and defaults for prompts of all types
  - Inform users when they may not log in
  - Log failed and/or successful logins to DSM
  - Detect non-printing characters in logins
- The LOGOUT command has been enhanced to permit you to select users for logout by name or user number, and to select multiple users for logout by process classes and/or wildcards.
- The LOUTQM configuration directive has been enhanced to provide a warning message a specified time before logging out an inactive user.
- The HELP facility has been enhanced to enable you to define terminal types for HELP support.
- The new INITIALIZE_SEARCH_RULES command supports faster user logins by loading system search rules into main memory.
- The MESSAGE command permits you to enable or disable the new longer message and multiline message features.
- The CNAME command permits you to change the name of a file system object while that object is in use.
- The Batch subsystem has been enhanced to permit
  - Longer JOB command lines and longer JOB –COMO pathnames.
  - Control of the JOB –STATUS and JOB –DISPLAY options to limit display to a specific job, or jobs submitted today, or submitted by a specified user, or on a specified queue.

- The Spooler subsystem has been enhanced to support
  - HP LaserJet printers
  - Recognition of spooled PostScript files and automatic formatting defaults
  - Aliases for mandatory printer environment attributes
  - Notification of specified users when a printer error occurs
  - Optional user access to PROP command options that control printer environments
  - Specification of maximum and minimum print job sizes
  - Enhancement of remote printing over TCP/IP
  - Rapid shutdown of multiple printer environments
  - Enhanced listing of spooler processes
- The CDD command has been enhanced with –INFO options that display record sizes recommendations for use when MAKEing a crash dump disk.
- The MAKE command has been enhanced with new and renamed options for dynamic badspot handling and sector interleaving. Other new options of MAKE support its execution by a phantom process.
- The FIX_DISK command has been enhanced with modified and renamed options for dynamic badspot handling and sector interleaving.
- A new version of FS_RECOVER is provided that provides more flexibility in specifying FIX_DISK processing and other enhancements.
- The new PDEV command enables you to interconvert a pdev number with its corresponding controller and device address. PDEV can also be invoked as a function.
- The new SPIN_UP command allows you to put a disk back on line, reversing the action of the SPIN_DOWN command.
- The STATUS DEVICES command has been enhanced to list both assigned and assignable disks.
- The SHUTDN command is provided with a new –VERIFY option, which verifies that there are no users active on a disk before shutting the disk down.
- The ISC3 communications controller has been enhanced.
- The terminal interface for CONFIG_USERS from a non-Prime terminal is documented in Chapter 4.

2 New Features for the User and Programmer at Rev. 23.3

Logging In

Implicit Login

If your System Administrator has enabled Implicit Login, you do not need to type the LOGIN command to log in to the system. You simply enter your user name at the login prompt. The system then prompts you for your password.

Implicit Login is an optional modification to the login procedure. If your system Administrator has not enabled Implicit Login, your login procedure is exactly the same as at previous Revisions. If Implicit Login is enabled, you may still specify the LOGIN command followed by your user name, exactly as you did at previous Revisions.

No Logins Permitted

The default login prompt is `Login Please.` (This prompt may be set to some other value by your System Administrator.) The login prompt is normally displayed whenever a terminal is awaiting a login (you may have to press the Return key to display it). However, at Rev. 23.3, when no user logins are currently permitted, this login prompt is not displayed at user terminals. Instead, the prompt `Please try again later` (or another value for this prompt, as established by the System Administrator) is displayed. The login prompt automatically returns when user logins are again permitted.

Online HELP Enhancements

You use the HELP command to access online help texts. At Rev. 23.3, these texts have been extensively expanded and corrected. The help texts also include new commands and command options introduced at Rev. 23.3.

At Rev. 23.3, access to the help texts facility has also been modified to permit a greater variety of terminal types to use online help. How to define additional terminal types for HELP is described in Chapter 3 of this document.

Help finds out what kind of terminal you have in one of two ways. If you specify the –TERMINAL_TYPE (–TTP) option, Help checks that the terminal type you specified is valid. If it is, Help opens the Main Menu. If you do not use the –TTP option, and the global variable .TERMINAL_TYPE$ is set, Help checks that the terminal type in .TERMINAL_TYPE$ is valid. If it is, Help opens the Main Menu.

If your terminal type is not defined, you are unable to open the Main Menu. Help displays the following message:

```
HELP could not display the MAIN MENU using the specified terminal type.
Please specify the KEYWORD and all options from the PRIMOS command level.
For additional assistance, type HELP HELP at the PRIMOS command level.
[HELP Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
```

You still have access to all the same textual information but you must enter at least the command, topic, or search-string on the command line in order to display the first menu. Note that Help uses an ASCII screen format when the terminal type is undefined; therefore screen features such as reverse video are not available.

LD Command Enhancements

At Rev. 23.3, the LD command has been enhanced with the following additional command options:

–BY_FILE_TYPE

Displays selected files sorted by file type (SAM, DAM, CAM). The –BY_FILE_TYPE option when used by itself does not limit selection to files. Abbreviated –BFT.

–CAM

Selects CAM type files. This option does not work when used with the –ON option to access a system running a PRIMOS revision prior to 23.3.

–DAM

Selects DAM type files. This option does not work when used with the –ON option to access a system running a PRIMOS revision prior to 23.3.

–SAM

Selects SAM type files. This option does not work when used with the –ON option to access a system running a PRIMOS revision prior to 23.3.

–TRUNCATED

Lists only the entries that have been truncated by FIX_DISK. Note that when using the –TRUNCATED option on directories mounted from pre-Rev. 23.3 systems, all selected file system objects will be listed, but only those files that were truncated will have `trunc` displayed in the trunc column. Abbreviated –TRUNC.

LD –DETAIL also displays the trunc column.

–NO_SIZE

Suppresses the display of file sizes. Because SAM and DAM file sizes must be calculated by PRIMOS each time they are displayed, the use of this option reduces the time required to run the LD command. The –NO_SIZE option is used only with LD –DETAIL, and cannot be used with the –SIZE or the –TOTAL options. Abbreviated –NS.

–HEIGHT *lines*

Provides support for different terminal and window sizes. *lines* is the maximum number of lines to be displayed between the `--More--` prompts. The default is 23 lines. When the –HEIGHT option is used with no value the default is assumed.

–WIDTH *characters*

Provides support for different terminal and window sizes. *characters* is the maximum number of characters to display on each line. The default is 80 characters. When the –WIDTH option is used with no value the default is assumed. Do not use the –WIDE option with the –WIDTH option.

LIST_USERS Command

LIST_USERS is a new user command at Rev. 23.3. It provides information about processes currently running on a specified system. By default, the system that you are logged in to is selected. Each process has a unique number on a specific system. User processes are named for their login ID. Several user processes may be owned by the same login ID.

The six types of processes are: user, child, slave, phantom, batch_jobs, and server.

LIST_USERS [{ *user* / –ME }] [*options*]

LIST_USERS options allow you to make selections and to control display characteristics. All options can take a wildcard string. For example:

```
LIST_USERS -ALL_DISKS SYS@@
```

selects all disks whose names begin with the string SYS.

The –HELP option displays the LIST_USERS syntax and options.

Selection Options

user	Lists users named *user*. Wildcarding is supported.
–ALL_DISKS	Lists the disks in use by all users. Abbreviated –ALLD.
–ASSIGNED_DEVICES	Lists the devices assigned to users. Abbreviated –DEV.
–BATCH_JOBS	Lists batch job processes. Abbreviated –JOB.
–CHILDREN	Lists child processes. Abbreviated –CHILD.
–DISK ***diskname***	Lists users on disk partition named *diskname*. *diskname* is the partition name without enclosing angle brackets. Use the STATUS DISKS or LIST_MOUNTS command to list available partition names.
–IDS_ONLY	Lists only the user IDs of users. It excludes user numbers and other attributes. This option cannot be used with the –DETAIL option. Abbreviated –IDO.

–ME	Lists processes belonging to you. Cannot be used if you specify *user*.
–ON *system*	Lists users on a remote system named *system*.
–PHANTOMS	Lists phantom processes. Abbreviated –PH.
–PROJECTS	Lists current project assignments for all users. Abbreviated –PROJ.
–SERVERS	Lists server processes. Abbreviated –SRV.
–SLAVES	Lists slave processes. Abbreviated –SL.
–SPOOLERS	Lists Spooler processes. This option does not work when used with the –ON option to access a system running a PRIMOS revision prior to 23.2. Abbreviated –SPOOL.
–USERS	Lists user processes. It excludes children, phantoms, servers, and slaves. Abbreviated –US.

Display Options

–DETAIL	Lists all attributes for each user you select. This option cannot be used with the –IDS_ONLY option. Abbreviated –DET.
–HEIGHT *lines*	Provides support for different terminal and window sizes. *lines* is the maximum number of lines to be displayed between the `--More--` prompts. The default is 23 lines. When you use the –HEIGHT option with no value, the default is assumed.
–NO_HEADER	Suppresses the top header line and process type headers. This option is most useful when combined with the –SINGLE_COLUMN option. Abbreviated –NHE.
–NO_WAIT	Suppresses the `--More--` prompt and does not pause after every page of output, but scrolls the output continuously. Abbreviated –NW.
–SINGLE_COLUMN	Lists each entry on a separate line. This option is useful with the –NO_HEADER option to provide an uninterrupted list of all users. Abbreviated –SGLCOL.

-WIDTH ***characters***	Provides support for different terminal and window sizes. *characters* is the maximum number of characters to display on each line. The default is 80 characters. When you use the -WIDTH option with no value, the default is assumed.

Examples

To list processes belonging to you, use the command

```
OK, LIST_USERS -ME

 2 users:

73  BSB    163 BSB(me)
OK,
```

To list all processes with LOG in their names, use wildcards as follows

```
OK, LIST_USERS @@LOG@@

 2 users:

18 KELLOGG          36 LOG

 3 servers:

316 LOGOUT_SERVER   317 LOGIN_SERVER  320 DSM_LOGGER
OK,
```

To list all the servers currently running, use the command

```
OK, LIST_USERS -SERVERS

 14 servers:

  1 SYSTEM           253 NETMAN           313 NTS_SERVER
314 TIMER_PROCESS    315 BUFFER_SERVER    316 LOGOUT_SERVER
317 LOGIN_SERVER     319 DSMSR            320 DSM_LOGGER
321 SYSTEM_MANAGER   323 NAME_SERVER      324 UBI_SERVER
325 DSMASR           326 DSMASR
OK,
```

To list all the disks currently in use, use the command

```
OK, LIST_USERS -ALL_DISKS

 5 users:

  3  FRED                 <OPRTNS>
  4  BARNEY               <OPRTNS>
 18  GLENN                <OPRTNS>
 29  WILMA                <OPRTNS>  <SYSS51>
 44  BETTY                <OPRTNS>  <SYSS51>

  3 phantoms:

254  DBN_SERVER           <SYSS49>
255  MAIL_SERVER          <SYSS49>
258  BATCH_SERVICE        <SYSS49>


 1 child processes:

288  GIVE                 <TPUBS5>  <OPRTNS>

 10 servers:

  1  SYSTEM               <OPRTNS>  <SYSS49>
314  TIMER_PROCESS        <SYSS49>
315  BUFFER_SERVER        <SYSS49>
316  LOGOUT_SERVER        <SYSS49>
317  LOGIN_SERVER         <SYSS49>
319  DSMSR                <SYSS49>
320  DSM_LOGGER           <SYSS49>
321  SYSTEM_MANAGER       <SYSS49>
323  NAME_SERVER          <SYSS49>
325  DSMASR               <SYSS49>
```

MESSAGE Command Enhancements

At Rev. 23.3, there are four user enhancements to the MESSAGE command.

- You can create messages longer than the previous maximum message line length.
- You can create multiline messages with explicit carriage returns.
- You can display the current system settings for message line length and multiline messages.

- The system provides notification of unsent messages.

Subroutine support is also provided for these new features.

Maximum Message Line Length

The maximum length of a message line is configured by the System Administrator as either 80 or 512 characters. By default, the maximum message line length is 80 characters.

Multiline Messages

The CRLF (Carriage Return/Line Feed) feature allows you to send multiline messages. This feature must be enabled on your system by the System Administrator. If enabled, you can use the following special characters to create multiline messages:

%/	Places a CRLF (line break) at that point in the message when displayed on the recipient's terminal.
%-Return	Places a CRLF (line break) at that point in the message both when displayed on the recipient's terminal and on your screen when typing the message. A return preceded by a % does not end the message line.
%%	Displays a percent sign in the message. The additional % suppresses the special meaning of the percent sign described above.
~	Provides line continuation when typing the message. If you wish to type a one-line message on more than one line, use the tilde (~) to suppress the CRLF (when a 512-character limit is set) or the end-of-message carriage return.

To enter a three-line message on one line, for example, type:

```
OK, M BOB -NOW
***************%/*      Hello?      *%/***************
```

The message as it appears on the recipient's terminal is

```
*** SALLY (user 45 on ARTEMIS ) at 12:34
***************
*   Hello?    *
***************
```

To type the message the same way as you want it to appear (to make sure the message is correctly lined up, for instance), use the %-Return combination, for example:

```
OK, M BOB -NOW
***************%
*   Hello?    *%
***************
```

The message appears at the recipient's terminal just as you typed it (without the % signs).

When you want to include a literal % sign at the end of a line, use the combination %%. (A literal % sign within a message line can be represented by either a single or double % sign.) For example:

```
OK, M BOB -NOW
This has improved 100%%
```

The recipient sees:

```
*** SALLY (user 45 on ARTEMIS ) at 12:38
This has improved 100%
```

To send a message you want to appear on one line that you need to enter on more than one line, use the tilde (~). When you enter the message:

```
OK, M BOB -NOW
I have been looking over the reports ~
and they look fine.
```

The recipient sees:

```
*** SALLY (user 45 on ARTEMIS ) at 12:45
I have been looking over the reports and they look fine.
```

MESSAGE -DISPLAY Command Option

The new MESSAGE -DISPLAY (-DP) option displays the current maximum message line length and the status of the CRLF features on your system, as shown in the following example:

```
OK, MESSAGE -DISPLAY
Maximum message length allowed: 80 characters.
CRLF substitutions: DISABLED
```

MSG$ST and SMSG$ Subroutine Support

MSG$ST: The MSG$ST subroutine supports two new keys:

K$CRLF — When you specify this key, the *user_num* parameter becomes an output parameter that returns the status of support for CRLF coding for multiline messages. If the returned value is 0, multiline messages are not currently supported on your system; if the returned value is 1, multiline messages are supported.

K$MLEN — When you specify this key, the *user_num* parameter becomes an output parameter that returns the maximum number of characters allowed in a message.

SMSG$: This routine returns a new error code:

E$METL — This error code is returned when you attempt to send a message to a remote system and the message is too long for the maximum message length on that system. The message was sent, but was truncated to the maximum message length on that system.

For further details, refer to *Subroutines Reference III: Operating System.*

Notification of Unsent Messages

The MESSAGE command now notifies the sender if a message was not sent due to insufficient output buffer size.

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

TERM Command Enhancement

At Rev. 23.3, the TERM –DISPLAY option has been enhanced to display the status of the break character, as set by TERM –BREAK. The following two TERM –DISPLAY examples show the on and off status of the break character:

```
OK, TERM -DISPLAY
Erase = (non-printing char)    = '210 octal
Kill  = ?    = '277 octal
Full Duplex
Xon/Xoff  enabled
Break on
OK, TERM -BREAK OFF
OK, TERM -DISPLAY
Erase = (non-printing char)    = '210 octal
Kill  = ?    = '277 octal
Full Duplex
Xon/Xoff  enabled
Break off, inhibit counter = 1
OK,
```

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

IF Command Function

IF provides conditional branching.

```
[IF ( expression | boolean-value)  -THEN statement
  [-ELSE statement]]
```

expression can be any unary or binary expression.

IF accepts expressions containing the logical operators & (and), | (or), and ^ (not); the arithmetic operators +, –, *, /, unary +, and unary –; and the relational operators =, <, >, <=, >=, and ^=. Note that exponentiation is not supported. The operator precedence is as follows:

```
Highest:   ^  unary +  unary –
           /  *
           +  –
           =  <  >  <=  >=  ^=
           &
Lowest:    |
```

Note All operators evaluated by IF must be delimited by blanks. This restriction resolves the ambiguity that can arise from the fact that *, <, and > are also valid pathname characters. If parentheses are used within *expression*, each (and) must be bounded by blanks.

For more information about expressions, see the CALC function in the *PRIMOS Commands Reference Guide.*

The following are examples of valid expressions:

```
1 < 2
FLORIDA <= CALIFORNIA
'FLORIDA' ^= 'CALIFORNIA'
%terminal_type$% = PT200
^ [exists foobar]
^ true
%.a% < %.b% & %.b% < %.c%
```

The following are examples of invalid expressions:

```
1 < hello   (attempt to compare integer and character data)
^ 123A      (attempt to negate a character string)
```

boolean-value is the string TRUE, FALSE, or any command function that returns TRUE or FALSE.

The following are examples of valid *boolean-values*:

[exists 'login.cpl'] *(returns a boolean value)*
TRUE
FALSE

statement may be any text string, command, or command function.

The following are examples of valid *statements*.

```
'Hello!'
[IF %.TERMINAL_TYPE$% ^= PT200-C -THEN EXIT]
EMACS
```

Examples

Use the IF command to handle variations in your input to a command by creating an abbreviation. (Remember that ABBREV commands must be on one line and must be preceded by the syntax suppression character (~) when variables are to be a part of the abbreviation.)

Creating an abbreviation called EXIST that uses IF to print different messages:

```
OK, ~ ABBREV -AC EXIST TYPE %1% [IF [EXISTS %1%]
-THEN ' EXISTS!' -ELSE ' DOES NOT EXIST!']
```

Using the abbreviation created above:

```
OK, EXIST LOGIN.CPL
LOGIN.CPL EXISTS!
OK, EXIST FOOBAR
FOOBAR DOES NOT EXIST!
```

In the above example, the abbreviation EXIST prints the argument you entered, then, based on the boolean value returned by the EXISTS function, prints the second part of the message.

Using IF to compare strings:

The following abbreviation called EDIT invokes ED when typed at the system console, and EMACS if typed at any other terminal.

```
OK, ~ ABBREV -AC EDIT [IF [USER_INFO -TYPE] = CONSOLE
-THEN ED -ELSE EMACS]
```

Use the IF to include your current attach point in your prompt string. The IF function in the example below checks to see if the home system name (the RDY variable %SN) matches the name of the system where the directory is located (%RN). If it does not match, then the function call is replaced with '*->remote_systemname*'. If it does match, then the function call becomes a null string. Note that the RDY command requires %[and %] for the function call delimiters.

```
OK, RDY -RL '%SN%[IF %SN ^= %RN -THEN ''->''%RN%]:%A %L'
-EP -LONG
PLATO:<PLTODK>ARISTOTLE
OK, A REMOTE_UFD
PLATO->SATRE:REMOTE_UFD
```

Other Function Enhancements

[ATTRIB -TRUNC]

The ATTRIB function supports the new -TRUNC option, which determines if the specified file system object has been truncated by FIX_DISK. It returns a TRUE if the object has been truncated. If the object has not been truncated, it returns FALSE. This function can be called from the command line or from within a CPL program. For further details on this function, refer to the *PRIMOS Commands Reference Guide*. For further details on truncation, refer to information on the LD command found in this chapter.

[WILD -TRUNC]

The WILD function supports the new -TRUNC option, which matches file system objects that have been truncated by FIX_DISK. It returns a list of names of truncated objects. If no objects have been truncated, it returns a null string (which appears on the screen as a blank line). This function can be called from the command line or from within a CPL program. For further details on this function, refer to the *PRIMOS Commands Reference Guide*. For further details on truncation, refer to information on the LD command found in this chapter.

DIRLS, DIRRD, and DIR$SE Subroutines

The DIRLS, DIRRD and DIR$SE subroutines all use structures named *dir_entry*. These structures are similar, but are *not* identical. Always use the structure specified for the subroutine that you are calling. For further details on these subroutines, refer to the *Subroutines Reference II: File System*.

The DIR$SE subroutine uses a new version of the structure named *selection_criteria*. Use this new Version 2 structure with Rev. 23.3.

```
DCL 1 selection_criteria BASED,
      2 version_no FIXED BIN(15),      /* Must be 2 */
      2 wild_ptr PTR OPTIONS (SHORT),
      2 wild_count FIXED BIN(15),
      2 desired_types,
        3 dirs BIT(1),
        3 seg_dirs BIT(1),
        3 files BIT(1),
        3 access_cats BIT(1),
        3 rbf BIT(1),
        3 trunc_files BIT(1),
        3 file_types,
          4 sam_files BIT(1),
          4 dam_files BIT(1),
          4 cam_files BIT(1),
        3 tbr_files BIT(1),     /* Must be '0'b */
        3 spare BIT(6),
      2 modified_before_date FIXED BIN(31),
      2 modified_after_date FIXED BIN(31),
      2 backed_up_before_date FIXED BIN(31),
      2 backed_up_after_date FIXED BIN(31),
      2 created_before_date FIXED BIN(31),
      2 created_after_date FIXED BIN(31),
      2 accessed_before_date FIXED BIN(31),
      2 accessed_after_date FIXED BIN(31);
```

The new fields in this *selection_criteria* structure are the following:

trunc_files
Set this bit to select directory entries that have been truncated by FIX_DISK.

sam_files
Set this bit to select all files of the SAM (Sequential Access Method) file type.

dam_files
Set this bit to select all files of the DAM (Direct Access Method) file type.

cam_files
Set this bit to select all files of the CAM (Contiguous Access Method) file type.

tbr_files
Set this field to '0'b. This field is reserved for future use.

PRIMENET X.3 Support

At Rev. 23.3, you can configure PRIMENET X.3 parameters on a per-user basis for remote login users. X.3 profiles are created by the user using an editor such as EMACS or ED. The profiles are stored in the directory named PRIMENET*>X3_PROFILES. Each profile is a separate file, identified by its filename.

A profile file defines one or more X.3 parameters, one parameter per line, using the following format:

```
parameter:value
```

For example, to create a profile named SAMPLE:

1. Attach to PRIMENET*>X3_PROFILES. Use ED or EMACS to create a file named SAMPLE within this directory.
2. Within this file, specify each X.3 parameter by number, followed by a colon (:), followed by the corresponding parameter value.

   For example, to set parameter 4 (Idle Timer Delay) to value 1 (1/20 second delay), type in 4:1
3. Specify the other X.3 parameters and values in the same way, one parameter per line. Parameters not defined in the profile file take their default values.

In PRIMENET*>X3_PROFILES, profile files to use with EMACS and PRIMOS have been provided.

When creating X.3 profiles, consider such factors as network delays, application design, and transmission costs. For example, if it is necessary for the application to do all character echoing rather than have the PAD echo the input, the Idle Timer Delay, parameter 4, should be set to a small value to minimize echo delay. As a result, however, network traffic will be greatly increased, thereby increasing transmission costs.

Per–user Profiles

The X29 command is used to load an X.3 profile into a PAD. It has the following format.

```
X29 profile_filename
```

A user can execute the X29 command in the LOGIN.CPL or LOGIN.COMI file. If multiple profiles are used by a single user, the user can create abbrevs for each profile. Do not use a pathname with the X29 command.

X.3 Parameters

Parameter	*Description*	*Values*	*Description*
1	PAD recall using a character defined by user	0	No PAD recall character.
		1	Character <DLE> (default).
		32 – 126	Using one IA5 (International Alphabet #5) graphic character.
2	Echo	0	No echo.
		1	Echo (default).
3	Selection of data forwarding signal	0	No data forwarding signal.
		2	Character <CR> (default).
		6	Characters <CR>, <ESC>, <BEL>, <ENQ>, <ACK>.
		18	Characters <CR>, <EOT>, <ETX>.
		126	All characters in columns 0 and 1 of IA5 and <DEL>.
4	Selection of idle timer delay	0	No idle timer.
		1 – 255	Value of idle timer in 20ths of seconds.
5	Ancillary device control	0	No use of X–ON and X–OFF.
		1	Use of X–ON and X–OFF.

6	Control of PAD service signals	0	No service signals are transmitted to the DTE–C.
		1	PAD service signals are transmitted.
		5	PAD service signals and the PAD prompt service signal are transmitted.
7	Selection of operation of PAD on receipt of break signal from the DTE–C	0	Nothing.
		1	Interrupt.
		2	Reset.
		8	Escape from data transfer state.
		21	Discard output, interrupt and indication of break.
8	Discard output	0	Normal data delivery.
		1	Discard output.
9	Padding after carriage return	0	No padding after <CR>.
		1 – 7	Number of pad characters to be inserted after <CR>.
10	Line folding	0	No line folding.
		1 – 255	Number of characters per line.
11	Binary speed	0	110 bits/second.
		1	134.5 bits/second.
		2	300 bits/second.
		3	1200 bits/second.
		4	600 bits/second.
		5	75 bits/second.
		6	150 bits/second.
		7	1800 bits/second.
		8	200 bits/second.
		9	100 bits/second.
		10	50 bits/second.
		11	75/1200 bits/second.
		12	2400 bits/second.
		13	4800 bits/second.
		14	9600 bits/second.
		15	19200 bits/second.
		16	48000 bits/second.
		17	56000 bits/second.
		18	64000 bits/second.
12	Flow control of the PAD by DTE–C	0	No use of X–ON and X–OFF.
		1	Use of X–ON and X–OFF.

13	Linefeed insertion after <CR>	0	No linefeed insertion.
		1	Insert linefeed after transmission of <CR>.
		4	Insert linefeed after echo of <CR> to DTE–C.
		5	Insert linefeed after transmission to the DTE–C and after echo of <CR>.
		6	Insert linefeed in data stream after <CR> from DTE–C and after echo of <CR> to DTE–C.
		7	Insert linefeed in data stream to and from the DTE–C and after echo of <CR> to the DTE–C.
14	Padding after linefeed	0	No padding after linefeed.
		1 – 7	Number of padding characters inserted after linefeed.
15	Editing	0	No editing.
		1	Use of editing in the data transfer state.
16	Character delete	0 – 127	One character from IA5.
17	Line delete	0 – 127	One character from IA5.
18	Line display	0 – 127	One character from IA5.

Documentation Corrections

Instruction Sets Guide

The JSXB instruction is only supported in V mode long. It is *not* supported in R mode long, as described on page 2–66.

The 64V mode short form summary of addressing modes shown in Table B–1 on page B–2 contains an error. The correct instruction type for IXS binary value 111 is Indirect, postindexed.

For further details, refer to the *Instruction Sets Guide* (DOC9474–2LA).

SEG and LOAD Reference Guide

The following error message was omitted from Chapter 8 of this manual:

```
xxxx: ILLEGAL REDEFINITION OF SHORT COMMON TO LONG COMMON
```

This error occurs when a user program has declared the same common block in different modules with different lengths. For example, the program in one module declares a structure with a length of 10 bytes, then in another module declares a structure with the same name as having a length of 20 bytes. Check your program for redundant and incompatible structure or array declarations.

Source Level Debugger User's Guide

Table 6–1, PL/I-G, Pascal, FORTRAN, and C Supported Functions, shown on page 6–10, contains an error. This table should not include UNSPEC. UNSPEC is not a function supported by DBG.

Programmer's Guide to Prime Networks

The XLGC$ subroutine, described on page 4–19, can take a XK$SAV key. The description of this key should contain the following note:

The status vector exists only from the time when XLGC$ is called with the XK$SAV key until the time when a call is made to accept a connection request through either X$ACPT, X$FACP, X$SACP, or XLACPT. The virtual circuit block, along with the status vector, is no longer valid for the user after the virtual circuit block is passed off.

MAGNET User's Guide

When using the COPY subcommand of the MAGNET utility to perform a tape-to-tape copy, make sure that the MAXIO parameter has been set high enough to include the largest tape record block on the source tape. The largest possible tape record block for the source tape drive is specified in the MTRS configuration directive (see the *System Administrator's Guide, Volume I: System Configuration*). Previously, if you set MAXIO too low when performing this operation, MAGNET would neither transfer the full contents of the original tape to the duplicate tape, nor issue a warning message. With Rev. 23.3 (and Revisions 22.1.4 and 22.1.5), MAGNET issues a DMx overrun warning message when this situation occurs. For further details, refer to the *MAGNET User's Guide.*

Subroutines Reference IV: Libraries and I/O

T$SLC0 Subroutine:

The key value of 2 for this subroutine does not work and should be avoided.

CUS$CHANGE_SYSTEM and CUS$LIST_SYSTEM Subroutines:

These subroutines take Version 2 of the PRIMOS_System_Attributes structure, which incorporates new fields added to support CONFIG_USER features introduced at Rev. 23.2 and Rev. 23.3. This structure is the same as described in the *Rev. 23.2 Software Release Document*, except that the final two fields were listed as Reserved in that document. Note also the correction of the data type for the `Prefix_for_Origin` field.

```
declare 1 PRIMOS_System_Attributes based aligned,
            2 Version fixed bin(15),
            2 Use_System_Defaults bit(1) aligned,
            2 Use_Projects bit(1) aligned,
            2 Use_Server_Attributes bit(1) aligned,
            2 No_Password_on_LOGIN_Line bit(1) aligned,
            2 Password_Minimum_Length fixed bin(15),
            2 Password_Maximum_Length fixed bin(15),
            2 Password_Required bit(1) aligned,
            2 Verify_Password_Format bit(1) aligned,
            2 Computer_Generated_Password bit(1) aligned,
            2 Specify_Password bit(1) aligned,
            2 Do_NOT_Create_Origin bit(1) aligned,
            2 Verify_Origin bit(1) aligned,
            2 Prefix_for_Origin char (128) var aligned,
            2 Specify_Origin bit(1) aligned,
            2 Password_Lifetime fixed bin(15),
            2 Command_Levels fixed bin(15),
            2 Program_Invocations fixed bin(15),
            2 Dynamic_Segments fixed bin(15),
            2 Static_Segments fixed bin(15),
            2 Ver_1,
               3 ISC_Sessions fixed bin(15),
               3 ISC_Synchronizers fixed bin(15),
               3 ISC_Timers fixed bin(15),
               3 Force_Password_Change bit (1) aligned,
               3 Initial_Password_Change bit (1) aligned,
               3 Use_Rev2_Encryption bit (1) aligned,
            2 Ver_2,
               3 Allowable_Failures fixed bin(15),
               3 Default_Concurrent_Logins fixed bin(15),
               3 Default_Accnt_Disposition fixed bin (15),
               3 Max_PW_CHG_HR fixed bin(15),
               3 PW_History bit(1) aligned,
               3 Log_Successful_Logins bit(1) aligned,
               3 Log_Failed_Logins bit(1) aligned;
```

Subroutines Reference V: Event Synchronization

The ISC Session Configuration Block contains a SendFailureConfig bit, as shown in the following structure. In *Subroutines Reference V: Event Synchronization* (pages 9-2 and E-4), this field is included in the reserved bits. Please update your documentation.

```
dcl 1 SessionConfigurationBlock based aligned,
       2 Version fixed bin(15),
       2 SessionServices,
          3 NormalService bit(1) unaligned,
          3 ExpeditedService bit(1) unaligned,
          3 SyncsToBeUsed,
              4 ReadyToSend bit(1) unaligned,
              4 ReadyToSendExpedited bit(1) unaligned,
              4 ReadyToReceive bit(1) unaligned,
              4 ReadyToReceiveExpedited bit(1) unaligned,
              4 BufferAvailable bit(1) unaligned,
              4 SessionResponsePending bit(1) unaligned,
              4 ExceptionPending bit(1) unaligned,
          3 SendFailureConfig bit(1) unaligned,
            /* Configure for ReadyToSend sync notification
               after send failure with threshold
                conditions ? */
          3 Reserved bit(6) unaligned,
       2 QueueLengths,
          3 NormalSend fixed bin(15),
          3 NormalReceive fixed bin(15),
          3 ExpeditedSend fixed bin(15),
          3 ExpeditedReceive fixed bin(15),
       2 QueueThresholds,
          3 NormalSend fixed bin(15),
          3 NormalReceive fixed bin(15),
          3 ExpeditedSend fixed bin(15),
          3 ExpeditedReceive fixed bin(15),
       2 MaxControlLength fixed bin(15),
       2 MaxDataLength fixed bin(15),
       2 MaxExpeditedLength fixed bin(15),
       2 ExistingSessionId fixed bin(15),
       2 MessageArea,
          3 BlockSize fixed bin(15),
          3 NumberOfBlocks fixed bin(15),
          3 Reserved fixed bin(15);
```

PRIMOS Error Codes

The following are additional PRIMOS standard error codes since Rev 23.0. For a list of the other PRIMOS error codes, refer to the *Advanced Programmer's Guide: Appendices and Master Index.*

`E$ND1S (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. Cold start 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 the *Rev. 23.1 Software Release Document.*

`E$UNLI (367)` `Specified user not logged in.`

Used by the WATCH command, as described in the *PRIMOS Commands Reference Guide.*

`E$MULI (368)` `Multiple users logged in with this user name.`

Used by the WATCH command, as described in the *PRIMOS Commands Reference Guide.*

`E$CWAT (369)` `Can not watch.`

Used by the WATCH command, as described in the *PRIMOS Commands Reference Guide.*

`E$UBW (370)` `User being watched by another user.`

Used by the WATCH command, as described in the *PRIMOS Commands Reference Guide.*

`E$ICUT (371)` `Incompatible open unit types.`

Used by the UT$COPY subroutine, as described in the *Rev. 23.1 Software Release Document.*

E$IANL (372) Initial attach point is not local.

Used by the UT$COPY subroutine, as described in the *Rev. 23.1 Software Release Document.*

E$NOWR (373) Not open for write access.

Used by the UT$COPY subroutine, as described in the *Rev. 23.1 Software Release 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 the *Operator's Guide to System Commands.*

E$BFMT (376) Format not supported on this drive.

Used by the ASSIGN –FORMAT command for tape drives, as described in this document.

E$ACPG (377) Cannot assign active paging disk.

Used by the ASSIGN DISK command, as described in the *Operator's Guide to System Commands.*

E$ACDD (378) Cannot assign active crash dump disk.

Used by the ASSIGN DISK command, as described in the *Operator's Guide to System Commands.* Crash dump disks are described in the *Rev. 23.2 Software Release Document.*

E$PWRU (379) Password recently used.

Used by the CHG$PW subroutine when the system has been configured to prevent the reuse of old login passwords. The password specified is one of the last 16 passwords specified for this user. CHG$PW is described in *Subroutines Reference III: Operating System.* Preventing the reuse of passwords is described in the *Rev. 23.2 Software Release Document.*

E$CPLE (380) Change password limit exceeded.

Used by the CHG$PW subroutine when the system has been configured to prevent the reuse of old login passwords. You have attempted more password changes per hour than are permitted by your system's user configuration. CHG$PW is described in *Subroutines Reference III: Operating System.* Preventing the reuse of passwords is described in the *Rev. 23.2 Software Release Document.*

E$METL (381) Message truncated on remote system.
Used by the MESSAGE command, as described in this document and the *PRIMOS Commands Reference Guide.*

3 New Features for the Operator and Administrator at Rev. 23.3

This chapter describes new features at PRIMOS Rev. 23.3 that are of interest to system operators and System Administrators. In most cases, these features are summarized here, and the reader is directed to the appropriate book for further details.

Model 5520 and 5540 Computers

Note The Model 5520 and 5540 computers will be available shortly after the release of Rev. 23.3. A special fix revision of Rev. 23.3 is needed to support the Model 5520 and Model 5540 computers. Consult your PrimeService representative concerning availability of these computers and further details on the operating system Revision required for them.

The information in this section applies only to users of 5520 and 5540 computers. Users of other Prime computers can ignore the information in this section.

The 5500 Family of computers is a family of high-end computers employing a multiprocessor architecture and a new instruction execution pipeline that significantly improves throughput. The initial two models in this family are the Model 5520 dual processor and the Model 5540 quad processor. These processors are described in greater detail in the *5500 Family Handbook.*

CPUID$ Subroutine

The new 5500 processors return the following values to the CPUID$ subroutine.

46	P5520
48	P5540

For further details, refer to the *Subroutines Reference III: Operating System.*

USAGE –MULTI Command Option

The USAGE operator command supports the new –MULTI option. This option is only used with multiprocessor systems. It displays up to eight separate *%Idl* values for each processor on a multiprocessor system, showing the actual percentage of idle time on each CPU. *%Idl* values show the average amount of idle time. These are represented as *%Idl1*, *%Idl2*, etc. Because these values are independent, each can approach 100% of usage.

Issuing USAGE without the –MULTI option displays a new pair of fields, *%Idle* and *%Sch*. *%Idle* represents the average idle time for all processors. *%Sch* represents the percent of the CPU time used by the Scheduler process. Prior to Rev. 23.3, this time used to be included in the *%Idl* totals.

If you specify –ALL without –MULTI, the eight *%Idl* values are not displayed.

For further details, refer to the *Operator's Guide to System Commands*.

5500 Family System Limits

Some system limits have been increased to accommodate the 5500 Family computers. The following PRIMOS limits have been increased:

- Locate buffers = 8192
- Segments = 64K
- Physical memory = 2 gigabytes

The increased limits for segments and physical memory apply only to 5500 Family computers. If you specify one of these increased limits for a non-5500 system, the system defaults to the maximum permitted value for that system.

NLBUF Configuration Directive: You use the NLBUF directive to configure the number of locate buffers on your system. NLBUF has been modified to permit up to 8192 locate buffers. The optimal number of locate buffers to allocate depends on the applications running on your system. Specify the number of locate buffers to NLBUF as an octal number.

For further details, refer to the *System Administrator's Guide, Volume I: System Configuration*.

NSEG and NVMFS Configuration Directives: You use the NSEG directive to configure the number of segments on your system. NSEG has been modified to permit up to 65536 segments. The total of NSEG and NVMFS directives has been increased to 65536. The maximum value for NVMFS has not changed; it is still 4096.

For further details, refer to the *System Administrator's Guide, Volume I: System Configuration*.

G$METR Subroutine

To accommodate the larger system limit values for 5500 Family computers, the G$METR subroutine has been modified. At Rev. 23.3 and subsequent revisions, declare the following *meter_mem* structure in your program:

```
DCL 1  meter_mem BASED,
       2  version FIXED BIN (15),
       2  max_segs FIXED BIN (31),
       2  total_segs FIXED BIN (31),
       2  max_pages FIXED BIN (31),
       2  total_pages FIXED BIN (31),
       2  wired_pages FIXED BIN (31),
       2  init_pgrecs (8) FIXED BIN (31),
       2  cur_pgrecs (8) FIXED BIN (31),
       2  cur_vmfa FIXED BIN (31),
       2  tot_vmfa FIXED BIN (31),
       2  iopfcn FIXED BIN (31),
       2  user (4063),
          3  segments FIXED BIN (15),
          3  physical_mem FIXED BIN (31),
          3  wired_mem FIXED BIN (31);
```

Specify *version* 10 to use this structure. Allocate a buffer of 49 for the static portion of this structure, and a buffer of 5 for each user.

For further details, refer to *Subroutines Reference III: Operating System.*

System Limits

Locate Buffers	The maximum number of Locate buffers has increased for all 50 Series system to 8192 decimal (20000 octal). The NLBUF configuration directive has been modified accordingly.
PRIMENET Users	The maximum number of PRIMENET™ remote users has increased to 511. You specify the number of remote users in the NRUSR configuration directive. See the *System Administrator's Guide, Volume I: System Configuration.*

Redundant Power Supply

All Prime computer systems shipped since Revision 23.2 contain an improved redundant power supply. Hardware redundancy prevents the failure of a component of the power supply from resulting in a power failure. In the event of a power supply component failure, the system issues the following message to the DSM log (or to the system console, if DSM is not operating):

```
REDUNDANT SYSTEM POWER SUPPLY FAILURE. Schedule PrimeService
```

PrimeService should be contacted because this message may indicate the loss of redundant protection of your power supply.

Model 4601 Cartridge Tape Drive

Rev. 23.3 supports the new Model 4601 tape drive. This is a high-performance, high-capacity 8mm cartridge tape drive packaged in the industry-standard 5.25-inch form factor. It uses the same tape cartridges as Model 4595, 4596, and 4598 tape drives. (These drives are referred to collectively as Model 459x tape drives.)

The Model 4601 uses helical-scan recording technology employing dual read and write head pairs. Because each pair of heads can read or write two tracks of data simultaneously, the data transfer rate and maximum storage capacity is double that of the Model 4595 tape drive, which uses unpaired read and write heads. The Model 4601 provides up to 5 gigabytes of data storage capacity with a maximum sustained data transfer rate of 500 Kbytes/second.

The Model 4601 tape drive is a SCSI tape drive that is only supported on the Model 7210 controller. For further details, see the *Prime 8mm Cartridge Tape Drive User's Guide.*

ASSIGN Command for the Model 4601 Tape Drive

The tape cartridges used with the Model 4601 and the Model 459x tape drives are identical; however, the default recording formats for these two drives are incompatible. To permit exchange of tapes between these two tape drives, the Model 4601 supports the reading and writing of data in both tape formats: the dual-track Model 4601 format or the single-track Model 459x format. Tapes written in Model 4601 format can only be read by a Model 4601 tape drive. Tapes written in Model 459x format can be read either by an Model 459x or by a Model 4601 tape drive. The Model 4601 tape drive automatically determines which format a tape has been written in, and reads or writes the tape in the appropriate format.

At Rev. 23.3, the ASSIGN command for magnetic tapes takes a new –FORMAT option. You can use ASSIGN –FORMAT to specify which format you wish to use when writing a tape on a Model 4601 tape drive. (ASSIGN –FORMAT is only used with a Model 4601 tape drive.) By default, the Model 4601 writes tapes in Model 4601 format. The following are the available –FORMAT values:

–FORMAT 4598	Writes tape in Model 459x-compatible mode. Tapes written on a Model 4601 tape drive in this format can be read on either a Model 4601 or a Model 459x tape drive. However, less data (approximately 2GB) can be written per tape than in Model 4601 mode.
–FORMAT 4601	Writes tape in Model 4601-compatible mode. This option is the default. Tapes written on a Model 4601 tape drive in this format *cannot* be read on a Model 459x tape drive. Approximately 5GB of data can be written per tape in this mode.

You can use these options to write a new tape or to overwrite an existing tape. You can only use a –FORMAT option when the tape drive you have specified is a Model 4601 and the tape is positioned at the beginning of tape (BOT). (Any other use of the –FORMAT option is ignored.) When adding to an existing tape, the Model 4601 automatically writes in the existing format of the tape cartridge. Once you have assigned a Model 4601 tape drive to Model 459x format, it continues to write every new tape in Model 459x format until you issue another ASSIGN command. An ASSIGN command without a –FORMAT option restores a Model 4601 to its default mode.

T$MT Subroutine Support for the Model 4601 Tape Drive

You can perform tape I/O with the Model 4601 using the T$MT subroutine. The T$MT controller ID for the Model 4601 is controller version 13, which represents a Model 7210–001 controller with a drive type Model 4601. Refer to Table 7-4 in the *Subroutines Reference Guide IV: Libraries and I/O*.

The Model 4601 uses two new commands that are only supported on controller version 13:

- Write in Model 459x format
  Octal: 100400
  Hex: 8100
- Reset the drive to write in Model 4601 format
  Octal: 100420
  Hex: 8110

Controller version 13 supports the following commands:

hex	*octal*	
8000	100000	Select transport and get status
C000	140000	Return controller ID
8020	100040	Rewind and unload
0020	000040	Rewind to BOT
2440	022100	Backspace one file mark
6440	062100	Backspace one record
2490	022220	Write file mark
6480	062200	Forward one record
2480	022200	Forward one file mark
4590	042620	Write record, two characters per halfword
4580	042600	Read record, two characters per halfword
80D0	100320	Erase from current position to EOT
80E0	100340	Select unbuffered mode
80F0	100360	Select buffered mode
8100	100400	Select the Model 459x format to write in
8110	100420	Reset to write in Model 4601 format

Refer to Table 7-5 in the *Subroutines Reference Guide IV: Libraries and I/O*.

For version 13 controllers, T$MT returns values for statv(4) bit 8. If statv(4) bit 8 is set to 1, then Model 459x write format is selected. If statv(4) bit 8 is set to 0, then Model 4601 write format is selected. Refer to Table 7-8 in the *Subroutines Reference Guide IV: Libraries and I/O*.

Error Messages

At Rev. 23.3, tape drives return the following additional messages:

`WARNING: the tape drive needs to be cleaned.`

Indicates that the tape drive heads and rollers need to be cleaned. A similar message is also logged to DSM. DSM also logs a message when a tape drive has been cleaned.

`Unable to reserve drive. Already reserved on another system.`

The drive is dual-ported and the other system has control of the drive. You cannot ASSIGN the drive until the other system relinquishes control.

`Format not supported on this drive.`

You attempted to use the ASSIGN –FORMAT option on a tape drive that is not a Model 4601. The –FORMAT option was ignored and the ASSIGN operation completed.

System Boot Procedure

When booting the system, the boot procedure prompts you for the system name. This prompt has been slightly modified as follows:

```
Enter SYStem NAMe:
```

This typographical change makes the system name prompt reflect the SYSNAM configuration directive and agree in form with the other boot prompts. For further details on booting, refer to the Handbook for your computer system.

Login Enhancements

Rev. 23.3 gives the System Administrator greater control over the display of login prompts, including whether users need to actually type the word LOGIN. It also enables you to maintain a DSM log of all successful and/or failed login attempts.

START_LSR Command

At Rev. 23.3, the START_LSR command has been enhanced with four new options. The complete syntax for START_LSR is as follows:

START_LSR [**-ERROR_PROMPT** *error_prompt*
-IMPLICIT_LOGIN
-MAXUSR_PROMPT *maxusr_prompt*
-PROMPT *lsr_prompt*
-READY_PROMPT *ready_prompt*
-REDISPLAY_PROMPT
-RETRIES *n*
-HELP]

The four new options are as follows:

-ERROR_PROMPT ***error_prompt***

Sets up the LOGIN_SERVER *error_prompt*. The default prompt is ER! unless the System Administrator has used SET_LSR_DEFAULTS to change the Login server *error_prompt* default.

–IMPLICIT_LOGIN

Allows users to log in to the system by typing their user ID at the login prompt and their password at the password prompt. This allows users to log in without typing the LOGIN command.

By default, users must type the LOGIN command unless the System Administrator enables implicit login, either using SET_LSR_DEFAULTS or this START_LSR –IMPLICIT_LOGIN option.

–MAXUSR_PROMPT *maxusr_prompt*

Sets the *maxusr_prompt*, which is the prompt displayed when users attempt to log in when the maximum number of users specified in the MAXUSR command has been exceeded. The default prompt is: `Please try again later.` unless the System Administrator has used SET_LSR_DEFAULTS to change the Login server *maxusr_prompt* default.

–READY_PROMPT *ready_prompt*

Sets the LOGIN_SERVER *ready_prompt*. The default prompt is `OK,` unless the System Administrator has used SET_LSR_DEFAULTS to change the Login server *ready_prompt* default.

All four configurable prompts (set using –PROMPT, –MAXUSR_PROMPT, –READY_PROMPT, and –ERROR_PROMPT) can use the RDY expandable prompt variables. These variables are listed in the description of the RDY command in the *PRIMOS Commands Reference Guide*. The maximum size of these prompts has increased at Rev. 23.3. The unexpanded prompt can be up to 80 characters long; after variable expansion, the prompt can be up to 256 characters long.

For further details on this command, refer to the *Operator's Guide to System Commands*.

SET_LSR_DEFAULTS Command

The SET_LSR_DEFAULTS command has been enhanced to permit you to display and modify default values for the new options provided in START_LSR.

For further details on this command, refer to the *Operator's Guide to System Commands*.

MAXUSR Command

The MAXUSR command has been modified to better inform users when they may log in. When the System Administrator specifies MAXUSR 0 (no logins permitted), the `Login Please` message is *not* sent to users' terminals following a user login attempt that failed because logins are not permitted. Instead the system displays the *maxusr_prompt* (described above) on the user's

terminal. The `Login Please` message is redisplayed on the user's terminal when the System Administrator resets MAXUSR to permit user logins. In prior Revisions, the `Login please` message was displayed regardless of whether MAXUSR permitted user logins.

Issuing a LOGOUT ALL command automatically sets MAXUSR to 0, and also removes the `Login Please` message from users' terminals.

For further details on this command, refer to the *Operator's Guide to System Commands*.

Logging LOGINs: a CONFIG_USERS Enhancement

At Rev. 23.3, the CONFIG_USERS utility has been enhanced to support the logging to DSM of successful and/or failed user LOGIN attempts. This security feature enables you to keep a record of all LOGINs performed on the system.

Note Similar information can be displayed on the supervisor terminal by setting the LOGMSG and LOGBAD configuration directives. Refer to the *System Administrator's Guide, Volume I: System Configuration* for details.

You establish LOGIN logging from the CONFIG_USERS System Operations screen:

```
                (Config Users - System)

        Rebuild SAD                              <  >
        List/Change System Defaults              <  >
        Change System Administrator              <  >
        Restore ACL Protection                   <  >
        List SAD Contents                        <  >
        List/Change Password Attributes          <  >
        List/Change User's Origin Attributes     <  >
        List/Change Security Features            <  >
        List/Change DSM Logging                  <  >

        Use Projects?                            (  )
        Use System Defaults?                     (  )
        Use Server (ISC) Attributes?             (  )

Enter=Confirm Cancel=ExitScrn F1=KeyHelp Help=FldHelp F2=Zoomin
```

To set DSM logging of LOGIN attempts, select `List/Change DSM Logging`, and press the ZoomIn key in this field. This displays the following screen:

```
                Log Failed Attempts to DSM?      ( )

                Log Successful Attempts to DSM?  ( )




Enter=Confirm Cancel=ExitScrn F1-KeyHelp Help=FldHelp F2-ZoomIn
```

Select either one or both of these fields by using the Enter key.

If you have specified DSM logging of LOGINs, DSM begins logging all user LOGINs of the specified type(s) under the product name LOGIN. The following are typical successful and failed login attempts, as recorded in the DSM log:

```
*** DSM_LOGGER (User 193 on EN.D44) at 12:49
12:49 Prime product LOGIN (EN.D44) Information
User JNS in project DEFAULT logged in as user number 2 on line number 0
User type is normal user


*** DSM_LOGGER (User 193 on EN.D44) at 12:49
12:49 Prime product LOGIN (EN.D44) Failure
Failed login attempt due to unknown error by user JNS in project <default>
on line number 0. User number is 2, user type is normal user
```

The type of login attempt is indicated by the severity code:

Successful LOGIN	DS$INFORMATION severity code
Failed LOGIN	DS$FAILURE severity code

The DSM entries contain the following information:

User ID	User ID of user attempting LOGIN. Non-printable characters are logged as @ characters.
Project ID	Project ID of user; NULL if default project. Non-printable characters are logged as @ characters.
Line Number	For remote LOGINs, either the NTS logical line number (currently 1024 or greater) or a remote line indicator. For local LOGINs, the physical line number.
User Number	User number of user attempting LOGIN
User Type	Normal, Remote, Slave, or NTS
Originating Network Address	PRIMENET node ID, X.25 address, Ethernet address, or LAN300 address/ port number
Status	0 if successful LOGIN, otherwise specifies reason for LOGIN failure.

When a user LOGIN fails, the system records the reason for the failure as a status code. The following are the available status codes:

0	Successful login
1	Bad project ID supplied
2	Bad password supplied
3	Unsuccessful password change
4	Concurrent login limit exceeded
5	Not enough segments available to complete login
6	Initial attach point unreachable
7	General validation error (failed attempt was trapped by LOGIN_SERVER)
8	Unknown error

For further information on CONFIG_USERS, refer to the *System Administrator's Guide, Volume III: System Access and Security*. For further information on DSM refer to the *DSM User's Guide*.

User ID Security Enhancement

At Rev. 23.3, if a user specifies a non-printing character (such as an escape sequence) in a user ID or project ID, PRIMOS converts that character to an @ character when it displays the invalid LOGIN at the system console. This feature provides additional security protection to prevent unauthorized users from sending escape sequences to the system console.

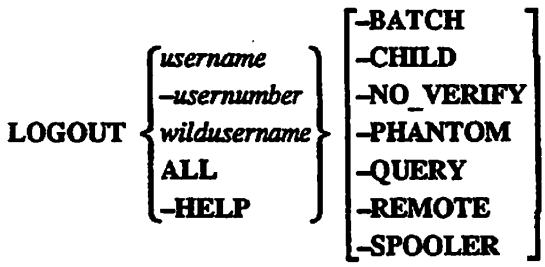
Logout Enhancements

System Administrators can log out groups of users by type or select users for logout using wildcards. At Rev. 23.3, you can configure the system to provide warning before the automatic logout (timeout) of an inactive user.

LOGOUT Command Enhancements

At Rev. 23.3, the LOGOUT command has been enhanced to allow greater selectivity in logging out processes. The System Administrator can log out users by name or user number, and can use wildcards to select multiple users. The System Administrator can also select classes of jobs to be logged out. A non-privileged user may select jobs with the same user name.

Note To log out any users other than yourself, you must issue this command from the supervisor terminal.



username Specifies the user name of the user you wish to log out. (You can use the STATUS USERS command to determine the names and numbers of all users.)

-usernumber

Specifies the decimal number of the user being logged out. (You can use the STATUS USERS or LIST_USERS commands to determine the names and numbers of all users.) If the user is a local terminal using a remote process, the terminal is logged out of both systems. If the user is a remote terminal using a local process, the process is logged out and returned to the pool of free remote login processes.

Users can use wildcards with their own user numbers to select processes to log out. The System Administrator can use wildcards on any user numbers to select processes to log out. For example,

```
OK, LOGOUT -15+
```

where + matches any one character and logs out all user numbers of the form 15*n*.

wildusername

Allows the System Administrator to use wildcarding to match and log out appropriate user processes except for the originating login. A System Administrator could use this argument when logging out all user processes on the system that have the same prefix or suffix. For example,

```
OK, LOGOUT CS101_@@ -NO_VERIFY
```

ALL

Logs out most currently logged-in local and phantom users. Also disconnects remote users from your system and returns them to their own. Many system phantoms, including LOGOUT_SERVER, LOGIN_SERVER, and TIMER_PROCESS, are not affected by LOGOUT ALL; such processes have their own shutdown procedures.

In addition to logging out all users except User 1 (the supervisor terminal), LOGOUT ALL automatically performs a MAXUSR 0 command, which prevents any subsequent logins until MAXUSR is reset. You should issue LOGOUT ALL a few minutes before you issue a SHUTDN ALL command in order to allow a more orderly shutdown of PRIMOS. (See the Note below.) LOGOUT ALL can be issued only from the

	supervisor terminal. See also the MAXUSR command.
–HELP	Displays command syntax and options.
–BATCH	Selects and logs out only matching Batch phantoms. This option can be used by non-privileged users.
–CHILD	Selects and logs out only matching child processes. This option can be used by non-privileged users.
–NO_VERIFY	Disables querying when you use wildcarding. This option can be used by non-privileged users.
–PHANTOM	Selects and logs out only matching phantoms. This option can be used by non-privileged users.
–QUERY	Queries you to be sure that you want to log the process out when you are *not* using wildcarding. You are queried by default when you use wildcarding. This option can be used by non-privileged users.
–REMOTE	Selects and logs out only matching remote processes. Remote processes can be of the following three types: LOG THROUGH: a user that has logged in remotely using the LOGIN –ON option. FROM REMOTE: a user that has used NETLINK to access this system from another machine. GONE REMOTE: a user that has used NETLINK to access another system from this machine. This option can be used by non-privileged users. Abbreviated –REM.
–SPOOLER	Selects and logs out only matching spooler processes. This option can be used by non-privileged users. Abbreviated –SPOOL.

Note Please update your Rev. 23.3 copy of the *Operator's Guide to System Commands* to include the –REMOTE, –SPOOLER, and –HELP options.

Forced Logout

When configuring users, the System Administrator can specify a forced logout time in the LOUTQM directive to force users to be logged out if there has been no activity at their terminals for longer than the allowable maximum inactive time limit. When this force logout occurs, the normal logout message is preceded by the following message:

```
***FROM PRIMOS: maximum inactive time limit exceeded.
```

If users are logged out by a command issued at the supervisor terminal, the message `forced logout.` appears, followed by the normal logout message.

Note PRIMOS allows a process between one and two minutes of grace time following a forced logout if the program being run by the process is set up that way. For example, Prime's EMACS text editor uses that grace time to automatically save any work the user has in progress. Also, the actual process of logging out may take some time if the system is heavily loaded, particularly if your system has an external logout program that performs any lengthy functions (such as checking for electronic mail, performing accounting, and so on). Therefore, allow at least three to four minutes for a forced logout to take effect before issuing the SHUTDN ALL command.

For further details on the LOGOUT command, refer to the *Operator's Guide to System Commands*.

LOUTQM Configuration Directive Enhancement

At Rev. 23.3, you can notify an inactive user terminal a specified number of minutes before the user is automatically logged out (inactivity timeout). You specify this notification time using the LOUTQM configuration directive, as follows:

```
LOUTQM inactivity_time notification_time
```

Both variables are a number of minutes, expressed in octal. In the following example, the *inactivity_time* is set to 60 minutes (74 octal), and the *notification_time* is set to 8 minutes (10 octal):

```
LOUTQM 74 10
```

In this case, the system displays a message on an inactive terminal 8 minutes (10 in octal) before the expiration of the 60 minute inactivity limit. After 52 minutes of inactivity, the system causes the inactive terminal to beep and to display the following message:

```
***From PRIMOS: Warning, inactivity timeout in 8 minutes.
```

If the user does not respond (by pressing any key) within the indicated time, the system logs out the user.

User Support Enhancements

HELP Facility Terminal Types Support

In order for the Help facility to work properly, you must correctly specify your display terminal type. At Rev 23.3, Help uses the TERMINFO database to define the terminal type parameters. Any terminal type defined for TERMINFO can take advantage of the screen interfaces to Help. This greatly expands the number of terminal types that can use the PRIMOS Help facility.

Help finds out what kind of terminal you have in one of two ways. If you specify the -TERMINAL_TYPE (-TTP) option, Help checks that the terminal type you specify is valid for TERMINFO. If it is, Help opens the Main Menu. If you do not use the -TTP option, and the global variable .TERMINAL_TYPE$ is set, Help checks that the terminal type in .TERMINAL_TYPE$ is valid for TERMINFO. If it is, Help opens the Main Menu.

If your terminal type is not defined for TERMINFO, you are unable to open the Main Menu. Help displays the following message:

```
HELP could not display the MAIN MENU using the specified terminal type.
Please specify the KEYWORD and all options from the PRIMOS command level.
For additional assistance, type HELP HELP at the PRIMOS command level.
[HELP Rev. 23.3.0 Copyright (c) 1992, Prime Computer, Inc.]
```

You still have access to all the same textual information but you must enter at least the command, topic, or search-string on the command line in order to display the first menu. Note that Help uses an ASCII screen format when TERMINFO cannot be used; therefore screen features such as reverse video are not available.

Defining a Terminal Type for HELP

You can create your own terminal type definition file in the TERMINFO database. TERMINFO is a database of compiled 'termcap-like' source files. The source files are not TERMCAP files proper, but contain a similar mapping of terminal display and input functions to character sequences.

TERMINFO terminal definition files are found in the TERM* directory. This directory contains the following subdirectories:

- Single-letter subdirectories, such as TERM*>P. Each of these subdirectories contains the compiled terminal definition files for terminals that begin with that letter. For example, TERM*>P contains the definition files for the PST100, PT200, and PT45 terminals; TERM*>V contains the definition file for the VT100 terminal. There is also a # directory for terminals whose names begin with a number.
- SOURCES subdirectory, which contains source code files used to create a compiled terminal definition file.
- RUN subdirectory, which contains tools used for compiling or decompiling terminal definition files. This subdirectory contains the following:
  - COMPILE.RUN, which you can use to compile a source definition file. COMPILE.RUN takes an ASCII file defining the function mappings, compiles it into a TERMINFO file, and places it in the proper single-letter TERM* subdirectory. Therefore one must have write access to the TERM* directory in order to use COMPILE.RUN.
  - DUMP.RUN, which you can use to print out an ASCII file of a compiled definition file, showing the entries for each terminal function defined.

To create your own terminal definition file, perform the following steps:

1. Search the appropriate single-letter subdirectory for your terminal type. If a compiled file exists, run DUMP.RUN to generate an ASCII copy of the file.
2. If no compiled file exists, create an ASCII file in the SOURCES subdirectory.
3. Using EMACS or ED, write or modify the ASCII file, including the sequences required to define the desired terminal functions. Use the existing files in SOURCES as examples. Your terminal user guide should provide a listing of the appropriate sequences for that terminal. You may also wish to consult a book on TERMINFO, such as *Termcap & Terminfo* by O'Reilly and Associates, Inc.
4. Compile the ASCII file using COMPILE.RUN. Support for this device is now available.

INITIALIZE_SEARCH_RULES Command

INITIALIZE_SEARCH_RULES (ISR) is a new operator command at Rev. 23.3. It reads the current system search rules from SEARCH_RULES*, parses them, loads them into main memory, and then directs all user access to system search rules to this location. This speeds subsequent user logins and ICE commands. Significant login CPU time is saved because the search rules do not have to be parsed. This command is most commonly issued as part of the PRIMOS.COMI startup file. INITIALIZE_SEARCH_RULES has the following format.

INITIALIZE_SEARCH_RULES [-OFF | -STATUS | -HELP]

Issuing the command with no options loads the system search rules into memory. The options perform the following operations.

-OFF

Turns off INITIALIZE_SEARCH_RULES. This causes all access to system search rules to revert from main memory to the SEARCH_RULES* directory.

-STATUS

Displays whether or not INITIALIZE_SEARCH_RULES is currently activated on the system. You activate INITIALIZE_SEARCH_RULES by invoking the command; you inactivate INITIALIZE_SEARCH_RULES by either invoking INITIALIZE_SEARCH_RULES -OFF or by coldstarting the system. The possible values for this option are active and inactive.

-HELP

Displays command syntax.

Once you have set INITIALIZE_SEARCH_RULES, users receive their default search rules from main memory, not from the SEARCH_RULES* directory. Therefore, after making changes to SEARCH_RULES*, you must run INITIALIZE_SEARCH_RULES before users initializing their command environment see the changes.

For more information on search rules, refer to the *Advanced Programmer's Guide II: File System* and the *System Administrator's Guide, Volume I: System Configuration.*

MESSAGE Command

At Rev. 23.3, the MESSAGE command allows the System Administrator or operator to define message attributes. You can define a maximum line length to be 80 or 512 characters. Also, you can specify whether a Carriage Return/Line Feed (CRLF) may be placed in an output message. Instructions for placing a CRLF in a message are provided in Chapter 2 of this *Software Release Document.*

If you want messages to be longer than 80 characters, ensure that the output buffers are large enough to accommodate the extra length. If the output buffers are not large enough, long messages will not be delivered and an error will be reported to the sender.

The following are the administration options for the MESSAGE command:

–MAX_LENGTH *value*

Sets the maximum number of characters allowed in a message. *value* must be either 80 or 512 characters. This option can be used only at the supervisor terminal. The default is 80 characters with no carriage return/line feed substitution. Abbreviated –ML.

–ENABLE_CRLF

Enables carriage return/line feed substitution in messages. This option can be used only at the supervisor terminal. Abbreviated –EC.

–DISABLE_CRLF

Disables carriage return/line feed substitution in messages. This option can be used only at the supervisor terminal. Abbreviated –DC.

–DISPLAY

Allows all users to display the current message settings (which are set by the operator). For example,

```
Maximum message length allowed: 80 characters.
CRLF substitutions: DISABLED
```

Abbreviated –DP.

–HELP

Lists command syntax.

For further details on this command, refer to the *Operator's Guide to System Commands.*

MGSET$ Subroutine Support: The MGSET$ subroutine supports these new message options with four new key values:

K$ENCR	Enable multiline messages (CRLF substitutions).
K$DSCR	Disable multiline messages (CRLF substitutions).
K$ENSH	Set maximum message length to short (80 characters).
K$ENLG	Set maximum message length to long (512 characters).

File System Support Enhancements

CNAME –FORCE Command

At Rev. 23.3, the CNAME command has a new option, –FORCE. This option allows you to change the name of a file system object that is currently in use. The only restriction is that the length of the source and target entrynames must be the same.

Caution Use this option carefully. Misusing it may cause unexpected results, such as the failure of a running program to locate essential directories or log files.

The following examples show valid and invalid uses of the –FORCE option.

The following two examples are valid because FOO and BAR are the same length, and the length of CDE equals the length of FGH:

```
OK, CNAME FOO BAR -FORCE -REPORT
"FOO" name changed to "BAR".
OK,


OK, CNAME A>B>CDE FGH -FORCE
OK,
```

The following two examples are invalid because the length of FOO is greater than the length of F and the length of CDE is greater than the length of AB:

```
OK, CNAME FOO F -FORCE
The source and target entrynames must be the same length
when using the -FORCE option. F (CNAME)
ER!


OK, CNAME A>B>CDE AB -FORCE
The source and target entrynames must be the same length
when using the -FORCE option. AB (CNAME)
ER!
```

The entryname lengths *must* be equal *only* when using –FORCE.

Note Do not change the names of special directories such as CMDNC0.

For further details on this command, refer to the *Operator's Guide to System Commands.*

Batch Subsystem Enhancements

At Rev. 23.3, the JOB command has been enhanced. The JOB command is used to monitor batch job queues.

- The JOB command can take a command line of up to 1024 characters. The previous maximum command line size was 160 characters.
- The JOB –COMO option can specify a pathname of up to 128 characters. This is the maximum length of the pathname *after* the expansion of *> into the pathname of the currently attached directory. The previous maximum pathname length was 64 characters.
- The JOB –STATUS and JOB –DISPLAY command options have additional syntax options that permit you to display batch job information for specific users or queues.

JOB –STATUS and JOB –DISPLAY

At Rev. 23.3, the JOB –STATUS command has the following syntax.

JOB [*job_id*] { **–STATUS** / **–ST** } [{ **ALL** / **TODAY** }] [**–USER** *user_id*] [**–QUEUE** *qname*]

At Rev. 23.3, the JOB –DISPLAY command has the following syntax.

JOB [*job_id*] { **–DISPLAY** / **–DP** } [{ **ALL** / **TODAY** }] [**–USER** *user_id*] [**–QUEUE** *qname*]

job_id

The job identification number.

ALL

Displays all jobs, regardless of state (including jobs in states other than waiting or executing). If you do not specify ALL or TODAY, the JOB command displays all waiting and executing jobs only.

TODAY

Displays all jobs submitted today, regardless of state (including jobs in states other than waiting or executing). If you do not specify TODAY or ALL, the JOB command displays all waiting and executing jobs only.

–USER *user_id*

Only displays jobs submitted by the specified user. If this option is omitted, these commands default to displaying all user jobs for Administrators. Users that are not Batch Administrators can only display their own jobs, unless the Batch subsystem was configured with –DISPLAY ALL.

–QUEUE *qname*

Only displays jobs in the *qname* job queue. Use the BATGEN –DISPLAY command to list the queue names and characteristics. If you do not select this option, jobs in all queues are displayed.

Normally –DISPLAY and –STATUS display all currently waiting and executing jobs in all queues. The modifiers, ALL and TODAY, show all jobs regardless of their state. Two options, –QUEUE *qname* and –USER *user_id*, limit which jobs are displayed.

You can combine the modifiers and options. If you use one of the modifiers (ALL or TODAY), you *must* specify it before any additional options.

Batch Queue Full

If a user submits a job to a full batch queue, Batch returns the new error message `Queue is full.`

Spooler Subsystem Enhancements

The following enhancements were made to the Spooler at Rev. 23.3.

Support for HP LaserJet Printers

The HP LaserJet family of printers is supported by a new DEVICE directive, and new attributes for setting page orientation, paper size and font size.

To use these features, the Spool Administrator needs to create (or modify) a printer environment file, which must include the following entries:

DEVICE HPJET [–DELAY *nnn*]

This option defines the printer as a HP LaserJet printer. The optional –DELAY parameter specifies how many seconds to wait between print jobs. You can adjust the –DELAY option to prevent buffer overflows at the printer. The default setting for –DELAY is 4 seconds. You can set –DELAY to values from 0 through 999 seconds, depending on the particular printer.

SET_PORTRAIT and/or SET_LANDSCAPE

These environment file directives specify the available orientations of text on the page. You must include both of these directives if you wish to make available both landscape and portrait page formatting.

ATTRIBUTE *name*

You must include one or more ATTRIBUTE directives in the environment file to define the paper size and fonts available for your HP LaserJet. The available *name* values are shown in the following tables.

Attributes for U.S. Paper Sizes			*Portrait*		*Landscape*	
Name	*Paper Size*	*Chars per Inch*	*Chars per Line*	*Lines per Page*	*Chars per Line*	*Lines per Page*
LETTER	8.5" x 11"	10	80	60	106	46
LETTER_MED	8.5" x 11"	12	96	82	127	62
LETTER_SMALL	8.5" x 11"	16.66	132	82	176	62
LEGAL	11" x 14"	10	80	78	134	46
LEGAL_MED	11" x 14"	12	96	105	163	62
LEGAL_SMALL	11" x 14"	16.66	132	105	226	62

Attributes for European Paper Sizes			*Portrait*		*Landscape*	
Name	*Paper Size*	*Chars per Centimeter*	*Chars per Line*	*Lines per Page*	*Chars per Line*	*Lines per Page*
A4	210x297mm	3.90	77	64	112	44
A4_MED	210x297mm	4.68	93	87	135	59
A4_SMALL	210x297mm	6.50	129	87	188	59
B5	182x257mm	3.90	66	54	97	37
B5_MED	182x257mm	4.68	80	74	116	50
B5_SMALL	182x257mm	6.50	111	74	167	50

Users specify the same *name* attributes when using the SPOOL command, for example:

```
SPOOL -ATTRIBUTE LETTER_MED
```

PostScript Formatting Defaults

Two new DEVICE directive options permit the environment file to control PostScript formatting defaults.

If you specify DEVICE POSTSCRIPT –POSTSCRIPT_CHECK in the environment file, the Spooler checks line 1 of each spooled print file for the standard PostScript header: %!PS.

- If the PostScript header is present, the job is automatically spooled with PostScript file formatting, regardless of whether or not you specified SPOOL –NO_FORMAT.
- If the PostScript header is absent, and you specified SPOOL –NO_FORMAT, the job is automatically spooled with PostScript file formatting, even though it is not a PostScript file.
- If the PostScript header is absent, and you did not specify SPOOL –NO_FORMAT, the job is spooled with standard file formatting.

If you specify DEVICE POSTSCRIPT –POSTSCRIPT_OVERRIDE in the environment file, the Spooler checks line 1 of each spooled print file for the standard PostScript header: %!PS.

- If the PostScript header is present, the job is automatically spooled with PostScript file formatting, regardless of whether or not you specified SPOOL –NO_FORMAT.
- If the PostScript header is absent, the job is spooled with standard file formatting, regardless of whether or not you specified SPOOL –NO_FORMAT.

Aliases for Mandatory Printer Environment Attributes

Up to ten aliases can be listed within each printer environment file for MANDATORY attributes. You can specify multiple –MANDATORY attribute lines, and you can specify more than one alias on each attribute line, as shown in the following example:

```
ATTRIBUTE alias1 alias2 alias3 -MANDATORY
ATTRIBUTE alias4 -MANDATORY
```

A user can specify any of these aliases when using the SPOOL command to spool a file for printing.

Printer Error Notification

A new environment directive, ERROR_NOTIFY, directs the spooler to send printer error messages to the user names listed in the ERROR_NOTIFY directive (up to eight users), on both local and remote systems. This environment directive has the following format:

```
ERROR_NOTIFY user1 user2 user3
```

Remote users can be specified as *username@systemname*. In addition to notifying the named users, the printer error message is always sent to the system console. This directive is for use only with printers connected over ASYNC and TCP/IP.

User Access to Printer Control

Spool Administrators can now give access rights to specific users to use those PROP command options that control printer environments. (Other PROP command options are not user-accessible.) The Spool Administrator does this by changing the ACL on the environment file to give the user a minimum of LXR access.

The following PROP command options can be made user-accessible: –ABORT, –BACK, –CONTINUE, –DROP, –HANG, –LINEUP, –RELEASE, –RESET, –RESTART, –START, –STOP, and –SUSPEND.

Size of Print Job

Two new options to the PROP command, –MAX_SIZE and –MIN_SIZE, enable Spool Administrators to control the size of print jobs sent to a printer.

PROP –MAX_SIZE [*nnnn*]

Sets the maximum disk record size of files entered into the local spool queue. Spooled files with record sizes greater than the maximum set by –MAX_SIZE are rejected by the spool queue. *nnnn* is the number of records. If you omit –MAX_SIZE, there is no limit on the file size. Only members of the group .SPOOL_ADMINISTRATOR$ are allowed to use this option. The default maximum size is zero (no maximum size limit). You can reset the maximum size to zero by entering 0 or nothing on the command line.

PROP –MIN_SIZE [*nnnn*]

Sets the minimum disk record size of files entered into the local spool queue. Spooled files with record sizes less than the minimum set by –MIN_SIZE are rejected by the spool queue. *nnnn* is the number of records. If you omit –MIN_SIZE, the default value of 0 (no minimum size limit) is used. Only members of the group .SPOOL_ADMINISTRATOR$ are allowed to use this option. The default minimum size is zero. You can reset the minimum size to zero by entering 0 or nothing on the command line.

Printing Over TCP/IP

The despoolers now utilize large packet sizes for TCP/IP instead of 1 line per packet. Although this feature is not user-visible, you should see a decrease in the use of TCP/IP packets.

The TCP/IP environment directive has two new options, as follows:

–RELEASE

Releases the socket after processing each printing job. This option alternates the processing of printing jobs between two environments that share the same printer. When one environment has finished printing a job, it releases control of the printer to the alternate environment for processing print jobs.

–DEFER *n*

Allows you to change the number of minutes that a deferred print job must wait before processing. If processing is attempted while the socket is busy, the status of the print job is automatically changed to deferred, with a default wait time of 5 minutes before the print job can try again. You can set *n* to fewer or more than the default value of 5 minutes.

Restarting the Spooler Subsystem

The PROP –COLDSTART command has a new suboption, –NO_QUERY. –NO_QUERY bypasses the query message response that is normally required before shutting down each printer environment when you restart the Spooler subsystem.

Caution The PROP –COLDSTART –NO_QUERY command deletes all queued or printing jobs on the system without further operator intervention. Use with extreme caution.

Listing Spooler Processes

At Rev. 23.3, there are two ways to list printer environments:

- LIST_USERS –SPOOLER, described in Chapter 2 of this document.
- STATUS USERS, which has been enhanced to label all spooler processes with the printer environment name in the `User` column, and the word `spool` in the `Line No.` column.

User-written External Login Programs

At Rev. 23.3, spooler processes are spawned using a new user type (U$SPLR). The addition of this new user type may cause some user-written external login programs to fail when they attempt to start a spooler environment using PROP –START. Recompiling these programs should resolve this problem.

For more detailed information about the Spooler subsystem, see the *Operator's Guide to the Spooler Subsystem.*

CDD –INFO Command Options

The CDD command provided at Rev. 23.2 permits you to activate a crash dump disk. At Rev. 23.3, the CDD –INFO options provide detailed online information to assist you in creating an appropriate crash dump disk. The CDD –INFO options provide the following information:

- The anticipated size of a full or partial crash dump
  - on the current system
  - on another system of known memory size
- The number of records to specify to the MAKE –SPLIT command to optimally allocate a crash dump disk:
  - the recommended split for the specified crash dump size and disk type
  - a table of all available splits for the specified disk type

CDD –INFO uses a series of screen displays to provide this information. There are four versions of CDD –INFO, each with a separate command line syntax, as follows:

CDD –INFO

The basic command to use when you want to size a crash dump disk for the current system.

CDD –INFO –MEMORY_SIZE

The basic command to use when you want to size a crash dump disk for another system.

CDD –INFO –DUMP_SIZE

Additional crash dump disk sizing information for a known number of records.

CDD –INFO –DUMP_SIZE_TABLE

Additional crash dump disk sizing information providing a list of possible sizes.

The following sections describe each of these options.

Note To display the CDD –INFO syntax online, specify CDD –HELP –INFO. To display all of the CDD syntax, including the CDD –INFO syntax, specify CDD –HELP.

CDD –INFO

This command provides the optimal size for a crash dump disk on the current system. Use this command before using MAKE to create a crash dump disk for your system. When you issue this basic CDD –INFO command, CDD calculates the anticipated crash dump sizes for your system, then displays the following screen:

```
This system has 16 MB of core memory.  Expected total sizes for
full and partial dumps are made up as follows:


                          FULL DUMP          PARTIAL DUMP
                        --------------      ----------------

     CORE memory dump :   8192 records      5734 records (approx)

     MAPS dump        :     41 records       41 records

     Safety margin    :    100 records      100 records

                        ----------------    ----------------

     TOTAL DUMP SIZE :   8333 records      5875 records (approx)
                        ================    ================

For MAKE recommendations, please specify the disk you intend to use
for CDD.
Enter "H" for help, or "Q" to quit.

Enter <pdev> or disk name:
```

The safety margin shown on this screen is usually 100 records, which provides some margin for disk write errors. However, CDD –INFO may increase this safety margin to provide a total dump size of at least 2000 records. 2000 records is considered to be the smallest viable crash dump size.

Note If your system is a 6000 Series computer, this screen also displays the number of PIOS dump records.

After viewing this screen, you identify the disk you intend to use to hold these records, as follows.

```
Enter <pdev> or disk name: 160765
```

You may specify the disk that you intend to activate as a crash dump disk either by its physical device (pdev) number (as shown above), or by its name. You can only specify a disk name if the disk has already been made and split (using MAKE –SPLIT) and the file system portion of the split disk has been ADDed.

After specifying the disk, CDD –INFO prompts you to specify the disk type, as follows:

```
Please specify a MAKE-compatible disk type for disk
160765.
Enter disk type (e.g., "Model_4729"): model_4729
```

Note You can avoid these prompts by specifying the pdev and/or the disk type as part of the command line, as follows:

OK, CDD 160765 –INFO –DT model_4729

After CDD –INFO displays the first screen, press Return to advance to the second screen. If the specified pdev and disk type are incompatible, CDD –INFO returns an error message and prompts you to respecify the pdev or disk type.

You can specify a partition name, rather than a pdev, if the disk has been split and ADDed. If you specify the pdev of a disk that is added or is a paging disk or an activated crash disk, CDD– INFO automatically supplies the correct disk type and does not prompt you for it. If you wish to specify a different disk type, specify it on the command line.

You specify the model of disk that you intend to activate as a crash dump disk. CDD uses the same disk model names as MAKE, but supports only those disk models that may be connected to a IDC1 or 7210 controller.

For a list of permissible disk model names supported by CDD, press Return at the prompt. For a more detailed list of disk models supported by CDD, enter D (or DETAIL) at the prompt. Enter H at the prompt to display the Help screen. For crash dump disk controller and disk drive model restrictions, refer to the *Rev. 23.2 Software Release Document.*

When you specify a disk drive model, the system displays a screen such as the following:

```
The crash disk you have specified has the following characteristics:

   Disk 160765      :   3 heads, at offset 28 (ctlr '23, unit 2)
   Disk model       :   MODEL_4729
   Total disk size  :   31242 records

To MAKE this disk with the maximum possible crash dump capacity:

   MAKE disk with   :     -SPLIT 30989       (see note 1 below)
   Maximum dump size:     30988 records      (see note 2 below)

For this disk to accommodate a full or partial dump of the size
predicted  earlier, the smallest -SPLIT value you can specify to MAKE is:

                          FULL DUMP          PARTIAL DUMP
                        ---------------  ----------------
   MAKE disk with   :     -SPLIT 8382         -SPLIT 6096
   Maximum dump size:    (8381 records)      (6095 records)
```

To view the explanatory notes for this screen, press the Return key. The following screen is displayed:

```
NOTE 1:   Maximum Dump Capacity and Optimal -SPLIT values

CDD requires a split disk, whose file system (FS) is a certain
minimum size: in this case, 9 records.  You must MAKE this disk with
a -SPLIT no bigger than S = total disk size - 9.  The display shows
an optimal -SPLIT, usually less than S, which balances maximum dump
capacity with maximum FS size.  (All displayed -SPLIT values are
optimal in this way.)

NOTE 2:   Derivation of -SPLIT value and Maximum Dump Size:

The -SPLIT value (S) needed for an arbitrary dump size (D) is
normally slightly bigger than D.  For reasons to do with dump
performance, S must be rounded up to a value close to an exact
number of tracks or cylinders.
```

If the disk you specified is too small for a full crash dump, a screen such as the following is displayed:

```
The crash disk you have specified has the following characteristics:

 Disk 460          :     2 heads, at offset 0  (ctlr '26, unit 0)
 Disk model        :     MODEL_4730
 Total disk size   :     7112 records

To MAKE this disk with the maximum possible crash dump capacity:


    MAKE disk with   :     -SPLIT 6859       (see note 1 below)
    Maximum dump size:     6858 records      (see note 2 below)

************* This disk is TOO SMALL for a full dump. **************

For this disk to accommodate a partial dump of the size predicted
earlier, the smallest -SPLIT value you can specify to MAKE is:

                          FULL DUMP           PARTIAL DUMP
                       ----------------     ----------------
    MAKE disk with   :   ** TOO SMALL **       -SPLIT 6096
    Maximum dump size:                        (6095 records)
```

CDD –INFO –MEMORY_SIZE

This command provides the optimal size for a crash dump disk on a system other than the current system. You must know the memory size of the other system. To invoke this command, issue a command line such as the following:

```
OK, CDD -INFO -MS 64 -CPU 6650
```

You use the –MEMORY_SIZE (–MS) suboption to specify the total core memory on the system to be calculated, expressed in megabytes. Use the –CPU suboption to specify the model number of the system to be calculated. You can omit the –CPU suboption if the model is not a 6000 Series computer. This displays a dump size screen such as the following.

```
For a system with 64 MB of core memory, expected total sizes for full
and partial dumps are made up as follows:

                          FULL DUMP          PARTIAL DUMP
                        ---------------     ---------------
    CORE memory dump :     32768 records    16384 records (approx)
    MAPS dump        :        41 records       41 records
    PIOS dump        :        65 records       65 records
    Safety margin    :       100 records      100 records
                        ---------------     ---------------
    TOTAL DUMP SIZE  :     32974 records    16590 records (approx)
                        ===============     ===============

 For MAKE recommendations, please specify the disk you intend to use
 for CDD.
 Enter "H" for help, or "Q" to quit.

 Enter <pdev> or disk name:
```

From this point you follow the same procedures as with a standard CDD –INFO command.

CDD –INFO –DUMP_SIZE

You use the CDD –INFO –DUMP_SIZE suboption when you wish to override the dump size recommended by the basic CDD –INFO command and determine a split value for a known number of records. Under normal circumstances, you should use the CDD –INFO command recommendation. This –DUMP_SIZE suboption is only used when it is necessary to create a crash disk larger or smaller than CDD –INFO recommends. In this case, you issue a command line such as the following:

```
OK, CDD 160765 -INFO -DT MODEL_4845 -DS 20000
```

You use the –DUMP_SIZE (–DS) suboption to specify the desired number of records. This example also specifies the pdev number (160765) and the disk type (–DT MODEL_4845) on the command line. If you omit these options, CDD prompts you for the appropriate values. Issuing this command line skips the full and partial dump screen and displays the split recommendation for the specified number of records on the specified disk, as shown in the following screen:

```
The crash disk you have specified has the following characteristics:

     Disk 41460       :   6 heads, at offset 8  (ctlr '26, unit 0)
     Disk model       :   MODEL_4845
     Total disk size  :   96672 records
     Restrictions     :   Must MAKE with -DBS ON (-IC) for CDD use

 To MAKE this disk with the maximum possible crash dump capacity:

     MAKE disk with   :      -SPLIT 96559       (see note 1 below)
     Maximum dump size:      96558 records      (see note 2 below)

 For this disk to accommodate a dump of total size 20000 records,
 the smallest -SPLIT value you can specify to MAKE is:


      MAKE disk with   :      -SPLIT 20064
      Maximum dump size:      20063 records   (see note 2 below)
```

After displaying this screen, you can optionally display the Notes screen for additional information.

CDD –INFO –DUMP_SIZE_TABLE

You use the CDD –INFO –DUMP_SIZE_TABLE (–DST) suboption to display a table of the optimal split values for your disk. This suboption is only used under unusual circumstances when it is necessary to fine tune the size of a crash dump disk. Normally, you should use the crash dump disk size recommended by the basic CDD –INFO command.

One use of this suboption is to select a dump size one size smaller to avoid having to use a crash dump disk one surface larger. Because CDD can only use a dump space that comprises an exact number of tracks or cylinders, you should never specify a split value not shown on this table. Using this table, you can select the best split value for dividing the disk space between crash disk space and file system disk space.

The following example lists all of the available split values for the specified disk. You may want to limit the display to begin at a specific minimum split value and/or to proceed by a specified increment. These options are described later in this section.

```
OK, CDD 160765 -INFO -DT MODEL_4729 -DST
The crash disk you have specified has the following
characteristics:
   Disk 160765     :     3 heads, at offset 28 (ctlr '23, unit 2)
   Disk model      :     MODEL_4729
   Total disk size :     31242 records

To MAKE this disk with the maximum possible crash dump capacity:

   MAKE disk with    :      -SPLIT 30989        (see note 1 below)
      Maximum dump size:    30988 records       (see note 2 below)

  DUMP SIZE TABLE:
For this disk, optimal splits are those for which either the
maximum dump size (MDS) or the -SPLIT value (S) is an exact
multiple of 254 records, and S = MDS + 1.  Below is a table of
optimal -SPLIT values, beginning from the dump size closest to
the supported minimum of 2000 records:

MAKE with -SPLIT  2032 for a maximum dump size of   2031 records
MAKE with -SPLIT  2286 for a maximum dump size of   2285 records
MAKE with -SPLIT  2540 for a maximum dump size of   2539 records
MAKE with -SPLIT  2794 for a maximum dump size of   2793 records
--More--
MAKE with -SPLIT  3048 for a maximum dump size of   3047 records
MAKE with -SPLIT  3302 for a maximum dump size of   3301 records
MAKE with -SPLIT  3556 for a maximum dump size of   3555 records
MAKE with -SPLIT  3810 for a maximum dump size of   3809 records
MAKE with -SPLIT  4064 for a maximum dump size of   4063 records
MAKE with -SPLIT  4318 for a maximum dump size of   4317 records
MAKE with -SPLIT  4572 for a maximum dump size of   4571 records
MAKE with -SPLIT  4826 for a maximum dump size of   4825 records
MAKE with -SPLIT  5080 for a maximum dump size of   5079 records
MAKE with -SPLIT  5334 for a maximum dump size of   5333 records
MAKE with -SPLIT  5588 for a maximum dump size of   5587 records
MAKE with -SPLIT  5842 for a maximum dump size of   5841 records
MAKE with -SPLIT  6096 for a maximum dump size of   6095 records
MAKE with -SPLIT  6350 for a maximum dump size of   6349 records
MAKE with -SPLIT  6604 for a maximum dump size of   6603 records
MAKE with -SPLIT  6858 for a maximum dump size of   6857 records
MAKE with -SPLIT  7112 for a maximum dump size of   7111 records
MAKE with -SPLIT  7366 for a maximum dump size of   7365 records
MAKE with -SPLIT  7620 for a maximum dump size of   7619 records
MAKE with -SPLIT  7874 for a maximum dump size of   7873 records
MAKE with -SPLIT  8128 for a maximum dump size of   8127 records
MAKE with -SPLIT  8382 for a maximum dump size of   8381 records
MAKE with -SPLIT  8636 for a maximum dump size of   8635 records
--More--q
-------------------------------------------------------------------

Type <return> for explanatory notes, or "Q" to quit: q
```

Start and Step: You can limit the dump size table display to the values of interest to you. To limit the listing, use the following syntax:

```
-DST start step
```

Specifying an optional *start* variable lists all split values greater than or equal to that starting value. The optional *step* variable lists only those split values separated by the closest increment greater than or equal to the *step* value. You must specify a *start* to specify a *step* value. These variables are shown in the following example:

```
OK, CDD 160765 -I -DT MODEL_4729 -DST 20000 1000

  The crash disk you have specified has the following
characteristics:

    Disk 160765     :   3 heads, at offset 28 (ctlr '23, unit 2)
     Disk model     :    MODEL_4729
    Total disk size :   31242 records


To MAKE this disk with the maximum possible crash dump capacity:


MAKE disk with   :     -SPLIT 30989          (see note 1 below)
Maximum dump size:     30988 records         (see note 2 below)


  DUMP SIZE TABLE:


For this disk, optimal splits are those for which either the
maximum dump size (MDS) or the -SPLIT value (S) is an exact
multiple of 254 records, and S = MDS + 1.  Below is a table of
optimal -SPLIT values, beginning from the dump size closest to
20000 records, and approx 1000 apart:

MAKE with -SPLIT 20066 for a maximum dump size of  20065 records
MAKE with -SPLIT 21082 for a maximum dump size of  21081 records
MAKE with -SPLIT 22098 for a maximum dump size of  22097 records
MAKE with -SPLIT 23114 for a maximum dump size of  23113 records
--More--
MAKE with -SPLIT 24130 for a maximum dump size of  24129 record
MAKE with -SPLIT 25146 for a maximum dump size of  25145 records
MAKE with -SPLIT 26162 for a maximum dump size of  26161 records
MAKE with -SPLIT 27178 for a maximum dump size of 27177 records
MAKE with -SPLIT 28194 for a maximum dump size of  28193 records
MAKE with -SPLIT 29210 for a maximum dump size of  29209 record
MAKE with -SPLIT 30226 for a maximum dump size of  30225 records
MAKE with -SPLIT 30989 for a maximum dump size of  30988 records
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
End of table: preceding line represents maximum capacity of disk
-------------------------------------------------------------------

Type <return> for explanatory notes, or "Q" to quit: Q
OK,
```

The above example specifies a *start* of 20000 records; therefore, it begins listing split values with the next largest value, 20066. It specifies a *step* of 1000 records; therefore, each successive split value is at least 1000 records larger than the preceding one.

MAKE and FIX_DISK Command Enhancements

At Rev. 23.3, the MAKE command and the FIX_DISK command have been enhanced to provide the user with more guidance when making or fixing partitions on disk types that support Dynamic Badspot handling (DBS). Other aspects of the user interface have also been enhanced to make these commands easier to use.

Caution The Rev. 23.3 versions of MAKE, FIX_DISK and ADD_DISK must only be used with Rev. 23.3 versions of the other two commands. For example, a disk created using Rev. 23.3 MAKE can only be fixed using the Rev. 23.3 version of FIX_DISK. Make sure to delete any pre-Rev. 23.3 versions of these commands. Disks created prior to Rev. 23.3 do not have to be modified.

MAKE Command

New Options: The following new MAKE command options have been introduced:

–USAGE

Prints a brief list of available options.

–LIST_BADSPOTS

At the end of the MAKE, prints a list of all known badspots. The badspots are sorted by head number, so that all for each partition are grouped together.

–NO_QUERY

Facilitates execution of MAKE by a phantom.

–FORMAT_OK

Like –FORMAT, but does not format unless MAKE determines this is necessary; intended for use with –NO_QUERY.

Renamed Options: Some old command line option names have been replaced by new option names, as listed below. The operation of these options has also changed considerably. The old option names are still supported as aliases, but their continued use is discouraged.

Obsolete Option Names	*New Equivalent Option Names*
–INTELLIGENT_CONTROLLER (–IC)	–DBS ON
–ALL_CONTROLLER (–AC)	–DBS OFF
–COPY_BADSPOTS_BY_DEVICE	–COPY_BADSPOTS *pdev*
–COPY_BADSPOTS_BY_NAME	–COPY_BADSPOTS *diskname*
–OVERRIDE_DEFAULT_INTERLEAVE	–SECTOR FORWARD
–RESTORE_DEFAULT_INTERLEAVE	–SECTOR REVERSE

Changes to Existing Options: The following changes have been made to existing options:

MAKE may now be invoked with no options. As previously, the options –DISK, –PARTITION, and –DISK_TYPE are always required. MAKE will always prompt for these options if they are not supplied, or are supplied with missing or invalid arguments.

For all other options which expect arguments, MAKE will prompt only if the option was supplied with a missing or invalid argument. In this respect the only change from previous revisions is that the order of the prompts has been changed for improved validation. The new order is: –DISK, –PARTITION, –DISK_TYPE, –DISK_REVISION, –SECTOR, –MIN_EXTENT_SIZE, –MAX_EXTENT_SIZE, –BADLEV, –COPY_BADSPOTS, –BAUD, –SPLIT.

The –DBS option is an exception to the preceding paragraph. –DBS ON (or –IC) will be ignored with disk types or controllers that do not support dynamic badspot handling. Where dynamic badspot handling is supported, MAKE will prompt for –DBS (or query the supplied –DBS option) before or after –SPLIT, according to circumstances.

At the prompt for –DISK_TYPE, press Return. This displays a list of supported disk types. The list has been enhanced to include other information sometimes found useful when making disks.

The –SPLIT option now supports the argument MAX, which reserves the maximum space possible for paging or CDD. When use of –SPLIT results in a prompt, MAKE now explicitly displays the maximum –SPLIT value you may enter. It is no longer necessary to estimate the maximum –SPLIT value from a display of total disk size, and no longer possible to enter a value that is too large and causes MAKE to abort with the message:

```
Insufficient room for minimum file system. Leave at least
N records.
```

The –HELP display, which contains descriptions of all options, has been slightly expanded. The new option –USAGE prints a brief list of all options.

A more consistent set of option abbreviations has been introduced. All options now have a three-letter abbreviation (e.g., –DSK, –FMT, –RPT). All old option abbreviations are still supported.

Further information about these option changes can be found in the MAKE –HELP and HELP MAKE online displays, and in the *Operator's Guide to File System Maintenance.*

Dynamic Badspot Handling Changes: New messages, warnings, and prompts may occur when making Revision 21 or later format partitions on these disk types: SMD, 68MB, 158MB, 160MB, 600MB, MODEL_4475, MODEL_4735, MODEL_4845, and MODEL_4860. The aim of these changes is to clarify and to more strongly enforce or recommend, as appropriate, the following rules for making partitions on a disk type that supports dynamic badspot handling:

1. Dynamic badspot handling is enabled on a spindle as a whole by making the first, or head zero, partition with –DBS ON. The head zero partition contains the DBS file that serves the entire spindle; this partition should therefore be made before any other partitions on the spindle.
2. After making the head zero partition with –DBS ON, you should assign it, if possible, before making any non-head-zero partitions on the spindle. This is because the DBS file on the head zero partition contains information that is required to MAKE any other partition on the spindle.
3. You must make non-head-zero partitions in the same DBS mode as the head zero partition. If the head zero partition is assigned, you do not have to specify a DBS mode for the non-head-zero partitions; these partitions will be made with the same mode as the head zero partition. In circumstances where it is not possible to assign the head zero partition, you must specify the same DBS mode (i.e., –DBS ON) for each non-head-zero partition that you specified for the head zero partition.
4. Whenever dynamic badspot handling is enabled on a spindle and the head zero partition is remade in a way that requires an active DBS file to be rebuilt or removed, all other partitions on the spindle may need to be remade with –FORMAT. MAKE will now print explicit warnings if such an operation risks loss of data on the other partitions, and, unless –NO_QUERY is specified, prompt for permission to continue.

Caution Failure to remake these partitions with –FORMAT when suggested may result in inconsistent behavior during subsequent use of the partitions by PRIMOS.

---

5. When making a non-head-zero partition with –FORMAT, it is especially important that the head zero partition be assigned. Failure to comply with this recommendation may result in inconsistent behavior during subsequent use of the partitions by PRIMOS.

For disk types that do not support dynamic badspot handling (i.e., not among the models listed earlier), the –DBS option is irrelevant and there is no need to assign the head partition when making other partitions.

Effect on CPL Programs and COMI Files: The changes described above mean that MAKE may now print new prompts, and some existing prompts may appear in a different order. CPL programs that anticipate the order of prompts and supply the answers in a &DATA block should be amended to use command line options to supply all the information that MAKE needs. COMI files that do the equivalent of this should be amended likewise.

FIX_DISK Command

Some old command line option names have been replaced by new option names, as listed below. The operation of these options has also changed considerably. The old option names are still supported as aliases, but their continued use is discouraged.

Obsolete Option Names	*New Equivalent Option Names*
–INTELLIGENT_CONTROLLER (–IC)	–DBS ON
–ALL_CONTROLLER (–AC)	–DBS OFF
–OVERRIDE_DEFAULT_INTERLEAVE	–SECTOR FORWARD
–RESTORE_DEFAULT_INTERLEAVE	–SECTOR REVERSE

–DBS ON

Selects Dynamic Badspot Handling mode for this partition on a spindle connected to a Model 6580 (IDC1) controller, allowing dynamic badspot handling and mirroring to take place. This option is recommended when you convert a partition to Rev. 21 format with the –CONVERT_21 option. You must use the –FIX option with –DBS OFF.

SCSI disk drives (Models 4721, 4729, 4730, 4731, and 4732) on a Model 7210 controller do not have a controller mode, and therefore ignore this option.

–DBS OFF

Selects non-Dynamic Badspot Handling mode for this Rev. 21 format or Rev. 22.1 format partition. This option, or the –DBS OFF option, is recommended

when you convert to Rev. 21 format with the –CONVERT_21 option. You must use the –FIX option with –DBS ON.

SCSI disk drives (Models 4721, 4729, 4730, 4731, and 4732) on a Model 7210 controller do not have a controller mode, and therefore ignore this option.

–SECTOR FORWARD

Sets the file record allocation direction to forward and the interleave factor to 3. This option can only be used on Rev. 20 and later formatted partitions. Because robust partitions always allocate records forward, this option is meaningless on such a partition. You must use the –FIX option with –SECTOR FORWARD

SCSI disk drives (Models 4721, 4729, 4730, 4731, and 4732) on a Model 7210 controller are automatically formatted with forward sectoring, and therefore ignore this option.

–SECTOR REVERSE

Sets the file record allocation direction to reverse and the interleave factor to 1 on standard partitions. On robust partitions this option is ignored, because robust partitions always allocate records forward. This option is valid only on partitions of Rev. 20 or later format. You must use the –FIX option with –SECTOR REVERSE.

SCSI disk drives (Models 4721, 4729, 4730, 4731, and 4732) on a Model 7210 controller are automatically formatted with forward sectoring, and therefore ignore this option.

FS_RECOVER Enhancements

Rev. 23.3 requires a new version of FS_RECOVER (Version 4), which is supplied with the operating system revision as an optional utility. Earlier versions of FS_RECOVER are not compatible with Rev. 23.3. FS_RECOVER Version 4 is compatible with operating systems back to Rev. 21.0

The following are changes from the previous version of FS_RECOVER:

- When invoked, FS_RECOVER checks for the existence of the Disk Manager's mailbox: <0>SYSTEM_DEBUG*>DISK_MANAGER.MBX. FS_RECOVER aborts with a system console message if this file does not exist.
- Crash recovery analysis provides you with a FIX_DISK recommendation. If you do not take this recommendation, you can directly specify the kind of FIX_DISK processing you do wish, using the keywords FULL, FAST, PARTIAL, or NONE. This keyword option replaces the FIX_DISK Menu, which is shown on page 3–11 of *Using FS_RECOVER*.

- The DISK_MANAGER program services two additional commands for FIX_DISK phantoms: MIRROR_ON and MIRROR_OFF.
- When an automated FIX_DISK phantom completes, it checks for a DSKRAT OK message from FIX_DISK indicating successful disk recovery. If FIX_DISK has not issued this message, the phantom sends a warning message to the system console stating that the partition has been partially recovered. (This functionality is restricted to systems that support 32 IX mode addressing; these include all systems with 4-digit designations except the Model 2250.)
- When you invoke INIT_RECOVER.CPL –AUTO_ANALYSIS with a CDD_AUTO_CONFIG file, a crash dump disk may be mounted. This removes the restriction previously noted on page 2–4 of *Using FS_RECOVER*. This enhancement benefits systems that have a crash dump disk partition on a split disk in which the file system portion contains data.
- INIT_RECOVER.CPL can now be used to send mail to users informing them when the system was restarted. For further information on this feature, read the comments found in the header to INIT_RECOVER.CPL.
- After displaying the state of currently mounted disks (Main Menu option 3), you can run FIX_DISK on any partitions that you select, whether or not the partitions are flagged as damaged. Hitherto, your options were either to run FIX_DISK on all partitions or only on partitions flagged as damaged.
- The crash recovery analysis Summary Display always lists the FS_RECOVER recommendations for FIX_DISK, not the type of recovery selected by the System Administrator.

Other Disk Support Enhancements

PDEV Command

Rev. 23.3 includes the new PDEV command. PDEV interconverts pdevs (physical device numbers) and their decoded values.

A pdev (physical device number) is a unique number generated from four values that specify a disk drive: controller address; unit number; starting head; and number of heads.

PDEV does the following:

- Displays the controller address, disk drive unit number, starting head (or surface), and number of heads of a given pdev.

- Displays the ldev, pdev, controller address, disk drive unit number, number of the starting head, and number of heads of one or more local disks.
- Converts a given controller address, disk drive unit number, starting head, and number of heads into a pdev.

Note PDEV checks whether the pdev you encode is a valid pdev value, but does not check whether the pdev corresponds to an actual disk drive. You should check the four values you use to generate a pdev against the values listed in Chapter 3 of the *Operator's Guide to File System Maintenance.*

For more information about pdevs, see the *Operator's Guide to File System Maintenance.*

PDEV { **-DECODE** *pdev* | **-DISK** *diskname* | **-ENCODE** *options* | **-HELP** }

-DECODE *pdev*

Decodes *pdev* and displays its four components:

- Disk controller address (in octal) followed by the controller number in parentheses
- Disk drive unit number
- Starting head number
- Number of heads in the partition

-DISK *diskname*

Displays the ldev (logical device number), the pdev, and the pdev's decoded components for disk *diskname.* When *diskname* includes a wildcard, PDEV displays this information for all disks matching the wildcarded name. Do not use enclosing angle brackets (< >) with *diskname.*

You can use PDEV with the -DISK option as a command function in another command. For instance, FIX_DISK [PDEV -DISK SYSTEM] returns only the pdev of the local disk SYSTEM in the FIX_DISK command. When you use PDEV as a command function, you cannot use wildcards in *diskname.*

-ENCODE -CONTROLLER *aa* **-UNIT** *u* **-STARTING_HEAD** *n* **-NUM_HEADS** *m*

Generates a pdev from the four values passed as options. PDEV displays the input information and the resulting pdev.

–CONTROLLER

Controller address (*aa*), which is one of these two-digit octal numbers: 22, 23, 24, 25, 26, 27, 45, and 46. Each controller address is followed by a corresponding controller number (0 through 7) in parentheses. Because controller number assignments vary in different contexts, it is recommended that you always refer to a controller by its two-digit controller address. Abbreviated –CTRL.

–UNIT

Disk drive unit number (*u*), which is an octal number from 0 through 7.

–STARTING_HEAD

Starting head (or surface) number (*n*), which is an even number from 0 through 30 (decimal). Abbreviated –START.

–NUM_HEADS

Number of heads (*m*) in the partition, which is an even number unless the partition is the last partition on a disk with an odd number of heads, or surfaces. These numbers range from 1 through 31 (decimal). Abbreviated –HEADS.

–HELP

Displays a summary of the command's functions.

Calculating PDEVs: You should check all pdevs that you calculate using the PDEV command against the pdevs listed in Chapter 3 of the *Operator's Guide to File System Maintenance*. The PDEV command allows you to calculate inappropriate pdevs. For example, you should not have an odd number of surfaces in a partition in the middle of a disk. In addition, the PDEV command allows you to calculate pdevs that are beyond the range of the table.

To obtain the values you need to construct a pdev for SMDs and FMDs, see the *Operator's Guide to File System Maintenance*.

Examples: Display the device values of a disk with a pdev of 4060:

```
OK, PDEV -DECODE 4060.
For PDEV 4060, controller = '26 (1), unit = 0, start head
= 0, heads = 16
```

Display the pdev and its device values of a disk named OSDSK3:

```
OK, PDEV -DISK OSDSK3
Disk      LDEV   PDEV    Controller   Unit   Start   Heads
-------   ----   ------  ----------   ----   -----   -----
OSDSK3      2    7660     '27 (5)       0       0      30
```

Use a wildcard to display a list of pdevs and their device values:

```
OK, PDEV -DISK P@
Disk      LDEV   PDEV    Controller   Unit   Start   Heads
-------   ----   ------  ----------   ----   -----   -----
PERFOR     4     6260    '27 (5)       0       0      24
PERF       5    41666    '27 (5)       3       8       6
PMANGR    12    62761    '23 (7)       0      12      11
OK,
```

Use PDEV to get the pdev of a disk device with the specified values:

```
OK, PDEV -ENCODE -CONTROLLER 27 -UNIT 2 -START 12 -HEADS
10
For controller = '27 (5), unit = 2, start head = 12,
heads = 10, PDEV = 62664
```

Use a PRIMOS argument list with PDEV to get the pdevs of disk unit 0 on each of eight different controllers:

```
OK, PDEV -ENCODE -CTRL (22 23 24 25 26 27 45 46) -UNIT 0 -START 2 -HEADS 28
For controller = '24 (0), unit = 0, start head = 2, heads = 28, PDEV = 17020
For controller = '26 (1), unit = 0, start head = 2, heads = 28, PDEV = 17060
For controller = '25 (2), unit = 0, start head = 2, heads = 28, PDEV = 17120
For controller = '22 (3), unit = 0, start head = 2, heads = 28, PDEV = 17160
For controller = '45 (4), unit = 0, start head = 2, heads = 28, PDEV = 17220
For controller = '27 (5), unit = 0, start head = 2, heads = 28, PDEV = 17260
For controller = '46 (6), unit = 0, start head = 2, heads = 28, PDEV = 17320
For controller = '23 (7), unit = 0, start head = 2, heads = 28, PDEV = 17360
```

You can also invoke PDEV as a command function from the command line or a CPL program. Enclose PDEV and its options in square brackets ([ ]); usage is otherwise identical.

For further details on this command, refer to the *Operator's Guide to File System Maintenance.*

SPIN_UP Command

Rev. 23.3 provides the new SPIN_UP command. This command is used to spin up a disk drive that is connected to a Model 7210 controller and mounted in either a Model 75500-6PK device module or a P53xx topcap assembly. SPIN_UP reverses the action of the SPIN_DOWN command. It is principally used to put back on line a disk that was accidentally spun down using the SPIN_DOWN command. SPIN_UP can only be issued from the supervisor terminal.

You issue the SPIN_UP command as follows:

```
SPIN_UP pdev
```

pdev is the physical device number (in octal) of the disk drive. You can only spin up a disk that is not in use; you cannot spin up a physical disk containing the COMDEV (unless COMDEV is mirrored), a paging, added, or assigned partition, or a partition activated for crash dump to disk.

Issuing a SPIN_UP command for a disk that is already spinning performs no operation and returns the `OK,` prompt.

Issuing a SPIN_UP command for a non-existent disk on a controller that supports SPIN_UP logs a disk error message to DSM and returns the following error message to the supervisor terminal:

```
Error back from spinup$. (spin_up)
```

Issuing a SPIN_UP command for a disk interfaced via a non-existent controller, a controller type that does not support SPIN_UP, or a controller that has not been downline loaded with code to support SPIN_UP returns the following error message:

```
SPIN_UP not supported by this disk controller in this
mode. (spin_up)
```

For further details on using the SPIN_UP and SPIN_DOWN commands, refer to the *Disk Replacement Procedure for the Model 75500-6PK Device Module.*

STATUS DEVICES Command Option

DEVICES

The DEVICES option displays information about assigned magnetic tape devices and disks.

At Rev. 23.3, the DEVICES option displays disks that are assigned to users, and, if you are using the system console or are the System Administrator or a member of the .RAS$ ACL group, it will also list the disks in the Assignable Disks Table that are not yet assigned.

For further details on this command, refer to the *Operator's Guide to System Commands.*

SHUTDN Command

SHUTDN –VERIFY: At Rev. 23.3, the SHUTDN command includes –VERIFY and –VERIFY –DETAIL. The –VERIFY option allows you to verify that there are no users on the disks to be shut down. If there are users on a specified disk, SHUTDN informs you how many users are on the disk and asks if you really want to shut the disk down. If there are no users on a specified disk, SHUTDN shuts down the disk. The –DETAIL option, which can be specified *only* with the –VERIFY option, lists the users that are currently using the disks to be shut down.

SHUTDN { *pdev1* [*pdev2* . . . *pdev9*] [**–FORCE**] | *pdev* **–RENAME** *diskname* } [**–VERIFY** [**–DETAIL**]]

–VERIFY [–DETAIL]

Checks to see if there are users attached to or have open files on the affected partitions. If there are any, SHUTDN asks `REALLY?` (as it does with the SHUTDN ALL command). You cannot use the –VERIFY option with the –ON option. If you attempt to use the –VERIFY option with the –ON option, the following error message is displayed:

```
The -VERIFY option may not be used with the -ON option.
(shutdn)
```

The –DETAIL suboption displays a list of users who have files or attach points on the affected partitions. You can only use –DETAIL with the –VERIFY option. If you attempt to use –DETAIL without the –VERIFY option, the following error message is displayed:

```
The -DETAIL option may only be used with the -VERIFY
option. (shutdn)
```

Note When you use –VERIFY and –VERIFY –DETAIL, no lock is applied. Therefore, the number of logged-in users may change between the time you issue this command and the time of the actual shutdown. Information you receive from these options is accurate only at the time that you issued these options.

Error Messages: When you shut down a disk, the following additional messages can be displayed.

```
*** Disk DTA/DTM Errors detected,
    Run FIX_DISK on DISK nnnnn  ***
```

```
*** Disk Return Record Errors detected,
    Run FIX_DISK on DISK nnnnn  ***

*** Disk CRA Mismatch Errors detected,
    Run FIX_DISK on DISK nnnnn  ***
```

These messages are displayed if the corresponding error has been detected since the disk was added. It is the System Administrator's responsibility to decide when it is best to run FIX_DISK.

For further details on this command, refer to the *Operator's Guide to System Commands.*

ICS3 Communications Controller Enhancements

Rev. 23.3 provides a new down-line load (DLL) file for the ICS3 communications controller. This DLL provides the following enhancements:

- Enhanced performance in passing X.25 frames to the PRIMENET process.
- Support for two 56KB lines on a single ICS3 controller.

For further details, refer to the *ICS User's Guide.*

Documentation Corrections

DSM User's Guide

Page 6–4

The ADMIN_LOG subcommand's –RETAIN option specifies how many days messages are to be retained on a log before being deleted. If you do not specify this option, the default retention period is 8 days. If you specify the –RETAIN option but do not specify the *days* value, the retention period is indefinite.

Prime INFORMATION RECOVERY Reference Guide

Page 2–2

change:	RECORD LOCKU
to:	RECORDLOCKU

Page 2–4, note at top of page

change: return on error.
to: return an error.

Page 5–2

change: 2. Locing/Reading
to: 2. Locking/Reading

CONFIG_USERS Terminal Interface Definition

4

This chapter describes how you can define the interface between the CONFIG_USERS utility and your terminal. The System Administrator uses the CONFIG_USERS screen interface to define the attributes of user accounts. The features described in this chapter permit the System Administrator to interact with these CONFIG_USERS screens using a non-Prime terminal.

Note This chapter does not describe any new functionality. The features described here have always been available to users of CONFIG_USERS. This chapter makes documentation of these features more readily available to users.

CONFIG_USERS terminal interface definition is a restrictive subset of the PRIFORMA forms design facility, which is available from Prime as a separately purchased software product. The features described in this chapter can only be used to interface with CONFIG_USERS; PRIFORMA includes many other features not included here. Users familiar with PRIFORMA forms design will find that they are already familiar with the utilities described in this chapter; these users do not have to change any of their existing procedures. For further information, refer to the *PRIFORMA Forms Design and Administration Guide.*

Terminal Administration

This section tells you how terminals are administered within PRIFORMA to achieve terminal independence, the default location of the Keys and Video files, and how PRIFORMA applications pick up these files at runtime.

The SMVARS File

The SMVARS file is the key to terminal independence. The SMVARS file defines the location of terminal description and other important files required by a PRIFORMA application at runtime.

In particular, the SMVARS file defines the location of

- Video files for all terminal types on the system
- Keys files for all terminal types on the system
- The PRIFORMA Messages file, which contains text for PRIFORMA runtime messages
- The Batchkeys file, which enables batch files created on one terminal to be interpreted correctly on another
- The language and character set to be used, for the localization of PRIFORMA applications

Location of the SMVARS File

The .SMVARS global variable defines the location of the SMVARS file. The .SMVARS global variable is set up initially to define the location of the SMVARS file in the PRIFORMA* directory.

Location of the Keys and Video Files

The SMVARS file uses the keyword SMTTPPATH to specify the default location of the terminal Video and Keys files. By default, the terminal description files are located in

```
PRIFORMA*>TERMINAL_FILES
```

For example, the supplied SMVARS file specifies the location of the terminal description files as follows:

```
SMTTPPATH = PRIFORMA*>TERMINAL_FILES
```

If you wish to move all the terminal description files to another directory, you should change the pathname specified by SMTTPPATH.

The Keys and Video files for a given terminal can be in any directory. If you wish to specify a location for the Keys and Video files other than that specified by SMTTPPATH, you use the keywords SMKEY and SMVIDEO in the SMVARS file, as follows:

```
SMKEY = (term) keys_pathname
```

```
SMVIDEO = (term) video_pathname
```

where *term* is the mnemonic of the terminal type for which the Keys and Video files have been written; *keys_pathname* and *video_pathname* are the pathnames of the Keys and Video files respectively.

For example, under PRIMOS, the following entries in the SMVARS file locate the Keys and Video files for a PT200 terminal, in a directory called TTP_TEST:

```
SMKEY = (PT200)PRIFORMA*>TTP_TEST>PT200KEYS.BIN


SMVIDEO = (PT200)PRIFORMA*>TTP_TEST>PT200VIDEO.BIN
```

Runtime Access to the Keys and Video Files

When a PRIFORMA application initializes a terminal, it calls the PRIFORMA runtime library routine ICRT (or uses the FML statement SET TERMINAL), passing, as a parameter, the mnemonic of the terminal it wishes to initialize. For example, to initialize a PT200 terminal, the PRIFORMA application calls ICRT passing the parameter "pt200".

At this point the SMVARS file is searched for the SMVIDEO entry corresponding to pt200. This entry defines the location of the video file *term*VIDEO.BIN. By default, PRIFORMA also locates the *term*KEYS.BIN file corresponding to the terminal mnemonic passed to ICRT, from the equivalent SMKEY entry. However, ICRT allows the application to specify a different Keys file if required.

If no SMVIDEO or SMKEY entry is found in the SMVARS file, the location specified by SMTTPPATH is taken by default.

Terminal Administration Summary

If you want to add a new terminal type to PRIFORMA, follow these steps:

1. Create a Video file for your terminal.
2. Compile the Video file using the COMPILE_FORMTERM utility, to generate a binary Video file of the form *term*VIDEO.BIN.
3. Copy the *term*VIDEO.BIN file into the required directory. (See earlier in this chapter for the default locations of the Video file.)
4. Create a Keys file for your terminal, using MODIFY_FORMKEY.
5. Compile the Keys file using the COMPILE_FORMKEY utility, to generate a binary Keys file, of the form *term*KEYS.BIN.
6. Copy the *term*KEYS.BIN file into the required directory. (See earlier in this chapter for the default locations of the Keys file.)

MODIFY_FORMKEY and the Keys File

This section tells you how to use the MODIFY_FORMKEY utility. You may also wish to refer to the previous section on Terminal Administration for background information on terminal independence and the Keys file.

This section also tells you how to compile the Keys file to produce a binary version for use by PRIFORMA.

Keys File Sequences

Different terminals generate different character sequences when a given key is pressed. PRIFORMA applications use a standard set of logical keys, which map onto physical keys on a given terminal. PRIFORMA uses the Keys file for a given terminal to translate physical key sequences into logical keys. The MODIFY_FORMKEY utility allows you to create or modify a Keys file for a terminal.

Note You should be careful, when selecting key sequences, that each sequence is unique. A long sequence cannot include a shorter sequence that has already been assigned to a different function. For example, if the sequence

```
ESC A B C
```

is defined as the EXIT key, and the sequence

```
ESC A
```

is defined as the REFR key, the EXIT sequence would not be recognized by PRIFORMA. The sequence ESC A B C would be interpreted as REFR 'B' 'C'.

Invoking MODIFY_FORMKEY

To invoke the MODIFY_FORMKEY utility, type

```
MODIFY_FORMKEY -TTP terminal_type
```

where `-TTP terminal_type` identifies the terminal type on which you invoke the command. If you do not specify the –TTP parameter, MODIFY_FORMKEY prompts you for the terminal type. MODIFY_FORMKEY is abbreviated MFKY.

For example, if you are using MODIFY_FORMKEY on a PT200 terminal, type

```
MODIFY_FORMKEY -TTP PT200
```

When you invoke the MODIFY_FORMKEY utility, the Entry Screen is displayed:

```
                    MODIFY_FORMKEY UTILITY

Using this utility you can edit a previously created KEYS file or
create a new one.

Enter the name of the file you want to create or modify in the field
below and then press the "+" key. The filename is of the format
"nameKEYS" and "KEYS" is assumed if not present. If you leave the
field blank and press "+", a default file name will be used. Default
names are of the format "nameKEYS" where name is the TERMINAL TYPE
of the terminal you are using.

To exit the MODIFY_FORMKEY utility without proceeding
further, press the "-" key.

File Name: __________ (Enter '<' to BACKSPACE
                      Enter "+" to TRANSMIT
                      Enter "-" to EXIT)

Note: Control keys are not active in this utility. Instead,
data keys are used for control purposes.
```

Control Keys and Data Keys

You use the MODIFY_FORMKEY utility to define cursor control, editing, and function keys. Because MODIFY_FORMKEY determines the escape sequences of the keys pressed, you are not able to use these keys in the MODIFY_FORMKEY utility itself. Instead, you must use standard ASCII data keys for cursor control and editing.

For example, to move from one field to the next, you would usually use the Tab key. However, because Tab is one of the keys that you can define in this utility, you cannot use it as a cursor control key. You must use the ASCII key T in its place. Similarly, you must use the B key in place of Backtab.

You can use data keys for control purposes, because in this utility you do not use these keys to type data into fields. This is explained in subsequent sections.

The control functions that the MODIFY_FORMKEY utility supports, and the keys you use to provide them, are listed below.

Key	*Control Function*
+	XMIT
–	EXIT
?	HELP
t	TAB
b	BACKTAB
<	BACK SPACE
d	FIELD ERASE
z	ERASE ALL UNPROTECTED

The Entry Screen

On the MODIFY_FORMKEY Entry Screen, you specify the Keys File that you wish to create or modify. Enter the name of the file into the field labeled `File Name:`, then press the + key to complete the screen. Note that you can enter a pathname in response to this prompt. Pathnames are restricted to a maximum of 128 characters.

If you make a mistake, press the < key to move the cursor back one position at a time. The < key erases previously entered characters as it moves.

If you enter a filename which is not of the format *name*KEYS, KEYS is appended to the *name* you supply. If you do not enter a filename, the default *term*KEYS is used, where *term* is the terminal type mnemonic supplied via the –TTP command line parameter.

If the file you specify exists, it is read and can then be modified. If it does not exist you create a new Keys file.

Note MODIFY_FORMKEY makes all modifications in memory. The utility updates files only at the conclusion of the program and at your request. See the section Exiting the Utility, later in this chapter, for details.

To exit the MODIFY_FORMKEY utility while the Entry Screen is displayed, press the hyphen key (–).

Main Menu

The Main Menu is displayed at the conclusion of the Entry Screen and whenever you return from another function.

```
          MODIFY_FORMKEY UTILITY MAIN MENU



     0.  Exit
     1.  Help
     2.  Define Cursor Control and Editing Keys
     3.  Define Function Keys
     4.  Define Shifted Function Keys
     5.  Define Application Function Keys
     6.  Define Miscellaneous Keys
     7.  Test Keys File

     Enter the desired option (0 - 7): _
```

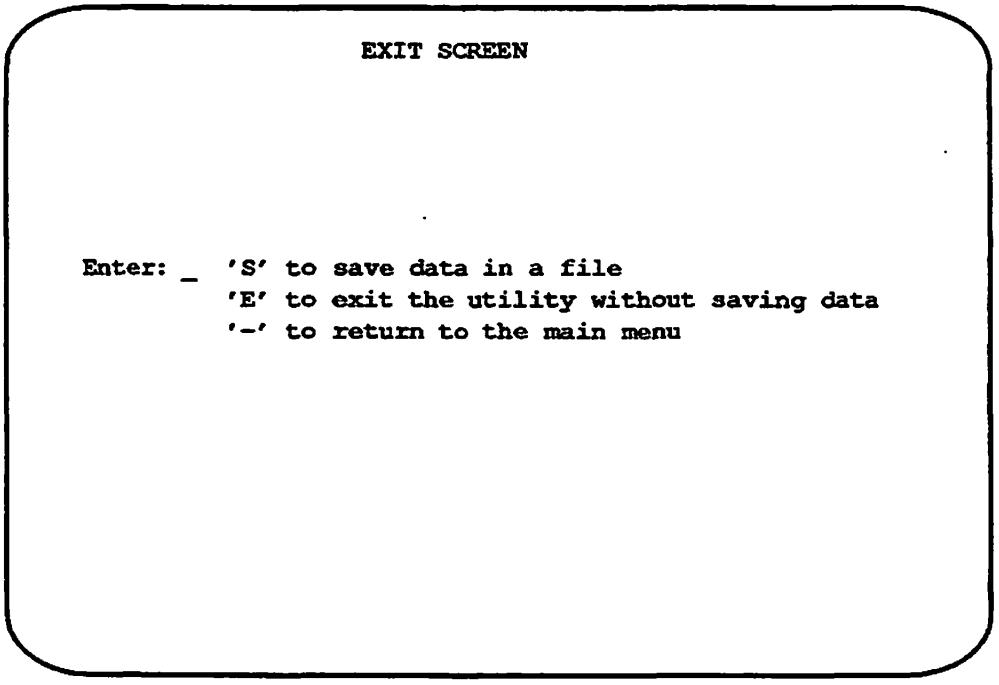
To select an option, type the number corresponding to the required option. For example, to get help, press 1.

If you make an invalid selection, an error message is displayed. You must press the hyphen key (–) to acknowledge the message. You cannot continue until you do so.

The functions that you can select from the Main Menu are described in the following subsections.

Exiting the Utility

To exit the MODIFY_FORMKEY utility, press 0 while the Main Menu is displayed. This invokes the Exit Screen.



The Exit Screen has a single field into which you must enter one of the following characters: S, s, E, e, or –. Any other character produces an error.

If you select –, the Main Menu is redisplayed, and the MODIFY_FORMKEY session continues, enabling you to make additional changes to the Keys file.

To exit the utility without saving the file, enter E.

To save the Keys file, enter S. Additional information is displayed on the Exit Screen as follows:

```
                    EXIT SCREEN


Enter: _   'S' to save data in a file
           'E' to exit the utility without saving data
           '-' to return to the main menu

File Name: _____________

Special Keys: + TRANSMIT (save changes in file)
              - EXIT (return to main menu)
              < BACKSPACE
```

The name of the Keys file that you are modifying is displayed in the `File Name:` field. (Note that you may enter a pathname in response to this prompt.)

To save the information in this file, type +.

To save the information in a different file, type < to place the cursor at the start of the response field and enter the new file name. Press the + key to save the data in the new file.

Note If the specified file exists, it is overwritten with the Keys file. If it does not exist it is created.

To return to the Main Menu from the `File Name:` field without writing the changes to disk, type –.

Help Screen

Press 1 to select the help screen from the Main Menu:

```
There are two types of keys on your keyboard - Data keys and Control keys. Data keys will
generate a single printable character when pressed. Control keys will generate a sequence
of one or more characters, the first of which is non-printable.

In subsequent screens, you will be asked to designate the control keys that should be used
for various functions. For example one control key will be designated as EXIT, another as
PF1. To assign a key to a function, the key must be pressed twice in succession. Try this
in the field below.

                                    Press key twice
CHARACTERS GENERATED:  _ _ _ _ _ _ _ _ _ _ _ _ _ _
           KEY STROKE: _ _ _ _ _ _

                      Use the "+" or "-" keys to exit to the main menu

When done correctly, the characters generated by the key will be shown in the KEY STROKE
field. As each key is typed, its characters are shown in the CHARACTERS GENERATED field.

If you get out of sync press the space bar repeatedly until a message appears
```

This screen provides help information and lets you test the kinds of keystroke entries that are required on subsequent screens in this utility. The Help Screen works in the following way:

- There are two types of key: those that generate a single printable ASCII character and those that generate a sequence of characters. When a sequence is generated, the first character is always an ASCII control character (value less than Hexadecimal 20 – see Appendix A).
- To see the characters that are generated for a particular key, press that key twice while the Help Screen is displayed. You must press the key twice because this facility can also accept a sequence of keystrokes. When you press the key twice, MODIFY_FORMKEY can detect a pattern and recognize the key, or key sequence.
- When you press the key the first time, the characters produced are shown in the CHARACTERS GENERATED row. When you press the key a second time and the utility recognizes the key, the sequence representing the key is shown in the KEY STROKE row.

You may sometimes wish to specify more than one keystroke to represent a key. For example, on a system with a small number of function keys, you may choose to implement the function keys F1 through F9 with the following keystrokes:

```
<CTRL F> n
```

where *n* is a number between 1 and 9. PRIFORMA can interpret these keystrokes as a single key. To demonstrate this, type

```
<CTRL F> 1
```

into the help screen, that is, press the F key while you hold down the control key and then press the 1 key. The sequence *ACK 1* appears in the CHARACTERS GENERATED row. If you press the sequence a second time *ACK 1* is duplicated in the CHARACTERS GENERATED row and also in the KEY STROKE row. When the KEY STROKE row is displayed, this indicates that the system has recognized the key sequence as complete.

Sequences cannot begin with a printable character; therefore if you enter a printable ASCII character as the first key in a sequence, the system immediately displays it in the KEY STROKE row.

If you enter two nonprintable characters in succession, the system assumes that you are trying to enter a sequence and continues to display these characters in the CHARACTERS GENERATED field. However, if the sequence exceeds six characters without repeats, the system displays the following error message:

```
Key sequence too long
```

You must press the hyphen key (–) to acknowledge the message.

If you enter a key or key sequence incorrectly and do not wish to duplicate that sequence, simply press any key repeatedly until the "Sequence too long" message is displayed. After you acknowledge the message you can start again.

To exit from the Help Screen and return to the Main Menu, press the hyphen key (–) as the first character in a sequence.

Defining Cursor Control and Editing Keys

Option 2 on the Main Menu allows you to specify the keys that are to be used for the various cursor control and editing operations. When you select option 2, the Cursor Control and Editing Key Definition Screen is displayed.

```
        CURSOR CONTROL AND EDITING KEY DEFINITION SCREEN

EXIT            __ __ __ __ __ RIGHT ARROW  __ __ __ __ __
TRANSMIT        __ __ __ __ __ LEFT ARROW   __ __ __ __ __
HELP            __ __ __ __ __ UP ARROW     __ __ __ __ __
LOCAL PRINT     __ __ __ __ __ DOWN ARROW   __ __ __ __ __
NEW LINE        __ __ __ __ __ CHAR DELETE  __ __ __ __ __
TAB             __ __ __ __ __ INSERT MODE  __ __ __ __ __
BACK TAB        __ __ __ __ __ FIELD ERASE  __ __ __ __ __
HOME            __ __ __ __ __ ERASE ALL    __ __ __ __ __
BACK SPACE      __ __ __ __ __ SCROLL UP    __ __ __ __ __
LAST FIELD      __ __ __ __ __ SCROLL DOWN  __ __ __ __ __
                               REFRESH      __ __ __ __ __
Each key or sequence of keys must be pressed twice in
succession.
Special Keys: + TRANSMIT      ? HELP
              - EXIT          d DELETE ENTRY
              t TAB           z ERASE ALL
              b BACKTAB
```

This screen has a field for each of the cursor control and editing functions that are supported by the system. This screen allows you to specify a sequence of characters for each key function.

Assigning a Key to a Function

To assign a key to a particular cursor control or editing function, enter that key twice, next to the function it is to represent. For example, to designate a key as the EXIT key, press that key twice in succession while the cursor is in the EXIT field.

When you have assigned a key correctly, the sequence of characters generated on behalf of the key is displayed and the cursor moves to the next field.

You cannot define a printable ASCII character as a cursor control or editing key. This means that the sequence of characters generated by the key must start with a value less than hexadecimal 20 (decimal 32). If you do not start the sequence with a value less than hexadecimal 20, an error is displayed. The utility also displays an error if the sequence of characters matches a sequence assigned to another function. Refer to Appendix A for a list of ASCII control characters.

If you leave a field blank, the function corresponding to that field does not operate in programs that use the Keys file you are defining. Therefore, if your program has no use for a particular key (such as the LAST FIELD key), you may leave that entry blank on this screen.

If by mistake you do not press the same key twice in succession, the system does not display the characters that were generated. To recover, simply press the hyphen key (–) repeatedly until the following message is displayed:

```
Key sequence too long
```

Press the hyphen key (–) to acknowledge the message and then enter the correct keystrokes.

Assigning a Sequence of Keys to a Function: You assign a sequence of keystrokes to a function in the same way as you assign an individual key; you first enter the entire sequence, then repeat it. As you repeat the sequence the field scrolls, allowing you to see the data that you enter. When you have successfully completed the sequence, the characters generated on behalf of that sequence are displayed.

Returning to the Main Menu: To save the changes that you have made in this screen and return to the Main Menu, press the + key as the first character in a sequence. To discard the changes and return to the Main Menu, press the – key.

Defining Function Keys

Option 3 on the Main Menu allows you to specify the keys that are to be used as the function keys (PF1 – PF24). When you select option 3 from the Main Menu, the Program Function Key Definition Screen is displayed.

```
          PROGRAM FUNCTION KEY DEFINITION SCREEN

PF1  ___ ___ ___ ___ ___          PF13 ___ ___ ___ ___ ___
PF2  ___ ___ ___ ___ ___          PF14 ___ ___ ___ ___ ___
PF3  ___ ___ ___ ___ ___          PF15 ___ ___ ___ ___ ___
PF4  ___ ___ ___ ___ ___          PF16 ___ ___ ___ ___ ___
PF5  ___ ___ ___ ___ ___          PF17 ___ ___ ___ ___ ___
PF6  ___ ___ ___ ___ ___          PF18 ___ ___ ___ ___ ___
PF7  ___ ___ ___ ___ ___          PF19 ___ ___ ___ ___ ___
PF8  ___ ___ ___ ___ ___          PF20 ___ ___ ___ ___ ___
PF9  ___ ___ ___ ___ ___          PF21 ___ ___ ___ ___ ___
PF10 ___ ___ ___ ___ ___          PF22 ___ ___ ___ ___ ___
PF11 ___ ___ ___ ___ ___          PF23 ___ ___ ___ ___ ___
PF12 ___ ___ ___ ___ ___          PF24 ___ ___ ___ ___ ___
Each key or sequence of keys must be pressed twice in
succession
Special Keys:       + TRANSMIT          ? HELP
                    - EXIT              d DELETE ENTRY
                    t TAB               z ERASE ALL
                    b BACKTAB
```

To designate a key (or key sequence) as a function key, press it twice in succession while the cursor is positioned in the field to which the sequence applies. For example, to define <CTRL F> as the PF2 key, position the cursor to the PF2 field and type <CTRL F> (press the F key at the same time as the CTRL key) twice in succession.

To save the changes you make and return to the Main Menu, press the + key as the first character in a sequence. To discard the changes and return to the Main Menu, use the – key.

Options 4 and 5 define which keys are used as the shifted function keys (SPF1–SPF24) and the application function keys (APP1–APP24) respectively. These options operate in a similar way to option 3, described above.

Defining Miscellaneous Keys

Option 6 on the Main Menu allows you to specify additional key functions that you cannot define on other screens. It also allows you to define alternative control sequences for previously defined keys; for instance, if you require more than one key (or key sequence) for the same function.

When you select option 6 from the Main Menu, the Miscellaneous Key Definition Screen is displayed.

```
          MISCELLANEOUS KEY DEFINITION SCREEN

KEY STROKE LOGICAL VALUE      KEY STROKE LOGICAL VALUE
- - - - - - ____________      - - - - - - ____________
- - - - - - ____________      - - - - - - ____________
- - - - - - ____________      - - - - - - ____________
- - - - - - ____________      - - - - - - ____________
- - - - - - ____________      - - - - - - ____________
- - - - - - ____________      - - - - - - ____________
- - - - - - ____________      - - - - - - ____________
- - - - - - ____________      - - - - - - ____________

LOGICAL VALUE DISPLAY MODE IS:

Special Keys: +  TRANSMIT      ?  HELP
              -  EXIT          d  DELETE ENTRY
              t  TAB           z  ERASE ALL
              b  BACKTAB       v  TAB TO VALUE FIELD
              c  CHANGE MODE   <  BACKSPACE IN VALUE FIELD
```

On this screen you must define the logical values as well as the sequences that produce them. On all other MODIFY_FORMKEY screens the logical value is implicitly determined by the field with which the key sequence is associated.

The Miscellaneous Key Definition screen has two columns for each key being defined. These columns are headed KEY STROKE and LOGICAL VALUE respectively. Enter a key or a key sequence into the KEY STROKE field twice in succession. The system displays the characters that are generated and the cursor moves to the LOGICAL VALUE column for that key.

Entering the Logical Values: When the cursor is in a LOGICAL VALUE field, enter the logical value that should be returned when the corresponding character sequence is read from the keyboard. Because logical values are numbers, you must enter printable ASCII data into this field. This is in contrast to most other fields where you cannot enter data characters.

When you enter logical values, you can use three keys in addition to the data keys necessary to enter the values. These three keys are +, –, and <.

- Use the + key to signify that the logical value you typed is correct and should be used. When you press this key, the cursor moves to the next field.
- Use the – key to signify that the logical value you typed is incorrect and should be ignored. When you type this key, the cursor moves to the next field, and the logical value is reset to the value that existed before the field was entered. If the logical value field was previously empty, it is set to zero.

- Use the < key to move the cursor backwards one position at a time so that you can make corrections to the logical value. The < key erases previously entered data as the cursor moves backwards.

Logical Value Modes: You can enter and display logical values in any one of four modes:

- Hexadecimal
- Mnemonic
- Decimal
- Octal

The current mode is displayed on the 17th line of the screen as follows:

```
LOGICAL VALUE DISPLAY MODE IS: current_mode
```

To change the current mode, type C as the first character of a sequence while the cursor is in any of the KEY STROKE fields on the screen. When you first invoke the MODIFY_FORMKEY utility, *current_mode* is set initially to Hexadecimal. It then cycles to Mnemonic, Decimal, Octal, and back to Hexadecimal each time you press the C key.

Hexadecimal Mode

When you set the screen to Hexadecimal mode (the initial default), you may enter up to eight hexadecimal digits (0–f) in the Logical Value field if the machine uses 4–byte integers, or up to four digits if it uses 2–byte integers. If you enter a nonhexadecimal digit, an error message is displayed: you must press the hyphen key (–) to acknowledge this message.

The largest hexadecimal value that you can store is ffffffff in a 4–byte integer, or ffff in a 2–byte integer.

Mnemonic Mode

When you set the screen to Mnemonic mode, you may enter the mnemonic associated with any of the standard set of logical key values.

Mnemonic tokens corresponding to the logical values of PRIFORMA keys are held in

```
SYSCOM>FN_KEYS.lang
```

For example, if you enter EXIT into the Logical Value field, the logical value of the EXIT key (0x03) is used.

If you enter an incorrect mnemonic, an error message is displayed. After you press the hyphen key (–) to acknowledge the message, a list of the valid mnemonics is displayed on the screen. To clear the list of mnemonics and resume entry, press the hyphen key (–).

Decimal Mode

When you set the screen to Decimal mode, you may enter a maximum of ten digits in the Logical Value field if the machine uses 4–byte integers, or a maximum of five digits if the machine uses 2–byte integers. If you enter a character that is not a digit, an error message is displayed: you must press the hyphen key (–) to acknowledge this message.

Octal Mode

When you set the screen to Octal mode, you may enter a maximum of 11 octal digits (0–7) in the Logical Value field if the machine uses 4–byte integers, or a maximum of six digits if it uses 2–byte integers. If you enter a non-octal digit, an error message is displayed: you must press the hyphen key (–) to acknowledge this message.

The largest octal number that you can store is 37777777777 in a 4–byte integer, or 177777 in a 2–byte integer. If you enter a larger value, a warning message is displayed and the last digit is truncated.

Returning to the Main Menu: To save the changes that you make in this screen and return to the Main Menu, press the + key while the cursor is in a KEY STROKE field. To discard the changes and return to the Main Menu, press the – key.

Test Keys File

Option 7 on the Main Menu allows you to test the Keys file you are defining. You can use this option to enter keystrokes and see the translated logical value displayed on the screen.

When you select option 7 from the Main Menu, the Test Keys File Screen is displayed.

This screen has two fields headed KEY STROKE and LOGICAL VALUE. If you enter a keystroke, or sequence of keystrokes, that you have previously defined, the system displays the logical value of that key. You need only press the keystroke once because you are testing the table to see how it will behave when used in a real application.

```
                    TEST KEYS FILE
This screen is used to test the Keys file being defined.To do
this, press any key in the field below. The characters generated
by the key will be displayed along with its logical value.

          KEY STROKE              LOGICAL VALUE

           ___ ___ __ ___ ___     _______________

          LOGICAL VALUE DISPLAY MODE IS: _______

If a multiple key sequence has been defined, the entire sequence
must be entered for the logical value to be displayed. Once the
sequence is started, the cursor will be turned off until it is
completed.
If you get out of sync press the space bar repeatedly until a
message appears.

          Special Keys:  +  TRANSMIT
                         -  EXIT
                         c  CHANGE MODE
```

Certain keys are reserved and are not available to application programs. Examples of such keys are

- PRINT SCREEN
- <CTRL C> (Program break on the PC),
- <CTRL P> (Program break on the 50 Series)
- <CTRL S> (Scroll pause).

If you press a key and it does not completely satisfy one of the previously specified sequences, the cursor is turned off and the system waits for another key. This continues until a sequence is matched or the system determines that no match is possible. When the sequence is matched, the logical value is displayed and you can enter a new key. When no match is possible, the following message is displayed:

```
Key not defined
```

Logical values can be displayed in one of four modes

- Hexadecimal
- Mnemonic
- Decimal
- Octal

These modes are presented in the order shown above. To change to the next mode, type the character C as the first character in a sequence.

To exit the screen and return to the Main Menu, press the + or – key.

To obtain help text, press the ? key.

Format of the Keys File

The Keys file output from MODIFY_FORMKEY is an ASCII file and can be edited using any text editor, as long as you retain the format of the file. This section describes the required format of the Keys file.

Each entry in the Keys file is contained on a single line.

Each entry has the following format:

```
logical_value description = char_sequence
```

logical_value

Can either be a mnemonic as defined in the file FN_KEYS.*lang*, or a hexadecimal value. There are mnemonic values in the FN_KEYS.*lang* file for the cursor control and editing keys (Tab, Home, etc.), the program function keys (PF1 to PF24), the shifted function keys (SPF1 to SPF24), and the application function keys (APP1 to APP24). If you use hexadecimal values, they must be in the form

```
0xhhhh
```

where *hhhh* is a string of hexadecimal digits (0–F). MODIFY_FORMKEY differentiates between mnemonic and hexadecimal entries in the following way:

- Hexadecimal values you specify all appear on the Miscellaneous Key Definition Screen. The MODIFY_FORMKEY utility can process a maximum of 24 entries specified in the hexadecimal format.
- Mnemonic values you specify are shown on the screen to which the mnemonic applies. For example, an entry specified with EXIT as its logical value is shown on the Cursor Control and Editing Key Definition Screen.

description

A text description of the key of up to 15 characters. Leading and trailing spaces are removed from the description. Descriptions longer than 15 characters are truncated. At least one space must separate the *logical_value* and the *description*. The *description* cannot contain the equals sign (=) character. See the subsection Use of Key Descriptions later in this chapter.

char_sequence

A sequence of up to six characters that is translated to the *logical_value*. The ASCII control characters are represented by their standard names (ESC, SOH and so on – see Appendix A). Other characters are simply entered. Each character in the sequence is separated by one or more spaces. If you want to enter the backslash character (\) you must prefix it with another backslash (\\).

A typical Keys file for a PT200 terminal is as follows:

```
EXIT   Cancel          = ESC N U
XMIT   Enter           = ESC N c
HELP   Help            = ESC _ 1 ESC \\
LP     Prt Scn         = ESC [ 0 i
NL     Return/Enter    = LF
TAB    Tab             = HT
BACK   Back Tab        = ESC [ Z
HOME   Home            = ESC $ A
BKSP   Backspace       = BS
EMOH   SHIFT Home      = ESC $ B
RARR   Right Arrow     = ESC [ C
LARR   Left Arrow      = ESC [ D
UARR   Up Arrow        = ESC [ A
DARR   Down Arrow      = ESC [ B
DELE   Delete          = ESC [ P
INS    Insert          = ESC [ 4 h
FERA   Erase           = ESC [ X
CLR    SHIFT Erase     = ESC [ K
SPGU   Scroll Up       = ESC [ S
SPGD   Scroll Down     = ESC [ T
REFR   CONTROL Clear   = ESC ?
PF1    F1              = ESC O !
PF2    F2              = ESC O "
PF3    F3              = ESC O #
PF4    F4              = ESC O $
PF5    F5              = ESC O %
PF6    F6              = ESC O &
PF7    F7              = ESC O '
PF8    F8              = ESC O (
PF9    F9              = ESC O M
PF10   F10             = ESC O NS
PF1    SHIFT F1        = ESC O )S
PF2    SHIFT F2        = ESC O *S
PF3    SHIFT F3        = ESC O +S
PF4    SHIFT F4        = ESC O ,S
PF5    SHIFT F5        = ESC O -S
PF6    SHIFT F6        = ESC O .S
PF7    SHIFT F7        = ESC O /S
PF8    SHIFT F8        = ESC O 0S
PF9    SHIFT F9        = ESC O [S
PF10   SHIFT F10       = ESC O \\
```

If the same mnemonic appears more than once in the file, MODIFY_FORMKEY uses the last occurrence of that mnemonic. If duplicate right–hand sides appear with different logical values, the results are unpredictable.

Any lines in the incorrect format are discarded by the MODIFY_FORMKEY utility and by any PRIFORMA programs.

Use of Key Descriptions

The Keys file allows you to associate a text description with each key. The key description text can be accessed at runtime by the application making a GET_DESC routine call. Key description text can be used to tailor messages generated by your application to the terminal on which the application is run.

For example, on the PT200 terminal keyboard, the Cancel key is commonly used as the EXIT key. In the example entries given above, the text description for the EXIT key is `Cancel`. Your application may want to tell the end user to press the EXIT key. Using the key description you can improve on messages of the type `Press the EXIT key to abort`. The key description for the EXIT key can be picked from the Keys file and substituted in the message to tailor it as follows: `Press the Cancel key to abort`.

Compiling the Keys File

COMPILE_FORMKEY converts a Keys file into binary format. To use COMPILE_FORMKEY you must have a Keys file named according to the convention *term*KEYS, where *term* is a mnemonic for the terminal for which the Keys file has been developed.

To run COMPILE_FORMKEY on the *term*KEYS file you have created, follow the steps outlined below.

Type

```
COMPILE_FORMKEY termKEYS
```

(If you do not specify the word KEYS, it is appended to the file.)

COMPILE_FORMKEY produces a file of the form *term*KEYS.BIN in your current directory. COMPILE_FORMKEY is abbreviated `CFKY`.

Before PRIFORMA will recognize your new Keys file, you need to make an entry in the SMVARS file, as described in the Terminal Administration section, found earlier in this chapter.

The Video File

Video files are supplied for a number of terminal types. This section tells you how to create a new (or modify an existing) Video file, to enable a terminal type for which no Video file is supplied to work correctly with PRIFORMA.

The theory of terminal independence within PRIFORMA is discussed in detail earlier in this chapter in the Terminal Administration section.

Format of the Video File

The PRIFORMA Video file is an ASCII text file that can be edited using a text editor. Each line in the file is either an instruction of the form

```
keyword = variable_data
```

or a comment line, denoted by /* at the first position on the line.

Each *keyword* denotes a characteristic of the terminal that PRIFORMA needs to know. The *variable_data* describes the instruction sequence required to generate that characteristic on the terminal for which the file is being written. For example:

```
CON = ESC $ R
```

describes the escape sequence required to turn the cursor on. *variable_data* can be a number, a sequence of characters, or a further list of commands, depending on the *keyword* being defined.

The following general formatting rules apply to lines in the Video file, regardless of the *keyword* being defined

White space: All white space (spaces and tabs) is ignored, except when you use it with the backslash (\) or double quote (") characters, as described below.

Continuation lines: A logical line can continue onto the next physical line by ending the previous physical line with a backslash. For example:

```
this line is too \
long
```

If you want to end the line with a backslash as part of the text, enter a double backslash:

```
this line ends with a backslash \\
```

If you want a line that ends in a backslash to continue on the next line, enter a backslash, a space, and a second backslash:

```
this line ends in a backslash \ \
but logically continues here
```

Character Strings: Character strings are delimited by double quotes. White space within quotes is *not* ignored. If you want to use the double quote (") character as part of a character sequence, that is, not as a string delimiter, precede it with a backslash (\").

Unprintable Characters: Use the standard ASCII mnemonics to enter unprintable characters in your video file. ASCII mnemonics should be entered in uppercase. These values are listed in Appendix A.

Hexadecimal Character Codes: Any character can be entered by entering its hexadecimal ASCII character code in the form 0x followed by two hexadecimal digits. Thus A could be referred to as 0x41.

Video File Commands

Certain character sequences cannot be specified in advance within the Video file and need to use values provided at runtime; for example, the escape sequence to position the cursor at a particular position. This escape sequence requires the line and column coordinates of the required position to be supplied at runtime. The commands that use such sequences are given parameters. Parameters are accepted and manipulated using percent commands, so called because they are identified by a preceding percent (%) sign. This section tells you how to use percent commands.

Keywords Taking Parameters

The following keywords take parameters:

Keyword	*Sequence and Parameters Required*
REPT	Repeat sequence Character to be repeated and number of times to repeat
EW	Erase window Start line, start column, and background color
CUP	Cursor position Line and column (relative to 0)
CUU	Cursor up Line decrement
CUD	Cursor down Line increment

CUF	Cursor forward Column increment
CUB	Cursor backward Column decrement
SGR	Set graphics rendition
ASGR	Area set graphics rendition (See the Attribute Keywords subsection later in this chapter.)

Summary of Percent Commands

The percent commands are described in detail later in this chapter. The following list summarizes the percent commands and indicates whether the command is a TERMCAP command (C), a TERMINFO command (I), or a special PRIFORMA extension (E). (TERMCAP is an early version of TERMINFO.)

Output commands

%%	Output a percent sign (C and I)
%.	Output a character (C)
%c	Output a character (I)
%d	Output a decimal value (C and I)
%2d	Output a two–digit decimal value, blank fill (C and I)
%3d	Output a three–digit decimal, blank fill (C and I)
%02d	Output a two–digit decimal value, '0' fill (C and I)
%03d	Output a three–digit decimal, '0' fill (C and I)
%+	Add and output a character (C)
%#z	Output # (any decimal number) of binary zeroes (E)

Stack manipulation commands

%p#	Push parameter # (1 – 11) (I)
%'c'	Push the character constant (I)
%{#}	Push the integer constant (I)
%+	Arithmetic addition (I)
%–	Arithmetic subtraction (I)
%*	Arithmetic multiply (I)

%/	Arithmetic divide (I)
%m	Modulus operation (I)
%| %^ %&	Bitwise OR, XOR, AND (I)
%= %> %<	Logical condition tests (I)
%! %~	Logical NOT, ones complement (I)

Parameter sequencing commands

%#u	Use one or more parameters
%#b	Backup one or more parameters

Parameter changing commands

%i	Increment the next two parameters (C and I)
%r	Reverse the next two parameters (C and I)

Control commands

%? expr %t then–part %e else–part %; (I)

expr %t then–part %e else–part %; (E)

%#(...%)	Repeat sequence # times (E)
%l(... %)	Select sequence from a list (E)

The Stack – Internal Processing of Parameters

To understand how to write percent command sequences you need to know how such command sequences are processed. A four-level stack is used to process parameters. Arithmetic and logical operations take their operands from the stack and leave the result on the stack.

Output commands output the value on the top of stack and then remove it.

Arithmetic and logical operations use post-fix (reverse polish) notation. First the operands are pushed onto the stack, then the operation takes place leaving the result on the stack. For example:

```
%p1 %p2 %p3 %+ %*
```

leaves the following result at the top of the stack:

parameter1 * (parameter2 + parameter3)

Note — This mechanism is the same as that used by terminfo. Therefore, terminfo sequences with parameters can be used in Video files without alteration.

PRIFORMA Video files also support the simpler termcap method of sequencing parameters. A current index to the parameter list is maintained and, when a parameter is required on the stack, the current parameter is pushed on the stack and the current index is incremented. If an output command is encountered when there is no value to output from the stack, an automatic push is performed using the current index. The effect of this is that terminfo and termcap sequences can both be used within Video files. For example, both of the following sequences output two parameters with decimal values:

```
%d %d

%p1 %d %p2 %d
```

Examples of Stack Manipulation and Arithmetic Commands

The %p command pushes parameters and constants onto the stack. For example:

%p10	Push parameter 10
%p'y'	Push the character y
%p{10}	Push the number 10
%p{0}	Push binary zero
%p'0'	Push the character 0

There are various logical and arithmetic commands, which require one or two operands on the stack. The following example shows the terminfo and termcap methods of logical ORing three parameters with the asterisk character (*) and outputting the result:

```
%'*' %p1 %| %p2 %| %p3 %| %c

%'*' %| %| %| %c
```

The following example adds the value of the space (SP) character to the parameter and outputs the result. First the parameter is pushed on the stack, then the SP character code (0x20) is pushed. The %+ command adds the value on the top of the stack and the value one level below the top of the stack, and places the result on the top stack.

```
%p1 %'SP' %+ %c

%'SP' %+ %c
```

Examples of Parameter Sequencing Commands

It can sometimes be useful to skip a parameter, or to repeat the same parameter. Parameter sequencing commands let you alter the sequence in which commands are output. The following example outputs the same parameter twice:

```
%p1 %d %p1 %d


%d %b %d
```

The command %u *uses* one parameter, simply incrementing the parameter index. %b *backs up* one parameter, decrementing the parameter index. This example outputs two parameters in reverse order:

```
%p2 %d %p1 %d


%u %d %2b %d
```

Examples of Output Commands

%c, %. and %d output characters and decimal values in a similar way to the C printf library function.

%#z outputs a number of binary zeros and is normally used for time delay. For example %100z sends 100 zero bytes to the terminal.

Examples of Parameter Changing Commands

%i increments the next two parameters and is used almost exclusively in cursor positioning sequences. PRIFORMA line and column parameters are calculated internally relative to (0,0). Many terminals expect cursor positioning to be relative to (1,1). %i increments the line and column parameters, but does not increment the parameter index or output any values. The following example shows how %i is used to define cursor position (CUP) sequence for a PT200 terminal:

```
ESC [ %i %d ; %d H
```

%r reverses the next two parameters and is often used when line and column parameters for cursor positioning need to be given in column and line order. %r can be replaced using explicit parameter pushes as described earlier.

```
FS G %r %c %c
```

Examples of Control Commands

PRIFORMA Video files support three control commands:

- If–then–else construct
- Repeat command
- List command

The if–then–else construct uses %?, %t, and %e to delimit the three expressions as follows:

```
%? expression %t then_expression %e else_expression %;
```

You can omit the %? for brevity, if required.

%t pops the result of the *expression* from the stack and tests it for TRUE/FALSE (using the C conventions for Booleans: zero is FALSE, nonzero is TRUE). The *then_expression* is performed if the result is TRUE. The *else_expression* is performed if the result is FALSE.

If %t finds the stack empty it automatically pushes the next parameter onto the stack and then pops it for evaluation. The parameter index is not incremented until the entire conditional construct has been executed.

The following example tests the value of a parameter. The parameter is only output if its value is not 1.

```
%? %{1} %p1 %= %t %e %p1 %d%;


%{1} %= %t %e %d %;
```

The value binary 1 is pushed on the stack, as is the parameter. The %= compares the two values on the stack and leaves Boolean TRUE or FALSE on the stack. This value is popped from the stack and tested for TRUE/FALSE to determine whether the *then_expression* or the *else_expression* is to be executed. In the above example, the *then_expression* is empty. The *else_expression* outputs the parameter as a decimal.

The if–then–else construct can be used to toggle an attribute on and off. The ESC A sequence is sent if the parameter is nonzero. The ESC B sequence is sent if the parameter is zero.

```
ESC %? %p1 %t A %e B %;
```

```
ESC %t ( %e ) %;
```

Note Nested if–then–else constructs are possible but quickly become difficult to follow. Instead, use the list command described later.

The repeat command performs the same sequence on a number of parameters. This command is intended for automatic parameter sequencing and is almost useless if explicit parameter pushes are used. The repeat command repeats the sequence within the (...%) a number of times defined by the %# in front of the bracket. For example:

```
%3( %d %)
```

outputs three decimal parameters, and

```
%4( %t %d %e %; %)
```

tests each of four parameters in turn and only outputs the tested parameter if it is nonzero.

The list command implements multiple if–then–else constructs in a similar way to select or case structure. List has the following format:

```
%l( value1: expr1 %; value2: expr2 %; ... %)
```

Value1, value2 and so on are single character constants. The value on the stack is popped and tested against value1, value2 and so on. If a match is found, the corresponding expression is processed. Each expression contains an implicit break; that is, only one expression from the list can be processed. The default case (if any) must be specified last in the list by omitting the value. The following example outputs nothing if the value on the stack is 0, outputs the character A if the value is 1, and otherwise (the default case) outputs the ESC character:

```
%l( 0:%; 1:A %; :ESC %)
```

The parameter index is incremented after the entire list is processed, even if the expressions within the list command use parameters.

Video File Keywords

The Video file contains keywords that identify escape sequences that PRIFORMA needs to send to the terminal at runtime, so that video effects, cursor positioning and so on, can be achieved.

This section describes the Video file keywords.

Keyword Summary

This subsection summarizes the keywords used to describe characteristics of a terminal.

Basic Capabilities

LINES	Lines on screen
COLMS	Columns on screen
INIT	Initialization sequence
RESET	Restoration sequence
REPT	Repeat character sequence
REPMAX	Maximum length of repeat
FMKRDS	Draw screen status line text
FMKRTM	Test mode status line text
FMKRCP	Copy field status line text
FMKRMV	Move field status line text
CURPOS	Cursor position display

Erasure Commands

ED	Erase display
EL	Erase to end of line
EW	Erase window

Cursor Control

CON	Turn cursor on
COF	Turn cursor off

SCP	Save cursor position and attribute
RCP	Restore cursor position and attribute

Cursor Position

CUP	Absolute cursor position
CUU	Cursor up
CUD	Cursor down
CUF	Cursor forward
CUB	Cursor backward

Latch Attributes

LATCHATT	List of attributes
SGR	Set graphics rendition
COLOR	List of colors

Area Attributes

AREAATT	List of attributes
ASGR	Area set graphics rendition
ARGR	Area reset graphics rendition

Message Line

OMSG	Open message line
CMSG	Close message line
MSGATT	Message line attributes

Borders

BORDER	Border definitions
BRDATT	Border attributes

Shifting Field Indicators

ARROW	Shift indicators

Block Mode Routine Entry Points

BLK_INIT	Initialize block mode
BLK_RESET	Reset character mode
BLK_LOCK	Lock display attributes
BLK_UNLOCK	Unlock display attributes
BLK_PROT_SCRN	Protect whole screen
BLK_SLA	Set logical attributes
BLK_GET_DATA	Get data from screen
BLK_GET_KEY	Get key for block mode
BLK_CPR	Get cursor position

Basic Capabilities

This subsection describes the Video file keywords that define the basic capabilities of the terminal. Examples given are for a PT200 terminal, unless stated otherwise.

LINES

The number of lines on the form. By default this is 24. One line is normally reserved as a status line. The maximum form size is therefore one less than the value supplied here:

```
LINES = 24
```

COLMS

The number of columns on the screen. The default value is 80.

```
COLMS = 80
```

INIT

The initialization sequence for the terminal. There is no default sequence if this keyword is omitted. This command can be used to change the mode of the terminal, map function keys and so on. Padding commands (%#z) are often required to cause delay while the terminal initializes.

```
INIT = ESC P ! SP ~ 8 & ~ . / 0 ~ R ~ C & ~
       $ ESC \\ ESC [ > 2 1 h
```

RESET

Reset sequence for the terminal. There is no default if this keyword is omitted. The reset sequence should undo the effects of initialization. For many terminals this can mean a hard reset.

```
RESET = ESC ? ESC [ > 2 1 l
```

REPT

The character repeat sequence. If the terminal supports character repeat, this sequence has two parameters: the character to be repeated and the number of times to repeat. PRIFORMA uses this sequence, if defined, to redraw borders and to clear areas of the screen.

```
REPT = %c ESC [ %d b
```

REPMAX

The maximum number of characters that REPT is allowed to repeat.

ED

The sequence required to erase the display and home the cursor. This command must clear all attributes including the background color (PCs only). For example:

```
ED = ESC ?
```

EL

The sequence required to erase from the cursor to the end of the line. If omitted, PRIFORMA erases the line by writing blanks. The following example is for a PC, taking five parameters (start line and column, number of lines and columns, and background color):

```
ESC [ %i %d; %d; %d; %d; %c w
```

CON

The sequence to turn the cursor on. See also COF, below.

```
CON = ESC $ R
```

COF

The sequence to turn the cursor off. If possible the CON and COF sequences should always be given, because PRIFORMA needs to move the cursor around the screen to fill data fields and so on. Screen updates look much better to the end user if PRIFORMA can turn the cursor off during such operations. CON and COF are often referred to as cursor attributes in many terminal manuals.

```
COF = ESC $ S
```

SCP

The sequence to save the current cursor position.

RCP

The sequence to restore the current cursor position. PRIFORMA often needs to temporarily save the cursor position; for example, before updating the status line. Specifying SCP and RCP reduces the output required and therefore, if specified, improve performance. Terminal manuals refer to these sequences in many obscure ways; for example, as cursor description.

CUP

The sequence to specify the absolute cursor position. The sequence takes two parameters: the line and column. These parameters are relative to 0. If your terminal specifies the top left corner as (1,1) then the %i command can be used to increment the parameters, as shown in the example below. Some terminals add a character constant to the line and column parameters. You can use the %+ command to add the constant to each parameter before output.

```
CUP = ESC [ %i %d;%d H
```

CUU, CUD, CUF, CUB

The sequences to move the cursor up, down, forward, or backward. All these sequences take a single parameter which is the number of lines or columns the cursor must move. If these sequences are not provided, PRIFORMA uses the slower CUP command.

```
CUU = ESC [ %d A
CUD = ESC [ %d B
```

```
CUF = ESC [ %d C
CUB = ESC [ %d D
```

Attribute Keywords

PRIFORMA supports the following attributes:

- Highlight
- Blink
- Underline
- Reverse video

If highlight or blink is not available on a given terminal, low intensity is substituted. The keywords LATCHATT and AREAATT list the attributes available and the character associated with each attribute. The keywords SGR and ASGR actually output the character sequence needed to set a particular group of attributes.

PRIFORMA supports three kinds of attribute handling modes commonly found on terminals:

Latch attributes

The most common kind of attribute handling; any characters written to the terminal screen after a latch attribute setting sequence are displayed using those attributes.

Area attributes

The cursor position is important. When the area attribute setting sequence is sent, all characters from the cursor position to the next attribute area (or end of the line or end of the screen) are redisplayed with the new attributes. PRIFORMA positions the cursor to the end of the area to be changed, sets an ending attribute, then positions the cursor to the start of the area and sets the attributes. Attributes do not occupy a screen position (compare with on–screen attributes below).

On-screen attributes

On-screen attributes are like area attributes, but the attribute markers occupy space on the screen. In this mode, fields and display areas cannot be immediately adjacent, as room is needed for the attribute markers. Window display may be hampered by lack of space for attribute markers.

Terminals often support more than one mode. If the terminal supports both area and on–screen attribute handling, you are advised to select area attribute handling. Some terminals support one latch attribute and several area attributes simultaneously.

If the terminal can only display one attribute at a time, but that attribute can be selected from a list, you are recommended to specify reverse video.

LATCHATT

Specifies the characters that SGR outputs to set attributes. The format is given below. Attributes can be REVERSE, BLINK, UNDERLN, HILIGHT and DIM (low intensity). If the value for any attribute is missing, the attribute is treated as a Boolean and is assigned the value binary 1.

```
LATCHATT = attrib1 = value1 attrib2 = value2 ...
```

The LATCHATT sequence for a PT200 terminal is

```
LATCHATT = REVERSE = 7 DIM = 2\
           UNDERLN = 4 BLINK = 5
```

AREAATT

Specifies the characters that ASGR outputs to set particular attributes for area attribute mode terminals. This is similar to LATCHATT. Additionally a number of flags can be set indicating how area attributes are implemented by the terminal. The flags required are added to the end of the attribute list, and are as follows:

ONSCREEN

The attribute uses a screen position (see description of on–screen mode earlier).

LINEWRAP

The attribute wraps over line boundaries.

SCREENWRAP

The attribute wraps from the bottom of the screen to the top.

REWRITE

Rewrite the attribute when writing a character.

MAX = #

The maximum number of attributes per line.

For example, on a Wyse 75 terminal:

```
AREAATT = BLINK = 2 DIM = p REVERSE = 4\
          UNDERLN = 8 ONSCREEN LINEWRAP
```

SGR

Set Graphics Rendition sequence. The character sequence output used to set a particular group of attributes on a latch attribute mode terminal. SGR has the following parameters:

- Standout – not supported, always 0
- Underline
- Reverse video
- Blink
- Dim (low intensity)
- Highlight (bold)
- Blank – supported by software, always 0
- Protect – supported by software, always 0
- Alternate char – supported in other sequences, always 0
- Foreground color – if available
- Background color – if available

SGR defines the character sequence for setting terminal attributes. The attribute values are defined by the LATCHATT keyword, for example:

```
LATCHATT = REVERSE = 7 DIM = 2\
           UNDERLN = 4 BLINK = 5
```

The SGR command sequence must test the parameters passed to it and generate the character sequence to set the required attributes on the terminal.

For a PT200 terminal the character sequence

```
ESC [0 ; a ; b ; c ... m
```

sets the attribute values *a*, *b*, *c*, and so on, where *a*, *b*, and *c* are attribute values defined by the LATCHATT keyword. For example, the following character sequence sets underline and blinking attributes on a PT200 terminal:

```
ESC [0 ; 4 ; 5 m
```

To generate a given set of attributes, PRIFORMA passes 9 parameters (or 11 parameters if color is supported) to SGR. Each parameter is zero if the attribute is not to be set, or not supported. Otherwise the parameter contains the attribute code defined by the LATCHATT keyword. For example, to set underline and blinking attributes on a PT200, PRIFORMA would send the following 9 parameters:

```
0, 4, 0, 5, 0, 0, 0, 0, 0
```

The SGR command sequence required for a PT200 terminal tests each parameter received. If the parameter is non–zero then the parameter value is sent to the terminal. The SGR command sequence

```
SGR = ESC [0 %9(%t ; %c %; %) m
```

performs this function. This sequence would take the 9 parameters given in the earlier example and output

```
ESC [0 ; 4 ; 5 m
```

ASGR

The set area graphics rendition sequence. This works in a similar way to SGR, but is for area attribute mode terminals. ASGR does not support color, therefore PRIFORMA passes nine parameters only.

ARGR

The remove area graphics rendition sequence, for area attribute mode terminals. ARGR causes PRIFORMA to remove attributes (for example, within a field), so that the number of attributes on a line does not exceed the maximum. The remove area sequence does not stop the propagation of a previous attribute (unlike the set area sequences, which do stop the propagation of a previous attribute).

Use of Color

PRIFORMA supports color as latch attributes only. If color is available on the terminal, the COLOR keyword should be used to associate a character with each supported color (in a similar way to the way LATCHATT associates a character with each video attribute). The primary colors must be specified, at least. PRIFORMA can generate other colors from the primary colors as follows:

BLACK	BLUE & GREEN & RED
WHITE	BLUE | GREEN | RED
MAGENTA	BLUE | RED
CYAN	BLUE | GREEN
YELLOW	RED | GREEN

where & is logical AND, and | is logical OR.

The flag BACKGRND can also be added to the COLOR keyword, indicating that the terminal supports background color (for example, the PC).

Where the COLOR keyword is included in the Video file, PRIFORMA passes SGR 11 parameters, rather than 9 in the case of monochrome terminals, and the SGR sequence must be changed to take account of this. The tenth and eleventh parameters are the foreground color and background color. The following example is for a PT200–C terminal:

```
COLOR = CYAN = 0 MAGENTA = 1 BLUE = 2\
        YELLOW = 3 GREEN = 4 RED = 5\
        BLACK = 6 WHITE = 7
```

Message Line Keywords

PRIFORMA takes a line from the number of lines specified by the LINES keyword and uses this as a status/message display line. For a terminal where LINES = 24, 23 lines are available for form display, and 1 line is used for the message line.

Some terminals have a special message line that cannot be addressed by normal cursor positioning. If this is the case, two sequences need to be defined to open the message line so that it can be written to, and to close the message line. The OMSG and CMSG keywords define the escape sequences required to perform these two functions.

If the OMSG keyword is present in the Video file, then all lines defined by the LINES keyword are available for form display.

The keyword MSGATT lets you define which attributes are available on the message line as follows:

```
MSGATT = NONE

MSGATT = LATCH

MSGATT = AREA
```

The above examples define the message line attributes as no attributes, attributes as defined by LATCHATT, or attributes as defined by AREAATT. Additionally, if the message line has nonembedded attributes the flag ONSCREEN can be added:

```
MSGATT = AREA ONSCREEN
```

Often a terminal can have embedded attributes on the main screen and nonembedded attributes on the message line.

Form Borders

The BORDER keyword specifies the characters used for PRIFORMA form borders. A maximum of 10 border styles can be defined (applications refer to these styles using the numbers 0 – 9). Style 0 is usually reverse video spaces, but is replaced by I characters if reverse video is not available on the terminal. The ten default styles are

```
0.      IIIII          1.      ___
        I   I                 |   |
        IIIII                 |___|

2.      +++++          3.      ===
        +   +                 |   |
        +++++                 ===

4.      %%%%%          5.      .....
        %   %                 :   :
        %%%%%                 :...:

6.      *****          7.      \\\\\
        *   *                 \   \
        *****                 \\\\\

8.      /////          9.      #####
        /   /                 #   #
        /////                 #####
```

The BORDER keyword lets you specify other border styles by defining the character to be used in each of eight positions around the border:

- Upper left corner
- Top row
- Upper right corner
- Left side
- Right side
- Lower left corner
- Bottom row
- Lower right corner

The default border set could be defined as follows:

```
BORDER =  SP  SP  SP  SP  SP  SP  SP  SP  \
          SP  _   SP  |   |   |   _   |   \
          +   +   +   +   +   +   +   +   \
          SP  =   SP  |   |   SP  =   SP  \
          %   %   %   %   %   %   %   %   \
          .   .   .   :   :   :   .   :   \
          *   *   *   *   *   *   *   *   \
          \   \   \   \   \   \   \   \   \
          /   /   /   /   /   /   /   /   \
          #   #   #   #   #   #   #   #
```

The keyword BRDATT can be used to specify the attributes available when drawing form borders. BRDATT can be specified as

```
BRDATT = NONE

BRDATT = LATCH

BRDATT = AREA
```

Note If the terminal uses on-screen attributes, or has a limit to the number of attributes per line, it is best to prohibit border attributes as follows:
BRDATT = NONE.

Shifting Field Indicators

Shifting indicators are used to indicate that there is more data off-screen to the left or right of the visible portion of a field. When two shifting fields are adjacent there may not be sufficient space to display separate shifting indicators. In this case a combined indicator character is used.

The ARROWS keyword defines the characters used for left, right, and combined shifting field indicators. The default indicators are <, >, and X.

```
ARROWS  = < > X
```

DRAW_FORM Status Text

The following keywords are used to define status line and message text used within the DRAW_FORM utility:

FMKRDS	The status line text when DRAW_FORM is in DRAW mode
FMKRTM	The status line text when DRAW_FORM is in TEST mode
FMKRCP	Copy field help text
FMKRMV	Move field help text

The text to be used can simply be enclosed in quotes. Alternatively, attributes such as reverse video can be embedded. The following example shows the embedded attribute sequences used to turn reverse video on and off:

```
FMKRDS = 0x80 0x20 0x81 1: 0x80 0x20 0x07 DRAW/test SP\
         0x80 0x20 0x81 2: 0x80 0x20 0x07 form SP \
         0x80 0x20 0x81 3: 0x80 0x20 0x07 field SP\
         0x80 0x20 0x81 4: 0x80 0x20 0x07 delete SP\
         0x80 0x20 0x81 5: 0x80 0x20 0x07 move SP\
         0x80 0x20 0x81 6: 0x80 0x20 0x07 copy SP\
         0x80 0x20 0x81 7: 0x80 0x20 0x07 repeat SP


FMKRTM = 0x80 0x20 0x81 1: 0x80 0x20 0x07 draw/TEST SP\
         0x80 0x20 0x81 2: 0x80 0x20 0x07 form SP \
         0x80 0x20 0x81 3: 0x80 0x20 0x07 field SP\
         0x80 0x20 0x81 4: 0x80 0x20 0x07 delete SP\
         0x80 0x20 0x81 5: 0x80 0x20 0x07 move SP\
         0x80 0x20 0x81 6: 0x80 0x20 0x07 copy SP\
         0x80 0x20 0x81 7: 0x80 0x20 0x07 repeat SP


FMKRCP = 0x80 0x20 0x81 6: 0x80 0x20 0x07 \
"COPY use arrow keys to position, F6 to release"


FMKRCP = 0x80 0x20 0x81 5: 0x80 0x20 0x07 \
"MOVE use arrow keys to position, F5 to release"
```

Cursor Position Display

If an end user is typing at the keyboard, PRIFORMA only displays the cursor when a pause is detected. The CURPOS keyword lets you define how long a delay should be before the cursor is redisplayed. You should set this parameter so that a fast typist is not slowed down. CURPOS specifies the delay as an integer in tenths of a second. The choice of the best delay value depends on the baud rate and terminal itself, and it should not impair typing speeds. Example values are

CURPOS = 1 For use with a fast system, update cursor every 1/10th sec.

CURPOS = 3 For use with most systems

CURPOS = 7 For low baud rate systems

CURPOS = 0 For no cursor display (or omit CURPOS keyword)

Compiling a Video File

Before you can use the Video file, you need to use COMPILE_FORMTERM to compile the ASCII file and generate a binary version. To use COMPILE_FORMTERM type

```
COMPILE_FORMTERM termVIDEO
```

where *term*VIDEO is the name of the ASCII file you have created for your terminal. If you omit the Video suffix, this is automatically appended by the utility before attempting to open the file. COMPILE_FORMTERM creates a binary video file, *term*VIDEO.BIN. COMPILE_FORMTERM is abbreviated CFTRM.

For example, to compile a video file for a PT200 terminal, type

```
COMPILE_FORMTERM PT200VIDEO
```

This creates the file PT200VIDEO.BIN.

Before PRIFORMA can recognize your Video file, you must make an entry in the SMVARS file, as described earlier in this chapter in the section entitled Terminal Administration.

Appendices

ASCII Control Codes

A

Mnemonic	*Decimal*	*Hexadecimal*
NUL	00	00
SOH	01	01
STX	02	02
ETX	03	03
EOT	04	04
ENQ	05	05
ACK	06	06
BEL	07	07
BS	08	08
HT	09	09
LF	10	0A
VT	11	0B
FF	12	0C
CR	13	0D
SO	14	0E
SI	15	0F
DLE	16	10
DC1	17	11
DC2	18	12
DC3	19	13
DC4	20	14
NAK	21	15

Mnemonic	*Decimal*	*Hexadecimal*
SYN	22	16
ETB	23	17
CAN	24	18
EM	25	19
SUB	26	1A
ESC	27	1B
FS	28	1C
GS	29	1D
RS	30	1E
US	31	1F

Rev. 23.3 Publications

B

This appendix lists all books that are integral to Master Disk Revision 23.3. For any books not related to a specific revision, see the *Guide to Prime User Documents* (DOC13079–2PA), 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 as follows:

- 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 (RLN*nnnnn–nn*A) or an update (UPD*nnnnn–nn*A), you must also have the last full edition of the title (DOC*nnnnn–nn*A) and any updates to that edition that were published prior to the publication in the *Most Recent* column. If the *Most Recent* publication is a full edition (DOC*nnnnn–n*LA), 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 B–1. PRIMOS Administration and Operation				
Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
Rev 23.3 Software Release Document	23.3	DOC13134–1PA	23.2 23.1 23.0	DOC10001–9PA DOC10001–8PA DOC10001–7PA
RAS Guide for 50 Series System Administrators	23.3	DOC13156–1LA		
Operator's Guide to System Commands	23.3	DOC9304–6LA		
Operator's Guide to Data Backup and Recovery	23.3	UPD10324–12A	23.1 23.0	UPD10324–11A DOC10324–1LA
Operator's Guide to File System Maintenance	23.3	DOC9300–6LA		
Operator's Guide to the Spooler Subsystem	23.3	DOC9303–5LA		
Operator's Guide to System Monitoring	21.0	DOC9299–3LA		
Operator's Guide to the Batch Subsystem	21.0	DOC9302–3LA		
Operator's System Overview	22.0	DOC9298–3LA		
Operator's Master Index	23.3	DOC10110–5LA		
DSM User's Guide	23.0	DOC10061–3LA		
Rev. 23.0 Software Installation Guide	23.0	IDR10176–3XA		
Using FS_RECOVER	23.3	UPD13062–31A	23.2	DOC13062–3LA
MAGNET User's Guide	22.0	UPD10156–11A	21.0	DOC10156–1LA
Prime 8mm Cartridge Tape Drive User's Guide	23.3	DOC10275–3LA		
Disk Replacement Procedure for the Model 75500–6PK Device Module	23.2	IDR13100–1XA		

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

Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
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		
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	22.1.4	DOC10356–2LA		
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

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

Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
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
Using Your 4150	21.0.2	DOC10163–1LA	22.0	RLN10239–001
Release Note for 6250–bpi Tape Dump	22.1	RLN12005–1XA		

Table B–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 B–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	23.3	DOC3108–8LA		
New PRIMOS Help	23.0	DOC20012–1PA		
CPL Programmer's Companion	19.3	FDR7811–193		
CPL User's Guide	21.0	DOC4302–3LA		
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		

Table B–3. PRIMOS Use and Programming (continued)				
Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
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 B–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		

Table B–4. Communications (continued)

Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
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
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 B–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		
PRIFORMA Forms Design and Administration 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	v6.0.30	DOC10134–4LA		
PRISAM Programmer's Companion	21.0	DOC10088–1XA		

Table B–5. Data Management (continued)				
Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
PRISAM User's Guide	22.0	DOC7999–4LA		
ROAM Administrator's Guide	22.0	UPD7345–31A	21.0	DOC7345–3LA

Table B–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
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		

Table B–6. Languages (continued)

Book Title	*Most Recent*		*Also Required*	
	Revision	*Document Number*	*Revision*	*Document Number*
PL/I Conversion Guide: IBM PL/I to Prime PL/I	19.4	DOC5769–1LA		
PL/I Reference Guide	T1.0–21.0	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 B–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		

Master Index

Master Index to Software Release Documents

Symbols

! character, before commands in FS_RECOVER: *23.1* 4-14

@ character
- in user login: *23.3* 3-12
- WATCH monitoring: *23.1* 2-11
- wildcard: *23.1* 2-14, 2-18

< character, root directory: *23.0* 2-17

Numbers

10019 controller. *See* Disk and tape controllers

2382 controller. *See* Disk and tape controllers

4005 mode. *See* Disk and tape controllers

4594 tape drive: *23.2* A-6

4595 tape drive: *23.2* A-3; *23.3* 3-4

4596 tape drive: *23.2* A-3; *23.3* 3-4

4598 tape drive: *23.2* A-3; *23.3* 3-4

4601 tape drive: *23.3* 3-4

4721 disk drive. *See* SCSI disks

4729 disk drive. *See* SCSI disks

4730 disk drive. *See* SCSI disks

4731 disk drive. *See* SCSI disks

4732 disk drive. *See* SCSI disks

5520 computer: *23.3* 3-1

5540 computer: *23.3* 3-1

6580 controller. *See* Disk and tape controllers

7210 controller. *See* Disk and tape controllers

75500–6PK device module. *See* SCSI disks

A

Access categories, for sharing memory: *23.1* 2-20

Access Control Lists. *See* ACL protection

ACL groups
- adding: *23.0* 2-1, 2-38
- .BACKUP$: *23.0* 2-6
- changing projects: *23.2* 2-3
- listing project: *23.0* 2-38
- listing system: *23.0* 2-38
- listing user's: *23.1* 2-13
- .RAS$: *23.1* 3-10; *23.3* 3-46
- .SPOOL_ADMINISTRATOR$: *23.3* 3-25
- .SYSTEM_ADMINISTRATOR$: *23.0* 2-31
- validating IDs: *23.1* 2-17
- .WATCH$: *23.1* 2-10

ACL protection
- displaying at prompt: *23.1* 2-6
- FS_RECOVER: *23.1* 4-9
- MFD: *23.0* 2-2
- printer environments: *23.3* 3-25
- root directory: *23.0* 3-5
- shared memory: *23.1* 2-20
- System Administrator rights: *23.0* 2-2
- SYSTEM_DEBUG* directory: *23.1* 4-9

ADD_PORTAL command: *23.0* 2-21

Added disks
- crash dump disk restriction: *23.2* 3-25, 3-34
- directory other than root: *23.0* 2-18
- DISK_PAUSE restriction: *23.2* 3-25
- dual-porting: *23.2* 3-28
- error messages: *23.1* C-1
- logical mounts: *23.0* 2-18
- maximum number of: *23.2* 3-20
- remote disks: *23.0* 2-16
- SPIN_DOWN restriction: *23.2* 3-24
- split SCSI disks: *23.2* 3-23

ADDISK command
- *See also* Added disks
- –MOUNT_PATH option: *23.0* 2-18, 2-22
- –PRIVATE option: *23.0* 2-22
- –REPLACE option: *23.0* 2-22

AMLBUF directive: *23.2* 1-4; *23.3* 1-4

AMLC command: *23.2* 1-4; *23.3* 1-4

ASCII control codes: *23.3* A-1

ASSIGN command
- –FORMAT option: *23.3* 3-4
- model 4601 tape drive: *23.3* 3-4
- tape drive error: *23.3* 3-6

Assigned disks
- DISK_PAUSE restriction: *23.2* 3-25
- listing: *23.3* 3-46
- maximum number of: *23.2* 3-20
- SPIN_DOWN restriction: *23.2* 3-24

ASYNC environment directive, –MAXIMUM_SHUTDOWN_TIME option: *23.1* 3-10

Asynchronous lines, disallowing login if no carrier: *23.1* 3-13

AT$ABS subroutine: *23.0* 3-4

AT$ANY subroutine: *23.0* 3-5

AT$ROOT subroutine: *23.0* 3-5

Attach points
- changing projects: *23.2* 2-3
- current: *23.0* 3-5; *23.1* 2-13, 2-18
- defining: *23.0* 2-2
- determining pathnames: *23.0* 3-6
- displaying at prompt: *23.1* 2-5
- home: *23.0* 3-5; *23.1* 2-18
- initial: *23.1* 2-18
- multiple: *23.1* 2-18

origin: *23.0* 2-2; *23.1* 2-13; *23.2* 2-3
returned by function: *23.1* 2-13
setting to MFD directory: *23.0* 3-4, 3-5
setting to root directory: *23.0* 3-5
ATTACH$ search rule list: *23.0* 2-24
ATTACH_POINT function: *23.1* 2-13
ATTRIB function: *23.1* 2-23
-TRUNC option: *23.3* 2-13
ATTRIBUTE environment directive
-MANDATORY option: *23.3* 3-24
HP LaserJet options: *23.3* 3-23
Automated recovery. *See* System Recovery facility
AUTOPSY
run file: *23.1* 4-9
search rules file: *23.1* 4-9

B

Backup and recovery subsystem: *23.1* 3-13
See also MAGSAV/RST subsystem
ICOP+: *23.2* 3-26
BATCH command, -START DISPLAY_ALL option: *23.1* 3-6
Batch subsystem: *23.0* 2-35; *23.1* 3-6; *23.3* 3-21
BATCH -START -DISPLAY_ALL: *23.1* 3-6
batch queue: *23.0* 2-36
display all: *23.0* 2-35; *23.3* 2-4, 3-21
display today's: *23.3* 3-21
display user's: *23.3* 3-22
FIXBAT: *23.0* 2-35; *23.1* 3-6
INIT -SEM_INIT: *23.1* 3-6
project assignment: *23.2* 2-3
queue full: *23.3* 3-22
reinitializing semaphores: *23.1* 3-6
restarting: *23.1* 3-6
viewing job queues: *23.1* 3-6; *23.3* 3-21
BATCH_SERVICE utility: *23.1* 3-6
Boot. *See* Cold starts
Boot tape, creating: *23.0* 2-6
BOOT, VCP command: *23.2* 3-2
BOOTP, VCP command: *23.2* 3-2
BOOTQ, VCP command: *23.2* 3-1
BRMS (obsolete): *23.0* 1-4

C

C compiler
installation: *23.0* 1-2
opening files: *23.1* 2-23
passing array pointers: *23.1* 2-23
CAM files
available space: *23.0* 2-21
following system crash: *23.1* 4-8
truncating: *23.1* 2-24
Carriage return
in prompts: *23.1* 2-6
redisplaying login prompt: *23.1* 3-2
when spooler starts: *23.1* 3-10
Carrier signal: *23.1* 3-13
CBL compiler, 32IX mode: *23.1* 2-23
CDD command: *23.2* 3-36
-INFO options: *23.3* 3-28
status codes: *23.2* 3-40
CF$EXT subroutine: *23.1* 2-24
CHANGE_PROJECT command: *23.2* 2-3
enabling/disabling: *23.2* 2-5, 3-18
Character strings
See also Non-printable characters
collating sequences: *23.1* 2-25
maximum line length: *23.1* 2-25
reversing: *23.1* 2-17
validating numeric: *23.1* 2-17
CHG$PW subroutine, error messages: *23.3* 2-23
Child processes, listing: *23.3* 2-4
CHPRJ$ subroutine: *23.2* 2-4
CLOSE command, dropping remote line: *23.2* 3-17
CNAME command
-FORCE option: *23.3* 3-20
-REPORT option: *23.1* 2-1
COBOL. *See* CBL compiler
Cold starts
automated execution: *23.2* 3-50, 3-51
boot data switches: *23.2* 3-2
boot options: *23.2* 3-2
boot requirements: *23.0* 1-1
boot sense switches: *23.2* 3-2
BOOTP command: *23.2* 3-2
controller down-line load: *23.2* 3-28
crash recovery: *23.1* 4-1
diagnostic processing: *23.2* 3-1
disk drive for boot: *23.2* 3-2
disks inactive since: *23.1* 2-14
during PRIMOS installation: *23.1* 1-1; *23.2* 1-1; *23.3* 1-3
FS_RECOVER invocation: *23.1* 4-10
hang during Quick Boot: *23.2* 3-3
ICOP+ implementation: *23.2* 3-27
paging allocation checked: *23.0* 2-26
parallel shares: *23.2* 3-3
pausing PRIMOS.COMI: *23.1* 4-10
PNX activation: *23.2* 4-5, 4-9
PNX messages: *23.2* D-1
Quick Boot: *23.2* 3-1
required after RFS: *23.1* 4-3
resetting boot defaults: *23.2* 3-2
resetting system recovery: *23.2* 3-46
setting login prompt: *23.1* 3-3
setting remote login retries: *23.2* 3-16
system name prompt: *23.3* 3-7
time since: *23.1* 2-15
WATCH access re-enabled: *23.1* 2-9
COMDEV
damaged by crash: *23.1* 4-8
DISK_PAUSE restriction: *23.2* 3-25
dual-porting restriction: *23.2* 3-28
mirroring: *23.2* 3-24, 3-25
repairing using FIX_DISK: *23.1* 4-8
saving to tape: *23.0* 2-9
SPIN_DOWN restriction: *23.2* 3-24
SYSTEM_DEBUG*: *23.1* 4-8
COMM_CONTROLLER command: *23.2* 3-5
Command device. *See* COMDEV
Command functions. *See* Functions
Command line
See also ECL
COMO command options: *23.2* 2-1
functions: *23.1* 2-12
prompts: *23.1* 2-4
specifying CHANGE_PROJECT: *23.2* 2-3
within FS_RECOVER: *23.1* 4-14
Commands
See also PRIMOS commands; VCP commands
COMM_CONTROLLER: *23.2* 3-5
SHARE: *23.2* 3-5
Common file system name space: *23.0* 2-19
accessing objects outside: *23.0* 2-20
advantages of: *23.0* 2-16
Communications controllers
ICS3 controller: *23.3* 3-48

LHC controller: *23.1* 3-12; *23.2* 4-9
LTS controller: *23.2* 5-7
PNC-II controller: *23.2* 4-5
Communications lines
56KB lines: *23.3* 3-48
dislog: *23.2* 5-7
COMO –NTTY command
–QUERY prompts not displayed: *23.2* 2-2
returning current output state: *23.1* 2-13
setting: *23.2* 2-2
WATCH session: *23.1* 2-12
COMO –QUERY command: *23.2* 2-1
multiple options: *23.2* 2-1
undisplayed prompts: *23.2* 2-2
COMO files
crash recovery session: *23.1* 4-17
duplicating file unit info: *23.1* 2-18
FIX_DISK: *23.1* 4-7
FS_RECOVER: *23.1* 4-16, 4-17
overwriting existing: *23.2* 2-1
returning current state: *23.1* 2-13
WATCH session: *23.1* 2-10
COMO_INFO function: *23.1* 2-13
CONFIG directives
LOUTQM: *23.3* 3-15
NLBUF: *23.3* 3-2, 3-3
NRUSR: *23.3* 3-3
NSEG: *23.1* 3-16; *23.3* 3-2
NSLUSR: *23.2* 4-6, 4-7, 4-8
NVMFS: *23.3* 3-2
TPDUMP (obsolete): *23.2* 1-4; *23.3* 1-4
CONFIG file. *See* Configuring
CONFIG_USERS: *23.0* 2-1; *23.2* 3-9
command interface: *23.0* 2-1
concurrent logins: *23.2* 3-10, 3-12, 3-13
data structures for: *23.2* 3-12
disabling user ID: *23.2* 3-10, 3-14
force password change: *23.2* 3-10, 3-13
internal command interface: *23.0* 2-4
logging login attempts: *23.3* 3-9
multiple failed logins: *23.2* 3-10, 3-12
password changes per hour: *23.2* 3-9, 3-13
password history option: *23.2* 3-9, 3-13
password reuse restriction: *23.2* 3-9, 3-13
screen interface: *23.0* 2-1; *23.3* 4-1
status messages: *23.0* 2-3
subroutines: *23.0* 2-38; *23.2* 3-11
terminal definition: *23.3* 4-1
Configuring
adding a user: *23.0* 2-2
common name space: *23.0* 2-19
DTAR 2 memory sharing: *23.1* 2-22
FS_RECOVER segments: *23.1* 4-9
ISC sessions: *23.0* 2-11
login prompt: *23.1* 3-1
multiple tape controllers: *23.1* 3-17
number of user segments: *23.1* 3-16
physical memory: *23.0* 1-1
PNX: *23.2* 4-6
PNX error messages: *23.2* D-1
projects: *23.0* 2-3
slave processes: *23.2* 4-6, 4-7, 4-8
synchronizers and timers: *23.0* 2-11
System Administrators: *23.0* 2-4
user logins: *23.2* 3-9
user password: *23.0* 2-2
Configuring users. *See* CONFIG_USERS; EDIT_PROFILE utility
Control codes
during FS_RECOVER: *23.1* 4-13
listed: *23.3* A-1
WATCH monitoring: *23.1* 2-12
Controllers. *See* Communications controllers; Disk and tape controllers
COPY_DISK command (obsolete): *23.0* 1-4
CP> prompt. *See* Maintenance Processor; VCP commands
CPBOOT file: *23.2* 3-2
CPL
FIX_DISK, automated: *23.1* 4-16
FIX_DISK, running: *23.1* 4-7
for running MAKE: *23.3* 3-40
for running SYSTEM_RECOVER: *23.2* 3-51
functions: *23.1* 2-12, 2-23
CPL files
FED.CPL: *23.3* 1-3
FIX_DISK_MONITOR.CPL: *23.1* 4-16
FS_RECOVER.INSTALL.CPL: *23.1* 4-8
INIT_RECOVER.CPL: *23.1* 4-10; *23.3* 3-42
RUN_FIX_DISK.CPL: *23.1* 4-16
CPMPC command (obsolete): *23.0* 1-4
CPU
3-digit systems: *23.2* 1-4; *23.3* 1-1
crash dump sizing: *23.3* 3-32
displayed by CPUID$: *23.2* 5-1; *23.3* 3-1
MASTER CLEAR button: *23.2* B-3
Model 2250: *23.2* 1-4; *23.3* 1-1
Model 5520: *23.3* 3-1
Model 5540: *23.3* 3-1
power supply: *23.3* 3-4
returning processor type: *23.1* 2-14
supporting automated cold start: *23.2* 3-51
supporting PNX: *23.2* 4-4
supporting Quick Boot: *23.2* 3-1
supporting system recovery: *23.2* 1-3; *23.3* 1-2
CPU time
displaying at prompt: *23.1* 2-5
since boot: *23.1* 2-15
since login: *23.1* 2-16
system meters: *23.1* 3-12
CPUID$ subroutine: *23.2* 5-1; *23.3* 3-1
Crash disk. *See* Crash dump disks
Crash dump disks
activating: *23.2* 3-33, 3-36
checking if activated: *23.2* 3-40
cold start: *23.2* 3-36
controller support: *23.2* 3-34
creating: *23.2* 3-41; *23.3* 3-38
deactivating: *23.2* 3-36, 3-40
disk type: *23.2* 3-34
DISK_PAUSE restriction: *23.2* 3-25
file system portion: *23.2* 3-34
FS_RECOVER access: *23.2* 3-35, 3-44; *23.3* 3-42
full dump: *23.2* 3-38, 3-41
MAKE requirements: *23.2* 3-34
maximum size: *23.3* 3-38
minimum size: *23.2* 3-41
paging restriction: *23.2* 3-34
partial dump: *23.2* 3-38, 3-41
partitioned SCSI disk: *23.2* 3-23
pre-existing dump: *23.2* 3-37, 3-39, 3-40
recovering dump from: *23.2* 3-35, 3-38, 3-43

size of: *23.2* 3-41; *23.3* 3-38
sizing: *23.3* 3-28
SPIN_DOWN restriction: *23.2* 3-24
status: *23.2* 3-39
system recovery: *23.2* 3-49
SYSTEM_RECOVER: *23.2* 3-51
unrecovered dump: *23.2* 3-38, 3-40, 3-41
Crash dump to disk: *23.2* 3-33
See also CDD command
aborting: *23.2* B-6
activating a disk: *23.2* 3-33
analyzing: *23.2* B-7
automated execution: *23.2* 3-49
disk too small: *23.2* 3-35
disk type restrictions: *23.2* 3-34
error messages: *23.2* B-5
example: *23.2* B-2
full dump: *23.2* 3-38
in progress: *23.2* B-5
MAPS information: *23.2* 3-33
operator instructions: *23.2* B-2
partial dump: *23.2* 3-38
retrying: *23.2* B-6
Crash dump to tape
6250 bpi: *23.1* A-2; *23.2* A-1, A-6
automated execution: *23.2* 3-50
checking tape drive: *23.1* 4-15
error messages: *23.2* A-4
example: *23.2* A-2
full dump: *23.2* 3-20
initiating partial dump: *23.1* A-2; *23.2* A-3
labeling tapes: *23.1* A-3; *23.2* A-7
logging: *23.2* 3-42
MAPS information: *23.2* 3-33
messages during: *23.2* 3-20
multi-reel: *23.1* 4-16, A-2; *23.2* A-7
operator instructions: *23.2* A-1, A-3
partial dump: *23.2* 3-20, A-3
performing: *23.1* A-1; *23.2* 3-19, A-1
reading tapes: *23.1* 4-14, 4-16
size of: *23.1* A-2; *23.2* 3-42, A-7
tape reels: *23.1* A-2; *23.2* A-7
tape unit number: *23.2* 3-20
when disk dump fails: *23.2* 3-35
Crash dumps
See also Crash dump to disk; Crash dump to tape
allocating records for: *23.1* 4-10; *23.2* 3-41
analyzing: *23.1* 4-16
calculating record requirements: *23.1* 4-11
CORE dump: *23.2* 3-42
creating files: *23.1* 4-15
disk space for analysis: *23.1* 4-11
file pathname: *23.1* 4-17
full: *23.1* 4-11; *23.2* 3-41
MAPS dump: *23.2* 3-42
partial: *23.1* 4-11; *23.2* 3-41
performing: *23.1* 4-1, 4-12
PIOS dump: *23.2* 3-42
size of: *23.2* 3-41
Crash recovery tools
CDD: *23.2* 3-36
FS_RECOVER: *23.1* 4-4
Quick Boot: *23.2* 3-1
RECORD_TO_PATH: *23.1* 3-10
RFS: *23.1* 4-2
SYSTEM_RECOVER: *23.2* 3-47
Crashes. *See* System crashes
CREATE -REPORT command: *23.1* 2-1
CRMPC command (obsolete): *23.0* 1-4
CUS$ subroutines
listed: *23.0* 2-38
PRIMOS_attributes structure: *23.2* 3-13
PRIMOS_system_attributes structure: *23.2* 3-12; *23.3* 2-20
status codes: *23.2* 3-14

D

Data accessibility, dual-ported disks: *23.2* 3-27
Data integrity
duplicate disk data: *23.2* 3-27
following system crash: *23.1* 4-3
Date
displaying at login: *23.1* 3-1
displaying at prompt: *23.1* 2-5
modified by backup: *23.0* 2-6
modified by restore: *23.0* 2-7
validating: *23.1* 2-17
DBG, UNSPEC function: *23.3* 2-19
DBMS, in PRIMOS.COMI: *23.2* 3-3
DCD carrier signal: *23.1* 3-13
DECODENET: *23.2* 1-4, 3-1; *23.3* 1-2
Despooler
packet size: *23.3* 3-26
phantom: *23.1* 3-10
setting shutdown interval: *23.1* 3-10
setting sleep period: *23.1* 3-9
DEVICE environment directive
-DELAY option: *23.3* 3-22
-POSTSCRIPT_CHECK option: *23.3* 3-24
-POSTSCRIPT_OVERRIDE option: *23.3* 3-24
DIR$LS subroutine, dir_entry structure: *23.3* 2-14
DIR$RD subroutine, dir_entry structure: *23.3* 2-14
DIR$SE subroutine
dir_entry structure: *23.3* 2-14
selection_criteria structure: *23.3* 2-14
Directories
See also Attach points
converting to portal: *23.0* 3-6
creating: *23.1* 2-1
date/time attributes: *23.0* 2-6, 2-7
displaying ACL protection: *23.1* 2-6
listing contents: *23.1* 2-2
opening: *23.0* 3-7
portals: *23.0* 2-20
records used: *23.1* 2-4
root: *23.0* 2-17
size of: *23.2* 5-2
DISCOVER_DBMS, in PRIMOS.COMI: *23.2* 3-3
Disk and tape controllers
10019 for crash dump disk: *23.2* 3-34
10019 in ICOP node: *23.2* 3-26
10019 not in DBS mode: *23.2* 3-32
2382 with 6250 bpi tape drive: *23.2* A-6
6580 dynamic badspot handling: *23.3* 3-40
7210 controller mode: *23.2* 3-22
7210 disk interleaving: *23.2* 3-22
7210 DISK_PAUSE support: *23.2* 3-25
7210 for crash dump disk: *23.2* 3-34
7210 for dual-porting: *23.3* 1-1
7210 in 4005 mode: *23.2* 3-27, 3-32
7210 in ICOP+ mode: *23.2* 3-26, 3-31
7210 microcode loading: *23.2* 3-27, 3-28
7210 pre-Rev. 23.2: *23.2* 3-27
7210 SPIN_DOWN support: *23.2* 3-24

7210 with 75500-6PK disks: *23.2* 3-24, 3-25
7210 with tape drive: *23.2* 3-27, A-6; *23.3* 3-4
address: *23.3* 3-42, 3-44
configuring tape record size: *23.1* 3-17
never used: *23.1* 2-14
number on system: *23.1* 2-14
Quick Boot restrictions: *23.2* 3-3
Disk drives
See also SCSI disks
1.34 gigabyte drive: *23.2* 3-21
421 megabyte drive: *23.2* 3-21
list of: *23.2* 5-10
list of supported: *23.2* 3-21
Model 4721: *23.2* 3-22
Model 4729: *23.2* 3-22
Model 4730: *23.2* 3-22
Model 4731: *23.2* 3-21, 3-22
Model 4732: *23.2* 3-21, 3-22
never used: *23.1* 2-14
number on system: *23.1* 2-14
SCSI: *23.2* 3-22
unit numbers: *23.2* 5-9
usable for crash dump: *23.3* 3-30
Disk errors
Current Record Address: *23.2* 3-31
file system object: *23.1* 3-11
ICOP+ mode: *23.2* 3-30
Logical Block Address: *23.2* 3-31
Logical Status Word: *23.2* 3-31
OPCODE: *23.2* 3-31, 5-12
partition formatting: *23.3* 3-39
RCRA: *23.2* 3-31
read error: *23.1* 3-11
Disk mirroring
breaking: *23.2* 3-32
COMDEV: *23.2* 3-24, 3-25
controllers supporting: *23.2* 5-12; *23.3* 3-40
ICOP+ SCSI disks: *23.2* 3-27
requirements for: *23.2* 5-11
Disk table: *23.0* 2-16, 3-6
DISK_MANAGER program: *23.1* 4-23
DISK_PAUSE command: *23.2* 3-25
Disks
See also Added disks; Assigned disks; Crash dump disks; Dual-ported disks; Paging partitions; Robust partitions; SCSI disks; SMD disks; Splitting disks
adding: *23.1* C-1
assignable: *23.3* 3-46
assigned disks maximum: *23.2* 3-20
available space: *23.0* 2-21
bad record: *23.1* 3-11
controller address: *23.3* 3-42, 3-44
crash analysis space: *23.1* 4-11
crash dump errors: *23.2* B-5
disk-directed portal: *23.0* 2-20
duplicate names: *23.0* 2-16
dynamic badspot handling: *23.3* 3-38, 3-39, 3-40
FIX_DISK required: *23.1* 4-2
formatting: *23.1* 3-6
generating crash dump: *23.2* B-2
geometry: *23.3* 3-44
head zero partition: *23.3* 3-39
heads: *23.3* 3-42, 3-44
initial state: *23.1* 4-2
initialization error: *23.2* B-3
ldev: *23.0* 3-6; *23.3* 3-43
listing: *23.0* 2-23; *23.3* 2-4
listing accessible: *23.0* 2-21
listing local: *23.0* 3-6
listing supported types: *23.3* 3-38
logical mounts: *23.0* 2-18
maximum number: *23.0* 2-16
mount error messages: *23.1* C-1
multiple partitions: *23.3* 3-39
number on system: *23.1* 2-14
pdev: *23.3* 3-42
physical save to tape: *23.0* 2-9
recovering: *23.1* 4-1
remote disks: *23.0* 2-16, 2-19, 2-21, 2-23, 2-24, 3-6
shutting down: *23.0* 2-23; *23.1* 4-2; *23.3* 3-47
sizes (table): *23.2* 5-8
splitting: *23.2* 3-22; *23.3* 3-38
surfaces: *23.3* 3-44
unit number: *23.3* 3-42, 3-44
DISKS command, maximum assigned disks: *23.2* 3-20
Distributed applications: *23.0* 2-16
Distributed Systems Management. *See* DSM subsystem
Documentation
See also Help facility
changes: *23.1* 2-23, 3-18; *23.2* 5-1; *23.3* 2-18, 3-48
list: *23.1* D-1; *23.2* E-1, E-1; *23.3* B-1
NTS: *23.1* 3-18
online: *23.1* ix; *23.2* ix; *23.3* xi
ORACLE: *23.2* 5-5
PRIMOS revision: *23.2* ix; *23.3* xi
Rev. 23.1: *23.1* ix
DOWN_LINE_LOAD* directory, ICOP+: *23.2* 3-27, 3-28
DRB configuration file: *23.1* 3-14; *23.2* 5-7
DS$SEND_CUSTOMER_UM subroutine: *23.0* 2-29
DSM subsystem
ADMIN_LOG subcommand: *23.0* 2-31; *23.3* 3-48
CONFIG_DSM subcommand: *23.0* 2-31
CONFIG_UM subcommand: *23.0* 2-30
configuring for network: *23.0* 2-19
controller DLL messages: *23.2* 3-30
customer product support: *23.0* 2-32
default configuration file: *23.2* 5-6
defining common name space: *23.0* 2-19
DISPLAY_LOG subcommand: *23.0* 2-32
DISTRIBUTE_DSM subcommand: *23.0* 2-30
documentation corrections: *23.1* 3-18; *23.2* 5-6, 5-7; *23.3* 3-48
DSM HELP subcommand: *23.0* 2-30
error level reporting options: *23.0* 2-29
functions: *23.2* 5-6
login attempt messages: *23.3* 3-10
message retention period: *23.3* 3-48
NPX messages: *23.2* C-11
paging threshold exceeded: *23.0* 2-26
PNX messages: *23.2* C-2
power supply message: *23.3* 3-4
server startup: *23.2* 3-5
STATUS_DSM subcommand: *23.0* 2-31
subroutines: *23.0* 2-29
UMH default configuration: *23.2* 5-6
unsolicited messages: *23.0* 2-29
DTAR 1, segment sharing: *23.1* 2-20
DTAR 2, segment sharing: *23.1* 2-20
Dual-ported disks
ICOP+ SCSI disks: *23.2* 3-27

priority select: *23.0* 2-22
PROM required: *23.3* 1-1
SMD dual-porting: *23.2* 3-27
Dumps. *See* Crash dumps
Dynamic segments, FS_RECOVER requirements: *23.1* 4-9

E

ECL
effects on prompts: *23.1* 2-5
EXPAND_WILD command: *23.0* 3-8
FORWARD_SEARCH command: *23.0* 3-8
KILL_LINE command: *23.0* 3-8
non-printing characters: *23.0* 3-8
OPEN_LINE command: *23.0* 3-8
overlay mode: *23.0* 3-7
prompts: *23.0* 3-8
REVERSE_SEARCH command: *23.0* 3-8
–STATUS option: *23.0* 3-8
TOGGLE_OVERLAY command: *23.0* 3-7
using within FS_RECOVER: *23.1* 4-14
WATCH monitoring: *23.1* 2-12
ED editor: *23.1* 1-1; *23.2* 1-1; *23.3* 1-3
Edit Command Line. *See* ECL
EDIT_CMD_LINE. *See* ECL
EDIT_PROFILE utility (obsolete): *23.3* 1-4
CONFIG_USERS compatibility: *23.3* 1-4
duplicating user profile: *23.1* 3-17
incompatibility: *23.2* 3-9, 3-13
replaced by CONFIG_USERS: *23.0* 2-1
retired: *23.2* 1-4
security enhancements: *23.0* 2-10
–SERVER option: *23.0* 2-12
Eligibility timeslice
default value: *23.0* 2-15
listing: *23.0* 2-15
setting: *23.0* 2-14
EMACS
during installation: *23.1* 1-1; *23.2* 1-1; *23.3* 1-3
keybindings: *23.0* 3-9
WATCH monitoring: *23.1* 2-12
ENVCOM file: *23.1* 3-10
EPF$INVK subroutine: *23.1* 2-24
EPF$MAP subroutine: *23.1* 2-24
EPF$REG subroutine: *23.1* 2-24
Error codes
displaying at prompt: *23.1* 2-5
E$ codes: *23.1* 2-26; *23.3* 2-22
ERROR_NOTIFY environment directive: *23.3* 3-25
Ethernet controllers: *23.1* 3-12
EXB tape drives. *See* Tape drives, Model 4598; Tape drives, Model 4601
Extended index libraries: *23.1* 3-13
Extended indexes: *23.0* 2-7
EXTR$A subroutine: *23.0* 3-5

F

FED facility, installation: *23.3* 1-3
File system
at Rev. 23.0: *23.0* 2-16
integrity: *23.1* 4-1
pre-Rev. 23.0: *23.0* 2-16
File Transfer Service. *See* FTS subsystem
File units
duplicating: *23.1* 2-18
file being printed: *23.1* 3-10
Filenames
changing: *23.1* 2-1; *23.3* 3-20
determining from record address: *23.1* 3-10
separating pathname: *23.0* 3-5
suffixes: *23.1* 2-17; *23.2* 5-5
validating: *23.1* 2-17
Files
affected by a crash: *23.1* 4-8
COMO file, current: *23.1* 2-13
crash dump: *23.1* 4-15, 4-17
date/time attributes: *23.0* 2-6, 2-7
file unit info, duplicating: *23.1* 2-18
locks, displaying: *23.1* 2-2
name change during restore: *23.0* 2-8
name change while open: *23.3* 3-20
name change, reporting: *23.1* 2-1
open: *23.2* 2-7; *23.3* 3-20
printing: *23.1* 3-10
PRISAM: *23.2* 5-5
records used: *23.1* 2-4
search rules: *23.1* 4-9
size reporting: *23.2* 2-7
FIX_DISK command
–AC and –IC options: *23.2* 3-22; *23.3* 3-40
–CONVERT_21 option: *23.3* 3-40
–DBS option: *23.3* 3-40
–ODI and –RDI options: *23.2* 3-22; *23.3* 3-40
–SECTOR option: *23.3* 3-40
–COMDEV option: *23.1* 4-8
concurrent execution: *23.2* 3-20
FIX_DISK utility
automated: *23.1* 4-21
buffer write error: *23.0* 2-28
COMO files: *23.1* 4-7
deferred: *23.1* 4-18
determining if required: *23.1* 4-2; *23.2* 3-31
during PRIMOS install: *23.1* 1-2; *23.2* 1-2; *23.3* 1-3
FS_RECOVER recommendation examples: *23.1* 4-19
immediate: *23.1* 4-18
manager phantom: *23.1* 4-22; *23.3* 3-42
monitor: *23.1* 4-16
monitor phantom: *23.1* 4-7
not required: *23.1* 4-19
partial recovery: *23.3* 3-42
recommendations for running: *23.1* 4-18; *23.3* 3-41
SHUTDN command: *23.0* 2-28
warning messages: *23.3* 3-42
FIXBAT utility: *23.0* 2-35; *23.1* 3-6
Forced logout: *23.3* 3-15
Forced shutdown
disk errors: *23.0* 2-29
procedure for: *23.1* 4-12
Form feed. *See* Carriage return
FORMS facility: *23.3* 1-3
FS_RECOVER: *23.1* 4-4; *23.3* 3-41
ACL requirements: *23.1* 4-9
breaking out of: *23.1* 4-13
checking tape drive: *23.1* 4-15
COMO file for session: *23.1* 4-17
COMO files: *23.1* 4-16
control-P during: *23.1* 4-13
crash analysis (example): *23.1* 4-17
crash dump file: *23.1* 4-15, 4-17
crash dump to disk: *23.2* 3-35, 3-44
crash recovery analysis: *23.1* 4-16

data analysis messages: *23.1* 4-18
deferred FIX_DISK: *23.1* 4-18
directory: *23.1* 4-8
disk manager: *23.1* 4-23
disk manager's mailbox: *23.3* 3-41
dynamic segments: *23.1* 4-9
ECL environment: *23.1* 4-14
error messages: *23.1* 4-12, B-2
executing PRIMOS commands during: *23.1* 4-14
FIX_DISK not required: *23.1* 4-19; *23.3* 3-42
FIX_DISK recommendations: *23.1* 4-18; *23.3* 3-41
following forced shutdown: *23.1* 4-12
immediate FIX_DISK: *23.1* 4-18
installation errors: *23.1* 4-12, B-1
installation of: *23.1* 4-8
invocation messages: *23.1* B-1
invoking: *23.1* 4-12
invoking at coldstart: *23.1* 4-10
machine state: *23.1* 4-17
main menu: *23.1* 4-13
messages: *23.1* B-1
options: *23.1* 4-13
phantoms: *23.1* 4-8
PRIMOS.COMI changes: *23.1* 4-10
reading crash dump tapes: *23.1* 4-14, 4-16
record requirements: *23.1* 4-8
runtime warning messages: *23.1* B-3
security: *23.1* 4-9
stopping: *23.1* 4-13
using: *23.1* 4-12
version numbers: *23.1* 1-2; *23.2* 1-2; *23.3* 1-3, 3-41
working directory: *23.1* 4-16
FTP, incompatibility: *23.2* 3-9, 3-13
FTS subsystem, initializing: *23.2* 3-5
Function keys, defining keys for: *23.3* 4-13
Functions
ATTACH_POINT: *23.1* 2-13
COMO_INFO: *23.1* 2-13
displaying at prompt: *23.1* 2-6
GROUP_LIST: *23.1* 2-13
IF: *23.3* 2-11
PDEV: *23.3* 3-42
REVERSE: *23.1* 2-17
SYSTEM_INFO: *23.1* 2-14
SYSTEM_USAGE: *23.1* 2-14
USER_INFO: *23.1* 2-15
USER_USAGE: *23.1* 2-16
VALIDATE: *23.1* 2-17

G

G$METR subroutine, meter_mem structure: *23.3* 3-3
General registers: *23.2* 5-3
Global Mount Table: *23.0* 2-20
listing contents: *23.0* 2-21
reading: *23.0* 3-7
reading by AT$ANY: *23.0* 3-5
Global variables
displaying at prompt: *23.1* 2-6
.SMVARS: *23.3* 4-2
.TERMINAL_TYPES: *23.0* 2-2, 2-31; *23.3* 2-2, 3-16
GMT. *See* Global Mount Table
GPATH$ subroutine: *23.0* 3-6
GROUP_LIST function: *23.1* 2-13
Groups. *See* ACL groups

H

Half-duplex, WATCH sessions: *23.1* 2-12
Halts and hangs. *See* System crashes
Hardcopy terminal
accessing CONFIG_USERS: *23.0* 2-1
accessing HELP text: *23.0* 3-3
HEADER environment directive, -NO_INITIAL_FF option: *23.1* 3-10
HELP command: *23.0* 3-3
terminal support: *23.3* 2-2, 3-16
text enhanced: *23.3* 2-2
Help facility: *23.0* 3-2
ACL protection: *23.0* 3-3
adding text: *23.0* 3-3
installing: *23.0* 1-2
HELP* directory: *23.0* 1-2
Hexadecimal numbers, validating: *23.1* 2-18

I

ICE command, freeing shared segments: *23.1* 2-22
ICOP disk controller mode: *23.2* 3-26
ICOP+ disk controller mode: *23.2* 3-26
disk dual-porting support: *23.2* 3-27
disk error messages: *23.2* 3-30
disk mirroring support: *23.2* 3-27
DISK_PAUSE support: *23.2* 3-25
down line load messages: *23.2* 3-28
installing: *23.2* 3-27
losing ICOP+ microcode: *23.2* 3-32
SPIN_DOWN support: *23.2* 3-24
supported controllers: *23.2* 3-27
ICRT routine: *23.3* 4-3
ICS3 communications controller: *23.3* 3-48
IDC1 controller: *23.3* 3-40
Idle time, displaying: *23.3* 3-2
IF function: *23.3* 2-11
Index files: *23.0* 2-7, 2-8; *23.1* 3-13
INDEX_LIB_MANAGER command: *23.1* 3-13
INFORMATION. *See* Prime INFORMATION
INIT command, -SEM_INIT option: *23.1* 3-6
INIT_RECOVER.CPL: *23.1* 4-10; *23.3* 3-42
INITIALIZE_SEARCH_RULES command: *23.3* 3-17
Installation
COMI install files: *23.0* 1-3
FS_RECOVER: *23.1* 1-2, 4-4, 4-8; *23.2* 1-2; *23.3* 1-3
Help files: *23.0* 1-2
ICOP+: *23.2* 3-27
MAKE and FIX_DISK: *23.3* 3-37
PNX: *23.2* 1-2, 4-5
PRIMOS: *23.0* 1-1; *23.1* 1-1; *23.2* 1-1; *23.3* 1-2
PRIMOS microcode: *23.1* 1-2; *23.2* 1-3; *23.3* 1-1
Quick Boot: *23.2* 3-1
removing software: *23.0* 1-3
testing: *23.0* 1-3
translator family libraries: *23.2* 1-2; *23.3* 1-3
translator family release: *23.0* 1-1, 1-2; *23.2* 1-1; *23.3* 1-3
Instruction sets, documentation corrections: *23.2* 5-3; *23.3* 2-18
InterServer Communications facility. *See* ISC facility
I/O buffers, monitoring: *23.1* 2-10

I/O operations
metering: *23.1* 3-12
monitoring: *23.1* 2-8
pausing: *23.2* 3-25
I/O time
displaying at prompt: *23.1* 2-5, 2-6
since boot: *23.1* 2-15
since login: *23.1* 2-16
ISC facility
configuring: *23.0* 2-11
documentation corrections: *23.2* 5-3
SendConfigurationBlock structure: *23.3* 2-21
IX mode: *23.3* 1-1

J

JOB command
–COMO option: *23.3* 3-21
–DISPLAY option: *23.3* 3-21
–STATUS option: *23.3* 3-21
command line length: *23.3* 3-21
JSXB instruction, documentation correction: *23.3* 2-18

K

Keybindings file: *23.0* 3-9
Keys file
compiling: *23.3* 4-21
creating: *23.3* 4-4
example: *23.3* 4-20
format: *23.3* 4-19
location: *23.3* 4-2
modifying: *23.3* 4-4

L

LD command
–BY_FILE_TYPE option: *23.3* 2-2
–CAM option: *23.3* 2-2
–DAM option: *23.3* 2-3
–HEIGHT option: *23.3* 2-3
–NO_SIZE option: *23.3* 2-3
–RWLOCK option: *23.1* 2-2
–SAM option: *23.3* 2-3
–TOTAL option: *23.1* 2-4
–TRUNCATED option: *23.3* 2-3
–WIDTH option: *23.3* 2-3
header display: *23.1* 2-4
LDISK$ subroutine: *23.0* 3-6
LHC controllers
CPU usage: *23.1* 3-12
PRIMOS.COMI: *23.2* 4-9
LIST_CONTIGUOUS_BLOCKS command: *23.0* 2-21
LIST_DISKS command: *23.0* 2-21
LIST_EPF command, –REG option: *23.0* 2-34
LIST_LIBRARY_ENTRIES command, –REG option: *23.0* 2-34
LIST_MOUNTS command: *23.0* 2-21
LIST_REGISTERED_EPF command: *23.0* 3-1
LIST_SCHEDULER_ATTRIBUTES command: *23.0* 2-15
LIST_USERS command: *23.3* 2-4
examples: *23.3* 2-6
format: *23.3* 2-4
LLENT command. *See* LIST_LIBRARY_ENTRIES command
LOADTM, VCP command: *23.2* 3-2
Local Area Networks
all-PRIMOS systems: *23.2* 4-3
CPU usage: *23.1* 3-12
enhancing PRIMENET: *23.2* 4-3
performance: *23.2* 4-3
RINGNET: *23.2* 4-3
Locate buffers
flushing: *23.0* 2-28; *23.1* 4-3
maximum number of: *23.3* 3-2, 3-3
metering: *23.1* 3-12
records reserved: *23.0* 2-26
Logbook: *23.1* 4-17
LOGIN command: *23.3* 2-1
not required: *23.3* 3-8
LOGIN_SERVER program: *23.1* 3-1
setting defaults: *23.1* 3-3
Logins
changing projects: *23.2* 2-2
concurrent: *23.2* 3-10, 3-12, 3-13
configuring prompt: *23.1* 3-1
disabling after failed logins: *23.2* 3-10, 3-12
disabling until date: *23.2* 3-10, 3-14
disallowing if no carrier signal: *23.1* 3-13
external: *23.2* 2-6
failed attempt: *23.3* 3-9
failed remote login: *23.2* 3-17
implicit: *23.3* 3-8
initial: *23.0* 2-11
insufficient paging space: *23.0* 2-27
logging to DSM: *23.3* 3-9
maximum number per user: *23.2* 3-10, 3-12, 3-13
non-printing characters: *23.3* 3-12
password change forced: *23.0* 2-11
project assignment: *23.2* 2-3
prompt: *23.1* 3-1; *23.3* 3-8
remote retries: *23.2* 3-15
requiring password change: *23.2* 3-10, 3-13
time of: *23.1* 2-16
timeout during DISK_PAUSE: *23.2* 3-26
LOGOUT command
–BATCH option: *23.3* 3-14
–CHILD option: *23.3* 3-14
–HELP option: *23.3* 3-14
–NO_VERIFY option: *23.3* 3-14
–PHANTOM option: *23.3* 3-14
–QUERY option: *23.3* 3-14
–REMOTE option: *23.3* 3-14
–SPOOLER option: *23.3* 3-14
ALL option: *23.3* 3-9, 3-13
forced logout: *23.3* 3-15
format: *23.3* 3-12
wildcarding: *23.3* 3-12
Logouts, inactivity timeout: *23.3* 3-15
LOUTQM directive: *23.3* 3-15
LTS controllers, loading: *23.2* 5-7
LUDSK$ subroutine: *23.0* 3-6

M

MAGNET utility
COPY subcommand: *23.3* 2-19
documentation correction: *23.3* 2-19
Magnetic tapes. *See* Tapes
MAGSAV/RST subsystem
ACL protection rights: *23.2* 5-7
checkpoint restore: *23.0* 2-8
enhanced version: *23.0* 2-5
extended index libraries: *23.1* 3-13
extended indexing: *23.0* 2-7, 2-8
index library management: *23.1* 3-13
MAGRST options: *23.0* 2-7

MAGSAV –UPDT: *23.2* 5-7
MAGSAV options: *23.0* 2-5
MTRESUME reading: *23.0* 2-35
partial restore: *23.0* 2-8
.SAVE files: *23.0* 1-4
Maintenance Processor
See also VCP commands
entering: *23.1* 4-2
microcode: *23.3* 1-2
MAKE command
–AC and –IC options: *23.0* 2-24; *23.2* 3-22, 3-27, 3-34; *23.3* 3-38
–BADLEV option: *23.2* 3-22, 5-10
–COPY_BADSPOTS option: *23.3* 3-38
–DBS option: *23.3* 3-38, 3-39
–DISK_REVISION option: *23.2* 3-22
–FORMAT option: *23.3* 3-39
–FORMAT_OK option: *23.3* 3-37
–LIST_BADSPOTS option: *23.3* 3-37
–NEW_DISK option: *23.0* 2-24
–NO_QUERY option: *23.3* 3-37
–ODI and –RDI options: *23.2* 3-22, 5-9; *23.3* 3-38
–REPORT option: *23.1* 3-6
–RPT option: *23.1* 3-6
–SECTOR option: *23.3* 3-38
–SPLIT option: *23.3* 3-28, 3-34, 3-38
–UNCORR option: *23.2* 5-10
–USAGE option: *23.3* 3-37, 3-39
concurrent execution: *23.2* 3-20
CPL programs: *23.3* 3-40
crash dump disk: *23.2* 3-34; *23.3* 3-28
dynamic badspot handling: *23.3* 3-39
listing disk types: *23.3* 3-38
no options: *23.3* 3-38
option abbreviations: *23.3* 3-39
prompt sequence: *23.3* 3-38
Manuals. *See* Documentation
MAPS directory: *23.1* 4-9; *23.2* 3-33, 3-42
MASTER CLEAR button: *23.2* B-3
MAXUSR command, no logins permitted: *23.3* 3-8
Memory
amount configured: *23.1* 2-14
boot requirement: *23.0* 1-1
core dump size: *23.3* 3-29
determining size of: *23.1* 4-11
maximum size of: *23.3* 3-2
pages in use by process: *23.1* 2-16
pages in use by system: *23.1* 2-15
PNX network buffers: *23.2* 4-5, 4-10
sharing: *23.1* 2-20
wired pages: *23.1* 2-15, 2-16; *23.2* 4-10
Memory dump. *See* Crash dumps
MESSAGE command
–DISABLE_CRLF option: *23.3* 3-19
–DISPLAY option: *23.3* 2-9, 3-19
–ENABLE_CRLF option: *23.3* 3-19
–MAX_LENGTH option: *23.3* 3-19
effects on TALK: *23.1* 2-8
error messages: *23.3* 2-24
maximum message length: *23.3* 2-8
multiline messages: *23.3* 2-8
setting message attributes: *23.3* 3-18
subroutines: *23.3* 2-10, 3-19
unsent message: *23.3* 2-10
MGSET$ subroutine
maximum message length: *23.3* 3-19
multiline messages: *23.3* 3-19
Microcode requirements: *23.1* 1-2; *23.2* 1-3; *23.3* 1-1
boot code: *23.2* 3-1
MIDASPLUS
ADD1$ subroutine: *23.2* 5-5
documentation corrections: *23.2* 5-5
OPENM$ subroutine: *23.2* 5-5
MIRROR_ON command: *23.0* 2-22
Mirroring. *See* Disk mirroring
MM$SHARE_APM subroutine: *23.1* 2-21
error codes: *23.1* 2-26; *23.3* 2-22
MONITOR_NET utility, PNX monitoring: *23.2* 4-11, C-1
MOVEW$ subroutine: *23.2* 5-2
MPC4000 environment directive: *23.1* 3-8
MRG1$S subroutine: *23.1* 2-25
MSG$ST subroutine
maximum message length: *23.3* 2-10
multiline messages: *23.3* 2-10
MTRESUME command: *23.0* 2-35
MTRS directive: *23.1* 3-17
effects on T$MT: *23.1* 2-25
MAGSAV –BIG: *23.0* 2-6
Multiple attach points: *23.1* 2-18

N

NAM$AD_PORTAL subroutine: *23.0* 3-6
NAM$L_GMT subroutine: *23.0* 3-7
NAM$RM_PORTAL subroutine: *23.0* 3-7
Name Server: *23.0* 2-19
effects on LDISK$ subroutine: *23.0* 3-6
starting: *23.0* 2-21
stopping: *23.0* 2-22
updating: *23.0* 2-22
NETLINK
dropping line: *23.2* 3-18
login retries: *23.2* 3-15
Network considerations. *See* Common file system name space; PNX; PRIMENET
Network Process Extension. *See* NPX
Network Terminal Service. *See* NTS users
NLBUF directive: *23.3* 3-2, 3-3
Non-printable characters
control codes (list): *23.3* A-1
ECL support: *23.0* 3-8
LOGIN: *23.3* 3-12
WATCH monitoring: *23.1* 2-12
NPX
controller access: *23.2* 4-3
defined: *23.2* 4-3
DSM messages: *23.2* C-11
slaves: *23.2* 4-7
NRUSR directive: *23.3* 3-3
NSED editor: *23.1* 1-1; *23.2* 1-1; *23.3* 1-3
NSEG directive: *23.1* 3-16; *23.3* 3-2
NSLUSR directive, PNX: *23.2* 4-6, 4-7, 4-8
NTS users
CPU time used: *23.1* 3-12
documentation: *23.1* 3-18
failure: *23.0* 1-2
monitoring: *23.1* 2-8
Numbers, validating: *23.1* 2-18
NVMFS directive: *23.3* 3-2

O

Octal numbers, validating: *23.1* 2-18
Online help text: *23.0* 1-2
Operator commands. *See* PRIMOS commands

ORACLE subsystem
documentation: *23.2* 5-5
PRIMOS.COMI consideration: *23.2* 3-4
Origin directory
changing projects: *23.2* 2-3
defining user's: *23.0* 2-2
returning user's: *23.1* 2-13

P

Paging partitions
allocation checked: *23.0* 2-26
available paging records: *23.0* 2-26
creating: *23.2* 3-22; *23.3* 3-38
DISK_PAUSE restriction: *23.2* 3-25
I/O operations per second: *23.1* 3-12
maximum number of: *23.2* 3-20
maximum usage of: *23.0* 2-25
performance: *23.1* 3-12; *23.2* 3-23
SCSI disks: *23.2* 3-23
sizing: *23.3* 3-38
space for PNX slaves: *23.2* 4-10
SPIN_DOWN restriction: *23.2* 3-24
using as crash dump disk: *23.2* 3-34
PARSRV subroutine: *23.2* 5-2
Parallel shares. *See* Cold starts
Partitions. *See* Disks
Passwords
changes per hour: *23.2* 3-9, 3-13
changing projects: *23.2* 2-3, 2-4
configuring: *23.0* 2-2
EDIT_PROFILE options: *23.0* 2-10
encryption: *23.0* 2-10
restricting reuse: *23.2* 3-9, 3-13
specifying in EDIT_PROFILE: *23.1* 3-17
user must change: *23.0* 2-11; *23.2* 3-10, 3-13
validating: *23.1* 2-17
WATCH sessions: *23.1* 2-12
Pathnames
attach points: *23.0* 3-6
determining from record address: *23.1* 3-10
expanding partial: *23.0* 3-8
root directory: *23.0* 2-17
separating directory and entryname: *23.0* 3-5
PCL instruction: *23.2* 5-3
PDEV command: *23.3* 3-42
PDEV function: *23.3* 3-42
Performance
6250 bpi tape dumps: *23.1* A-2
backup to tape: *23.2* 3-26
changing attach points: *23.1* 2-18
crash dumps: *23.2* 3-33
number of processes: *23.2* 4-7
paging: *23.2* 3-23
PNX overhead: *23.2* 4-10
PRIMOS.COMI execution: *23.2* 3-3
restoring from tape: *23.0* 2-7, 2-8
scheduler tuning: *23.0* 2-14
system boot diagnostics: *23.2* 3-1
Phantoms
despooler: *23.1* 3-10
FIX_DISK manager: *23.1* 4-22
FIX_DISK monitor: *23.1* 4-7
for automated FIX_DISK: *23.1* 4-7, 4-22
for FS_RECOVER: *23.1* 4-8
for SHARES.COMI: *23.2* 3-4
identifying process as: *23.1* 2-15
listing: *23.3* 2-5
monitoring: *23.1* 2-8
project assignment: *23.2* 2-3
PHYSAV/RST subsystem: *23.0* 1-4
PL. *See* Prime INFORMATION
PIO/S system meter: *23.1* 3-12
PNC-II controller, PNX requirement: *23.2* 4-5
PNX: *23.2* 4-1
buffer requirement: *23.2* 4-5, 4-10
configuring: *23.2* 4-6
console messages: *23.2* D-1
controller requirements: *23.2* 4-3
defined: *23.2* 4-1
DSM messages: *23.2* C-1
initialization messages: *23.2* D-2
installing: *23.2* 1-2, 4-5
limits: *23.2* 4-3
modifying PRIMOS.COMI: *23.2* 4-8
packet size: *23.2* 4-4
performance monitoring: *23.2* 4-11
PNX_SERVER error messages: *23.2* D-2
PRIMOS revision requirement: *23.2* 4-4
RINGNET: *23.2* 4-3
SIM commands: *23.2* 4-11
slave processes: *23.2* 4-7
slave types: *23.2* 4-10
when used: *23.2* 4-3
PNX directory: *23.2* 4-5
PNX_SERVER server: *23.2* 4-9
Portals: *23.0* 2-20
creating: *23.0* 2-21, 3-6
deleting: *23.0* 2-21
deleting entry: *23.0* 3-7
listing: *23.0* 3-7
shutting down: *23.0* 2-23
Power supply, failure message: *23.3* 3-4
Power up. *See* Cold starts
PRIFORMA
ADMIT KEYS: *23.2* 5-4
assigning keys: *23.3* 4-12
color terminals: *23.3* 4-39
COMPILE_FORMKEY: *23.3* 4-21
COMPILE_FORMTERM: *23.3* 4-44
compiling keys file: *23.3* 4-21
CONFIG_USERS support: *23.3* 4-1
defining function keys: *23.3* 4-13
documentation corrections: *23.2* 5-3
DRAW_FORM: *23.3* 4-42
DTOFIELD: *23.2* 5-3
form borders: *23.3* 4-40
forms manipulation: *23.2* 5-4
ICRT routine: *23.3* 4-3
keys file format: *23.3* 4-19
LGET_FORM: *23.2* 5-4
message line attributes: *23.3* 4-40
MODIFY_FORMKEY: *23.3* 4-4
READ statement: *23.2* 5-4
SMVARS file: *23.3* 4-1
terminal definition: *23.3* 4-1
video file: *23.3* 4-22
WND_VIEW: *23.2* 5-4
PRIFORMA* directory: *23.3* 4-2
Prime INFORMATION
DISK_PAUSE effect upon: *23.2* 3-26
documentation corrections: *23.3* 3-48
PRIMENET
access to OSI levels: *23.2* 4-3
buffers: *23.2* 4-5
common file system name space: *23.0* 2-19
description: *23.2* 4-1
disk access: *23.0* 2-19
environment chart: *23.2* 4-2
extending with PNX: *23.2* 4-1

Local Area Network: *23.2* 4-3
maximum number of users: *23.3* 3-3
MONITOR_NET utility: *23.2* C-1
network topology: *23.2* 4-3
node ID at login: *23.3* 3-11
packet size: *23.2* 4-4
without PNX: *23.2* 4-3
X.25 compliance: *23.2* 4-1
X.25: *23.3* 3-48
X29 command: *23.3* 2-16
X.3 parameters: *23.3* 2-15
XLGC$ subroutine: *23.3* 2-19
PRIMENET Performance Tuned Extensions. *See* PNX
PRIMENET* directory
PNX files: *23.2* 4-9
X3_PROFILES: *23.3* 2-15
PRIMOS commands
See also VCP commands
ADD_PORTAL: *23.0* 2-21
ADDISK: *23.0* 2-22
ASSIGN: *23.3* 3-4
BATCH: *23.1* 3-6
BATGEN: *23.0* 2-36
CDD: *23.2* 3-36; *23.3* 3-28
CHANGE_PROJECT: *23.2* 2-3, 3-18
CNAME: *23.1* 2-1; *23.3* 3-20
COMO -QUERY: *23.2* 2-1
CREATE -REPORT: *23.1* 2-1
DISK_PAUSE: *23.2* 3-25
DISKS: *23.2* 3-20
exclamation point character: *23.1* 4-14
executing within FS_RECOVER: *23.1* 4-14
FIX_DISK: *23.3* 3-40
FIXBAT: *23.0* 2-35
HELP: *23.0* 3-3; *23.3* 2-2, 3-16
INDEX_LIB_MANAGER: *23.1* 3-13
INIT: *23.1* 3-6
INIT_RECOVER -PAUSE: *23.1* 4-10
INITIALIZE_SEARCH_RULES: *23.3* 3-17
JOB: *23.0* 2-35
LD: *23.1* 2-2; *23.3* 2-2
LIST_CONTIGUOUS_BLOCKS: *23.0* 2-21
LIST_EPF: *23.0* 2-34
LIST_LIBRARY_ENTRIES: *23.0* 2-34
LIST_MOUNTS: *23.0* 2-21
LIST_REGISTERED_EPF: *23.0* 3-1
LIST_SCHEDULER_ATTRIBUTES: *23.0* 2-15
LIST_USERS: *23.3* 2-4
LOGIN: *23.3* 2-1
LOGOUT: *23.3* 3-12
MAKE: *23.0* 2-24; *23.1* 3-6; *23.3* 3-37
MAXUSR: *23.3* 3-8
MESSAGE: *23.3* 2-7, 3-18
MIRROR_ON: *23.0* 2-22
MTRESUME: *23.0* 2-35
online help text for: *23.0* 3-2
PDEV: *23.3* 3-42
PROP: *23.3* 3-25
RDY: *23.1* 2-4
RECORD_TO_PATH: *23.1* 3-10
REGISTER_EPF: *23.0* 2-32
REMOVE_PORTAL: *23.0* 2-21
SEARCH_INDEX_LIB: *23.1* 3-15
SET_ASYNC: *23.1* 3-13
SET_LSR_DEFAULTS: *23.3* 3-8
SET_PGALARM: *23.0* 2-25
SET_SCHEDULER_ATTRIBUTES: *23.0* 2-14
SHOW: *23.1* 2-8
SHUTDN: *23.0* 2-23; *23.3* 3-47
SIZE: *23.2* 2-7
SPIN_DOWN: *23.2* 3-24
SPIN_UP: *23.3* 3-45
SPOOL: *23.2* 2-9, 2-10, 2-13
START_LSR: *23.1* 3-1; *23.2* 3-15; *23.3* 3-7
START_NAMESERVER: *23.0* 2-21
STATUS: *23.0* 2-23; *23.1* 4-11; *23.2* 2-7; *23.3* 3-46
STOP_NAMESERVER: *23.0* 2-22
SYSTEM_RECOVER: *23.2* 3-47
TALK: *23.1* 2-8
TERM: *23.3* 2-10
UNREGISTER_EPF: *23.0* 2-33
UPDATE_NAMESERVER: *23.0* 2-22
USAGE: *23.0* 2-36; *23.1* 3-12; *23.3* 3-2
WATCH: *23.1* 2-8
PRIMOS II (obsolete): *23.0* 1-4
PRIMOS revision
cpu models not supported: *23.3* 1-1
displaying at login: *23.1* 3-1
displaying at prompt: *23.1* 2-6
documentation for: *23.0* A-1; *23.1* ix, 3-18, D-1; *23.2* ix, E-1, E-1; *23.3* xi, B-1
FS_RECOVER support: *23.1* 1-2, 4-4; *23.2* 1-2; *23.3* 1-3
installing: *23.0* 1-1; *23.1* 1-1; *23.2* 1-1; *23.3* 1-2
microcode requirements: *23.1* 1-2; *23.2* 1-3; *23.3* 1-1
network nodes not same: *23.0* 2-21; *23.2* 3-15
new features listed: *23.0* 1-5; *23.1* 1-3; *23.2* 1-5; *23.3* 1-5
PNX support: *23.2* 4-1, 4-4
product retirements: *23.0* 1-4; *23.1* 1-3; *23.2* 1-4; *23.3* 1-4
Quick Boot support: *23.2* 3-1
returning to function: *23.1* 2-14
RINGNET nodes: *23.2* 4-8
translator family: *23.1* 1-1; *23.2* 1-1; *23.3* 1-3
PRIMOS.COMI file
activating crash dump disk: *23.2* 3-36
COMM_CONTROLLER command: *23.2* 3-5
DBMS restriction: *23.2* 3-3
DISCOVER_DBMS restriction: *23.2* 3-3
DSM server: *23.2* 3-5
execution: *23.1* 4-12
FS_RECOVER: *23.1* 4-10
FTS server: *23.2* 3-5
INIT_RECOVER -PAUSE: *23.1* 4-10
ORACLE subsystem: *23.2* 3-4
parallel sharing example: *23.2* 3-5
pausing: *23.1* 4-10, 4-12
performance: *23.2* 3-3
PNX support: *23.2* 4-8
PRISAM restriction: *23.2* 3-3
ROAM restriction: *23.2* 3-3
setting system recovery: *23.2* 3-46, 3-52
sharing resources: *23.2* 3-3
SPOOL restriction: *23.2* 3-3
START_NAMESERVER command: *23.0* 2-21
START_NET command: *23.2* 4-8
SYSTEM_RECOVER command: *23.2* 3-46
TCP server: *23.2* 3-5
timing considerations: *23.2* 3-4
PRINET directory: *23.2* 4-5
Print controllers, MPC4000: *23.1* 3-8

Printers
  See also Spooler subsystem
  Centronics: *23.1* 3-8
  Dataproducts: *23.1* 3-8
  double-sided laser printers: *23.1* 3-10
  draft copies: *23.2* 2-13
  error notification: *23.3* 3-25
  fonts: *23.2* 2-9
  form feed: *23.1* 3-10
  graphic: *23.1* 3-9
  HP LaserJet: *23.3* 3-22
  landscape format: *23.2* 2-10
  n-up printing: *23.2* 2-10
  no-format mode: *23.2* 2-13
  point size: *23.2* 2-9
  portrait format: *23.2* 2-10
  PostScript: *23.1* 3-8; *23.2* 2-9; *23.3* 3-24
  printing tables: *23.2* 2-10
  setting speed: *23.1* 3-9
  thumbnail printing: *23.2* 2-10
  turned pages: *23.2* 2-10
Printing. *See* Spooler subsystem
PRISAM subsystem
  documentation corrections: *23.2* 5-5
  PRIMOS.COMI restriction: *23.2* 3-3
  Z$OPEN subroutine: *23.2* 5-5
PRI$CHD subroutine: *23.2* 2-6
PRMPC command (obsolete): *23.0* 1-4
Processes
  idle: *23.1* 2-15
  listing: *23.3* 2-4
  maximum number of: *23.2* 4-7
  NETMAN: *23.2* 4-9
  NSP: *23.2* 4-9
  returning ID: *23.1* 2-15
  slaves: *23.2* 4-7
  spooler: *23.3* 3-27
  suspended: *23.1* 2-15
  user information: *23.1* 2-15
Project Administrators, creating: *23.0* 2-3
Project assignments
  adding project to system: *23.0* 2-3
  assigning user to project: *23.0* 2-2
  change in progress: *23.2* 2-6
  changing: *23.2* 2-3
  default: *23.2* 2-3; *23.3* 3-11
  deleting: *23.0* 2-3
  deleting project from system: *23.0* 2-3
  enabling changing: *23.2* 3-18
  listing: *23.2* 2-3; *23.3* 2-5
  logging to DSM: *23.3* 3-11
  returning ID: *23.1* 2-15
  TALK server: *23.1* 2-8
  using subroutines: *23.2* 2-4
  validating ID: *23.1* 2-17
Prompts
  carriage return: *23.1* 2-6
  error prompt: *23.3* 3-7
  error prompts: *23.1* 2-4
  login: *23.1* 3-1
  maximum size of: *23.3* 3-8
  maxusr prompt: *23.3* 3-8
  ready prompt: *23.1* 2-4; *23.3* 3-8
  ready prompt variables: *23.1* 2-5; *23.3* 3-8
  redisplaying login prompt: *23.1* 3-2
  variables: *23.1* 2-5
PROP command
  –COLDSTART option: *23.3* 3-26
  –MAX_SIZE option: *23.3* 3-25
  –MIN_SIZE option: *23.3* 3-26
  –START option: *23.3* 3-27
PRSER command (obsolete): *23.0* 1-4
PRVER command (obsolete): *23.0* 1-4
PSR command: *23.0* 2-9
  COMDEV restriction: *23.0* 2-9

Q

QBOOT file: *23.2* 3-2
Quick Boot mode: *23.2* 3-1
  automated execution: *23.2* 3-50
  boot failure: *23.2* 3-3
  controller restriction: *23.2* 3-3
  tape unit restriction: *23.2* 3-3

R

R-mode: *23.0* 1-3; *23.2* 1-4
.RAS$ ACL group: *23.1* 3-10; *23.3* 3-46
RDLIN$ subroutine: *23.1* 2-25
RDY command
  –EP option: *23.1* 2-4
  –EXPAND_PROMPT option: *23.1* 2-4
  –NEP option: *23.1* 2-4
  –NO_EXPAND_PROMPT option: *23.1* 2-4
  error messages: *23.1* 2-7
Read and write locks
  C open statement: *23.1* 2-23
  displaying: *23.1* 2-2
RECORD_TO_PATH command: *23.1* 3-10
  error codes: *23.1* 2-27; *23.3* 2-23
Records
  converting addresses: *23.1* 3-10
  disk capacities: *23.2* 3-21
  disk error: *23.2* 3-31
  displaying number used: *23.1* 2-4
  FS_RECOVER requirements: *23.1* 4-8
  maximum tape record size: *23.1* 3-17
  minimum number for split disk: *23.2* 3-23
  number in crash dump: *23.2* 3-40
  number in file: *23.2* 2-7
  paging requirements: *23.2* 3-23
  required for full crash dump: *23.1* 4-11; *23.2* 3-41
  required for partial crash dump: *23.1* 4-11; *23.2* 3-41
Recovery. *See* System Recovery facility
REGISTER_EPF command: *23.0* 2-32
Registered EPFs
  EPF$REG subroutine: *23.1* 2-24
  insufficient segments: *23.1* 2-26; *23.3* 2-22
  list dependencies: *23.0* 3-1
  list unresolved entrypoints: *23.0* 3-1
  listing: *23.0* 2-34
  listing registered libraries: *23.0* 2-34
  registering: *23.0* 2-32
  registration levels: *23.1* 2-26; *23.3* 2-22
  translator release requirement: *23.0* 1-3, 3-1
  unregistering: *23.0* 2-33
Remote files
  maximum line length: *23.1* 2-25
  Remote File Access: *23.0* 2-22, 2-23
Remote system name, displaying at prompt: *23.1* 2-6
Remote users
  during DISK_PAUSE: *23.2* 3-25
  failed login: *23.2* 3-17
  login retries: *23.2* 3-15
  monitoring: *23.1* 2-8
  user information: *23.1* 2-15
  WATCH sessions: *23.1* 2-12

REMOVE_PORTAL command: *23.0* 2-21
Resident Forced Shutdown. *See* RFS
Restoring from tape. *See* MAGSAV/RST subsystem
REVERSE function: *23.1* 2-17
RFS: *23.1* 4-2
  automated execution: *23.2* 3-50
  console messages: *23.1* 4-3
  initial disk state: *23.1* 4-2
  invoking: *23.1* 4-2
  warm start restriction: *23.2* A-1, B-1
RINGNET
  differing PRIMOS revisions: *23.2* 4-8
  PNX support: *23.2* 4-3
RJE emulators: *23.0* 1-4
RMSGD$ subroutine: *23.2* 5-2
ROAM subsystem
  PRIMOS.COMI restriction: *23.2* 3-3
  saving ROAM files: *23.0* 2-9
Robust partitions
  displaying: *23.0* 2-24
  system crash recovery: *23.2* 3-20
Root directory: *23.0* 2-17
  ACL protection: *23.0* 3-5
  attaching to: *23.0* 3-5
  replicating: *23.0* 2-19
  root-directed portal: *23.0* 2-20
RUN 660, VCP command: *23.2* 3-47
RUN 661, VCP command: *23.2* 3-35
  crash dump errors: *23.2* B-5
  example: *23.2* B-2
  initialization errors: *23.2* B-3
RUN 662, VCP command: *23.1* 4-2
RUN 773, VCP command: *23.2* 3-20
  example: *23.1* A-1
RUN 774, VCP command: *23.2* 3-19, A-2
  example: *23.2* A-2
RUN 775, VCP command: *23.2* 3-20
RUN 776, VCP command: *23.2* 3-20
RUN 777, VCP command: *23.2* 3-20
  example: *23.1* A-2
RUNTM, VCP command: *23.2* 3-2

S

S$START_COPY print handler key: *23.1* 3-10
SAD. *See* System Administration Directory
Scheduler: *23.0* 2-14
  monitoring: *23.0* 2-36
  returning priority: *23.1* 2-15
SCSI disks
  badspot checking: *23.2* 3-22; *23.3* 3-40
  controller mode: *23.2* 3-22; *23.3* 3-40
  crash dump disks: *23.2* 3-42
  current record address: *23.2* 3-31
  dual porting: *23.2* 3-27
  in 75500-6PK device module: *23.2* 3-24, 3-25, 5-9
  logical block address: *23.2* 3-31
  logical status word: *23.2* 3-31
  malfunctioning: *23.2* 3-24
  mirroring: *23.2* 3-27
  paging partitions: *23.2* 3-22
  partitioning: *23.2* 3-22, 3-23
  records per surface: *23.2* 5-9
  records unavailable (table): *23.2* 3-23
  sectoring interleaving: *23.2* 3-22; *23.3* 3-41
  sizes (table): *23.2* 5-8
  spin down: *23.2* 3-24
  spin up: *23.3* 3-45
  splitting: *23.2* 3-23
  swapping on-line: *23.2* 3-25
  write protection: *23.2* 3-22
Search rules
  ATTACH$: *23.0* 2-24
  AUTOPSY: *23.1* 4-9
  changes for FS_RECOVER: *23.1* 4-9
  COMMAND$: *23.1* 4-9
  ENTRY$: *23.1* 1-2, 4-9; *23.2* 1-2; *23.3* 1-3
  MAPS: *23.1* 4-9
  system defaults: *23.3* 3-17
SEARCH_INDEX_LIB command: *23.1* 3-15
Security
  See also ACL protection; Passwords
  EDIT_PROFILE options: *23.0* 2-10
  failed login attempts: *23.3* 3-9
  FS_RECOVER: *23.1* 4-9
  logging logins: *23.3* 3-9
  network node access: *23.2* 3-15
  online HELP text: *23.0* 3-3
  password encryption: *23.0* 2-10
  user password control: *23.2* 3-9
SEG utility, error message: *23.3* 2-19
Segment directories, displaying locks: *23.1* 2-2
Segments
  assigning for sharing: *23.1* 2-22
  configuring: *23.1* 3-16
  FS_RECOVER requirements: *23.1* 4-9
  in use by process: *23.1* 2-16
  maximum number of: *23.1* 3-16; *23.3* 3-2
  MOVEW$ subroutine: *23.2* 5-2
  number in use: *23.1* 2-15
  sharing: *23.1* 2-20
  VMFA: *23.1* 2-22
SEM$US subroutine: *23.2* 2-7
Semaphores
  DISK_PAUSE effect upon: *23.2* 3-26
  reinitializing for batch: *23.1* 3-6
  user to be notified: *23.2* 2-7
SEMFIL file: *23.1* 3-6
Servers
  listing: *23.3* 2-5
  Name Server: *23.0* 3-6
  PNX_SERVER: *23.2* 4-9, D-2
  user information: *23.1* 2-15
SET_ASYNC command
  -DCD_LOGIN option: *23.1* 3-13
  -NO_DCD_LOGIN option: *23.1* 3-13
SET_LSR_DEFAULTS command: *23.1* 3-3
  prompt defaults: *23.3* 3-8
  remote login retries: *23.2* 3-16
SET_PGALARM command: *23.0* 2-25
SET_SCHEDULER_ATTRIBUTES command: *23.0* 2-14
SETU$S subroutine: *23.1* 2-25
SHARE command: *23.2* 3-5
SHOW command: *23.1* 2-8
  syntax: *23.1* 2-9
SHUTDN command: *23.0* 2-23
  -VERIFY option: *23.3* 3-47
  errors during: *23.0* 2-28
  format: *23.3* 3-47
SIM commands, PNX monitoring: *23.2* 4-11, C-1
Singly-rooted file system: *23.0* 2-16
  AT$ABS subroutine support: *23.0* 3-4
  AT$ANY subroutine support: *23.0* 3-5
  AT$ROOT subroutine support: *23.0* 3-5

EXTR$A subroutine support: *23.0* 3-5
GPATH$ subroutine support: *23.0* 3-6
SIZE command: *23.2* 2-7
SIZE$ subroutine: *23.2* 5-2
Slave processes: *23.2* 4-7
configuring for PNX: *23.2* 4-6
listing: *23.3* 2-5
maximum number of: *23.2* 4-7
monitoring: *23.2* 4-10
PNX_SLAVES$: *23.2* 4-9, 4-10
slave types: *23.2* 4-10
SLAVE$: *23.2* 4-10
SMDs
badspot checking: *23.0* 2-24
crash dump disks: *23.2* 3-42
dual-porting: *23.2* 3-27
paging partitions: *23.2* 3-23
sizes (table): *23.2* 5-8
SMSG$ subroutine, maximum message length: *23.3* 2-10
SMVARS file
defined: *23.3* 4-1
keys file entry: *23.3* 4-21
location: *23.3* 4-2
video file entry: *23.3* 4-44
Software
See also PRIMOS revision
documentation: *23.0* A-1; *23.1* D-1; *23.2* E-1, E-1; *23.3* B-1
FS_RECOVER as IPR: *23.1* 1-2; *23.2* 1-2; *23.3* 1-3
independent products: *23.1* 1-1; *23.2* 1-1; *23.3* 1-2
list product names: *23.0* 2-30
removing from system: *23.0* 1-3
retired: *23.0* 1-4; *23.1* 1-3; *23.2* 1-4; *23.3* 1-4
SyncSort/PRIME: *23.3* 1-4
testing installation of: *23.0* 1-3
translator family release: *23.0* 1-2; *23.1* 1-1; *23.2* 1-1; *23.3* 1-3
SP$LIB library: *23.0* 2-13
SPIN_DOWN command: *23.2* 3-24
SPIN_UP command: *23.3* 3-45
Splitting disks: *23.2* 3-22
file system minimum size: *23.2* 3-23
for crash dump disk: *23.2* 3-34, 3-41
for paging partition: *23.2* 3-22
SPOOL command
-AT DRAFT/ option: *23.2* 2-13
-AT n-UP options: *23.2* 2-10
-ATTRIBUTE option: *23.3* 3-24
-LIST option: *23.0* 2-13
-NO_FORMAT option: *23.2* 2-13; *23.3* 3-24
-PROC option: *23.2* 2-12
-SET_FONT option: *23.2* 2-9
-SET_LANDSCAPE option: *23.2* 2-10
SPOOL* directory: *23.0* 2-13
Spooler subsystem: *23.1* 3-8; *23.2* 2-9; *23.3* 3-22
See also Printers
ACL groups: *23.3* 3-25
ASYNC -MST directive: *23.1* 3-10
attribute aliases: *23.3* 3-24
ATTRIBUTE directive: *23.3* 3-23
customizing display: *23.0* 2-13
data file unit number: *23.1* 3-10
DEVICE directive: *23.3* 3-22
DEVICE POSTSCRIPT directive: *23.3* 3-24
documentation corrections: *23.1* 3-21
draft copies: *23.2* 2-13
ERROR_NOTIFY directive: *23.3* 3-25
HEADER -NO_INITIAL_FF directive: *23.1* 3-10
in PRIMOS.COMI: *23.2* 3-3
landscape format: *23.2* 2-10; *23.3* 3-23
listing processes: *23.3* 2-5
listing spooler processes: *23.3* 3-27
mandatory attributes: *23.3* 3-24
MPC4000 controller: *23.1* 3-8
n-up printing: *23.2* 2-10
no-format mode: *23.2* 2-13
portrait format: *23.2* 2-10; *23.3* 3-23
print buffer: *23.1* 3-10
print job size: *23.3* 3-25
process type: *23.3* 3-27
restarting: *23.3* 3-26
SET_LANDSCAPE directive: *23.3* 3-23
SET_PORTRAIT directive: *23.3* 3-23
SP$LIB library: *23.0* 2-13
thumbnail printing: *23.2* 2-10
turned pages: *23.2* 2-10
type fonts and sizes: *23.2* 2-9
user access: *23.3* 3-25
SRCH$$ subroutine: *23.0* 3-7
SRTF$$ subroutine: *23.1* 2-25
START_LSR command: *23.1* 3-1
-ERROR_PROMPT option: *23.3* 3-7
-IMPLICIT_LOGIN option: *23.3* 3-8
-MAXUSR_PROMPT option: *23.3* 3-8
-READY_PROMPT option: *23.3* 3-8
-RETRIES option: *23.2* 3-15
syntax: *23.2* 3-15; *23.3* 3-7
START_NAMESERVER command: *23.0* 2-19, 2-21
STARTUP (obsolete): *23.0* 1-4
Startup file. *See* PRIMOS.COMI file
Static-mode execution
displaying at prompt: *23.1* 2-6
MTRESUME command: *23.0* 2-35
STATUS command: *23.2* 2-7
ALL option: *23.0* 2-23
DEVICES option: *23.3* 3-46
DISKS option: *23.0* 2-21, 2-24
SYSTEM option: *23.1* 4-11
USERS option: *23.0* 2-24
STOP, VCP command: *23.1* 4-2
STOP_NAMESERVER command: *23.0* 2-22
Strings. *See* Character strings
Subroutines
AT$ABS: *23.0* 3-4
AT$ANY: *23.0* 3-5
AT$ROOT: *23.0* 3-5
CF$EXT: *23.1* 2-24
CHG$PW: *23.3* 2-23
CHPRJ$: *23.2* 2-4
CPUID$: *23.2* 5-1; *23.3* 3-1
CUS$ subroutines: *23.0* 2-38; *23.2* 3-11, 3-14; *23.3* 2-20
DIR$LS: *23.3* 2-14
DIR$RD: *23.3* 2-14
DIR$SE: *23.3* 2-14
DS$SEND_CUSTOMER_UM: *23.0* 2-29
EPF$MAP: *23.1* 2-24
EPF$REG: *23.1* 2-24
EXTR$A: *23.0* 3-5
G$METR: *23.3* 3-3
GPATH$: *23.0* 3-6
LDISK$: *23.0* 3-6
LUDSK$: *23.0* 3-6
MGSET$: *23.3* 3-19
MM$SHARE_APM: *23.1* 2-21; *23.3* 2-22
MRG1$S: *23.1* 2-25

MSG$ST: *23.3* 2-10
NAM$AD_PORTAL: *23.0* 3-6
NAM$L_GMT: *23.0* 3-7
NAM$RM_PORTAL: *23.0* 3-7
PRJ$CHD: *23.2* 2-6
RDLIN$: *23.1* 2-25
SEM$US: *23.2* 2-7
SETU$S: *23.1* 2-25
SMSG$: *23.3* 2-10
SRCH$S: *23.0* 3-7
SRTF$S: *23.1* 2-25
T$MT: *23.1* 2-25; *23.3* 3-5
T$SLC0: *23.3* 2-20
UT$COPY: *23.1* 2-18; *23.3* 2-22
WTLIN$: *23.1* 2-25
Supervisor terminal. *See* System console
SYN$GRTR subroutine: *23.2* 5-2
SYN$GTWT subroutine: *23.2* 5-2
SYN$GWT subroutine: *23.2* 5-2
SYN$MVTO subroutine: *23.2* 5-2
Synchronizers and timers
configuring: *23.0* 2-11
documentation corrections: *23.2* 5-2
SyncSort/PRIME (obsolete): *23.3* 1-4
SYSCLR, VCP command: *23.2* A-3, A-4, B-3
System. *See* CPU
System Administration Directory
accessing: *23.0* 2-2, 2-38
command access: *23.0* 2-2
creating: *23.0* 2-2, 2-38
EDIT_PROFILE access: *23.2* 3-9, 3-13
FTP access: *23.2* 3-9, 3-13
incompatibility: *23.2* 3-9, 3-13
subroutine access: *23.0* 2-38
System Administrator
ACL groups: *23.0* 2-31
configuring: *23.0* 2-4
disabling WATCH monitoring: *23.1* 2-9
maintaining log book: *23.1* 4-17
RECORD_TO_PATH access: *23.1* 3-10
segment requirements: *23.1* 4-9
SET_ASYNC_DCD_LOGIN access: *23.1* 3-13
WATCH access permission: *23.1* 2-10
System console
broadcast messages: *23.2* 5-2
CONFIG_USERS: *23.0* 2-4
controller DLL messages: *23.2* 3-28
disk error messages: *23.0* 2-25
PNX error messages: *23.2* D-1
project assignment: *23.2* 2-3
remote login failure: *23.1* 3-13
System crashes
See also Cold starts; Crash dumps
analyzing data integrity: *23.1* 4-18
crash dump to disk: *23.2* 3-33
date and time of: *23.2* 3-38
determining machine state: *23.1* 4-17
during Quick Boot: *23.2* 3-3
effects on shared memory: *23.1* 2-22
forced shutdown: *23.0* 2-27; *23.1* 4-12
halting the machine: *23.1* A-2; *23.2* A-3, B-2
MASTER CLEAR button: *23.2* B-3
performing crash dump to disk: *23.2* 3-34
recovery recommendations: *23.1* 4-12
recovery tools: *23.1* 4-1
System metering: *23.1* 3-12
System name
displaying at login: *23.1* 3-1
displaying at prompt: *23.1* 2-6, 3-1
returning to function: *23.1* 2-14
System Recovery facility
automated execution: *23.2* 3-47
configuration options: *23.2* 3-49
configuring: *23.2* 3-52
manual execution: *23.2* 3-47
reporting configuration: *23.2* 3-51
systems supporting: *23.2* 1-3; *23.3* 1-2
SYSTEM_DEBUG* directory
ACL requirements: *23.1* 4-9
CRASH: *23.1* 4-9, 4-10, 4-15, 4-16; *23.2* 3-33
DISK_MANAGER.MBX: *23.3* 3-41
in COMMAND$ search rules: *23.1* 4-9
INIT_RECOVER.CPL: *23.1* 4-10
installing FS_RECOVER: *23.1* 4-8
SYSTEM_INFO function: *23.1* 2-14
SYSTEM_RECOVER command: *23.2* 3-47
example: *23.2* 3-52
status codes: *23.2* 3-51
SYSTEM_USAGE function: *23.1* 2-14

T

T$MT subroutine
documentation corrections: *23.1* 2-25
model 4601 tape drive: *23.3* 3-5
T$SLC0 subroutine, documentation correction: *23.3* 2-20
TALK facility: *23.1* 2-8
Tape controllers. *See* Disk and tape controllers
Tape drives
2382 controller: *23.2* A-6
7210 controller: *23.2* 3-27, A-6; *23.3* 3-4
8mm cartridge: *23.2* A-3; *23.3* 3-4
assigning for FS_RECOVER: *23.1* 4-15
cleaning message: *23.3* 3-6
crash dump commands: *23.2* 3-19
crash dump requirements: *23.2* A-1
crash dump to 6250 bpi: *23.1* A-2; *23.2* A-6
error messages: *23.1* 4-15; *23.3* 3-6
high density: *23.2* A-6; *23.3* 3-4
high speed: *23.3* 3-4
Kennedy: *23.2* A-6
Model 4594: *23.2* A-6
Model 4595: *23.2* A-3; *23.3* 3-4
Model 4596: *23.2* A-3; *23.3* 3-4
Model 4598: *23.2* A-3; *23.3* 3-4
Model 4601: *23.3* 3-4
model numbers: *23.1* 2-25
Quick Boot restriction: *23.2* 3-3
remote density: *23.2* A-6
setting tape density: *23.2* A-1
unable to reserve: *23.3* 3-6
Tapes
backup performance: *23.2* 3-26
backup tape contents: *23.0* 2-8
backups, labeling: *23.0* 2-6
backups, multi-reel: *23.0* 2-6
block size: *23.0* 2-6
boot tape, creating: *23.0* 2-6
buffers: *23.1* 2-25
cartridges (8mm): *23.2* A-3; *23.3* 3-4
compatibility: *23.3* 3-4
crash dump commands: *23.2* 3-19
crash dump, generating: *23.1* A-1; *23.2* A-3
crash dump, labeling: *23.1* A-3; *23.2* A-7
crash dump, multi-reel: *23.1* 4-16, A-2; *23.2* A-7

FS_RECOVER installation: *23.1* 4-8
maximum record size: *23.1* 3-17
physical save: *23.0* 2-9
restoring from: *23.0* 2-7
rewind and unload: *23.0* 2-6
T$MT subroutine: *23.1* 2-25; *23.3* 3-5
using crash dump: *23.1* 4-5, 4-14
write protection: *23.2* A-1, A-3
TCP/IP
printer error messages: *23.3* 3-25
printing over: *23.3* 3-26
starting server: *23.2* 3-5
TCP/IP environment directive
–DEFER option: *23.3* 3-26
–RELEASE option: *23.3* 3-26
TELNET
dropping line: *23.2* 3-18
login retries: *23.2* 3-15
TERM command, –DISPLAY option: *23.3* 2-10
TERM* directory: *23.0* 3-9; *23.3* 3-16
Terminals
See also Hardcopy terminal; System console
color: *23.3* 4-39
defining keys: *23.3* 4-4
defining screen display: *23.3* 4-22
display during WATCH session: *23.1* 2-12
for CONFIG_USERS: *23.3* 4-1
for HELP: *23.3* 3-16
keybindings: *23.0* 3-9; *23.3* 3-16, 4-1
modifying key definitions: *23.3* 4-4
monitoring: *23.1* 2-8
scroll left/right: *23.3* 4-42
TERMINFO facility: *23.3* 3-16
Text strings. *See* Character strings
Time
displaying at login: *23.1* 3-1
displaying at prompt: *23.1* 2-5
login: *23.1* 2-16
TPDUMP directive (obsolete): *23.2* 1-4; *23.3* 1-4
Translator family release: *23.0* 1-2; *23.1* 1-1; *23.2* 1-1; *23.3* 1-3
for registered EPFs: *23.0* 3-1
for source code: *23.3* 1-3
TTY mode, accessing HELP text: *23.0* 3-3

U

U$SPLR user type: *23.3* 3-27
UNREGISTER_EPF command: *23.0* 2-33
UNSPEC PL/I function: *23.3* 2-19
UPDATE_NAMESERVER command: *23.0* 2-22
USAGE command
–MULTI option: *23.3* 3-2
–SCHED option: *23.0* 2-36
–SYSTEM option: *23.1* 3-12
monitoring PNX: *23.2* 4-11
reading system fields: *23.1* 3-12
User 1. *See* System console
User commands. *See* PRIMOS commands
User IDs
adding to system: *23.0* 2-2, 2-38
changing projects: *23.2* 2-2
deleting from system: *23.0* 2-3, 2-38
displaying at prompt: *23.1* 2-6
logging to DSM: *23.3* 3-11
Project Administrator: *23.0* 2-3
returning: *23.1* 2-15
validating: *23.0* 2-38; *23.1* 2-17
waiting on semaphores: *23.2* 2-7
User numbers
displaying at prompt: *23.1* 2-6
returning: *23.1* 2-15
USER_INFO function: *23.1* 2-15
USER_USAGE function: *23.1* 2-16
Users
See also Remote users
disabling WATCH access: *23.1* 2-10
displaying at prompt: *23.1* 2-6
displaying WATCH monitoring: *23.1* 2-11
list: *23.0* 2-38
listing: *23.3* 2-5
login restrictions: *23.2* 3-9
monitoring: *23.1* 2-8
multiple logins and WATCH: *23.1* 2-11
returning number logged in: *23.1* 2-14
returning type: *23.1* 2-15
SYSTEM: *23.1* 4-9
viewing batch queues: *23.1* 3-6
UT$COPY subroutine: *23.1* 2-18
error codes: *23.1* 2-26; *23.3* 2-22

V

VALIDATE function: *23.1* 2-17
VCP commands
BOOT: *23.2* 3-2
BOOTP: *23.2* 3-2
BOOTQ: *23.2* 3-1
crash dump to disk: *23.2* 3-35
crash dump to tape: *23.2* 3-19
LOADTM: *23.2* 3-2
Resident Forced Shutdown: *23.1* 4-2
RUN 660: *23.2* 3-47
RUN 661: *23.2* 3-35, B-3
RUN 662: *23.1* 4-2
RUN 773: *23.2* 3-20
RUN 774: *23.2* 3-19, A-2
RUN 775: *23.2* 3-20
RUN 776: *23.2* 3-20
RUN 777: *23.2* 3-20
RUNTM: *23.2* 3-2
STOP: *23.1* 4-2
SYSCLR: *23.2* A-3, A-4, B-3
system recovery: *23.2* 3-47
Video file
color display: *23.3* 4-39
commands: *23.3* 4-23
compiling: *23.3* 4-44
creating: *23.3* 4-22
format: *23.3* 4-22
graphics rendition: *23.3* 4-37
keywords: *23.3* 4-30
location: *23.3* 4-2
modifying: *23.3* 4-22
percent commands: *23.3* 4-24
stack manipulation: *23.3* 4-25
VMFA segments: *23.1* 2-22

W

Warm starts
automated execution: *23.2* 3-50
following crash dump: *23.2* A-1, B-1
RFS restriction: *23.2* A-1, B-1
WATCH command: *23.1* 2-8
error codes: *23.1* 2-26; *23.3* 2-22
multiple sessions: *23.1* 2-12
syntax: *23.1* 2-11
.WATCH$ ACL group: *23.1* 2-10
disabling: *23.1* 2-9
Wide Area Networks: *23.2* 4-1

WILD function: *23.1* 2-23
  –TRUNC option: *23.3* 2-14
Wildcards
  listing groups: *23.1* 2-14
  validating strings: *23.1* 2-18
WTLIN$ subroutine: *23.1* 2-25

X

X.25 network protocol
  LAN support: *23.2* 4-3
  number of slave users: *23.2* 4-7
  performance: *23.3* 3-48
  PRIMENET with PNX: *23.2* 4-1
X.25PNX directory: *23.2* 4-5
X_RESTORE commands: *23.0* 1-4
X29 command: *23.3* 2-16
X.3 parameters: *23.3* 2-15
XLGC$ subroutine: *23.3* 2-19